

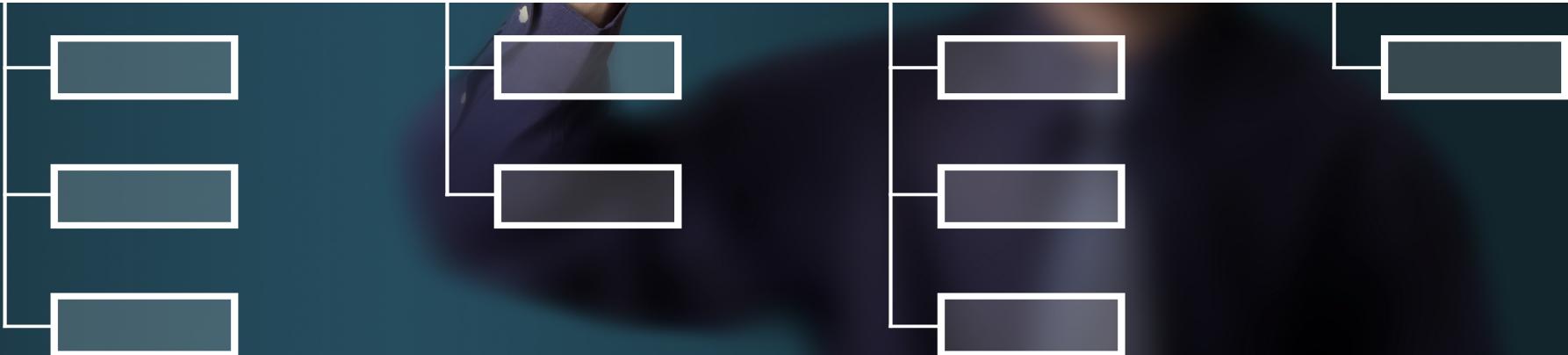


Lean Startup

A focus on practices

© Jerome Kehrli @ niceideas.ch

<https://www.niceideas.ch/roller2/badtrash/entry/lean-startup-a-focus-on>





1 The Lean Startup (Introduction)

2 The 4 steps to the Epiphany

2.1 Customer Discovery

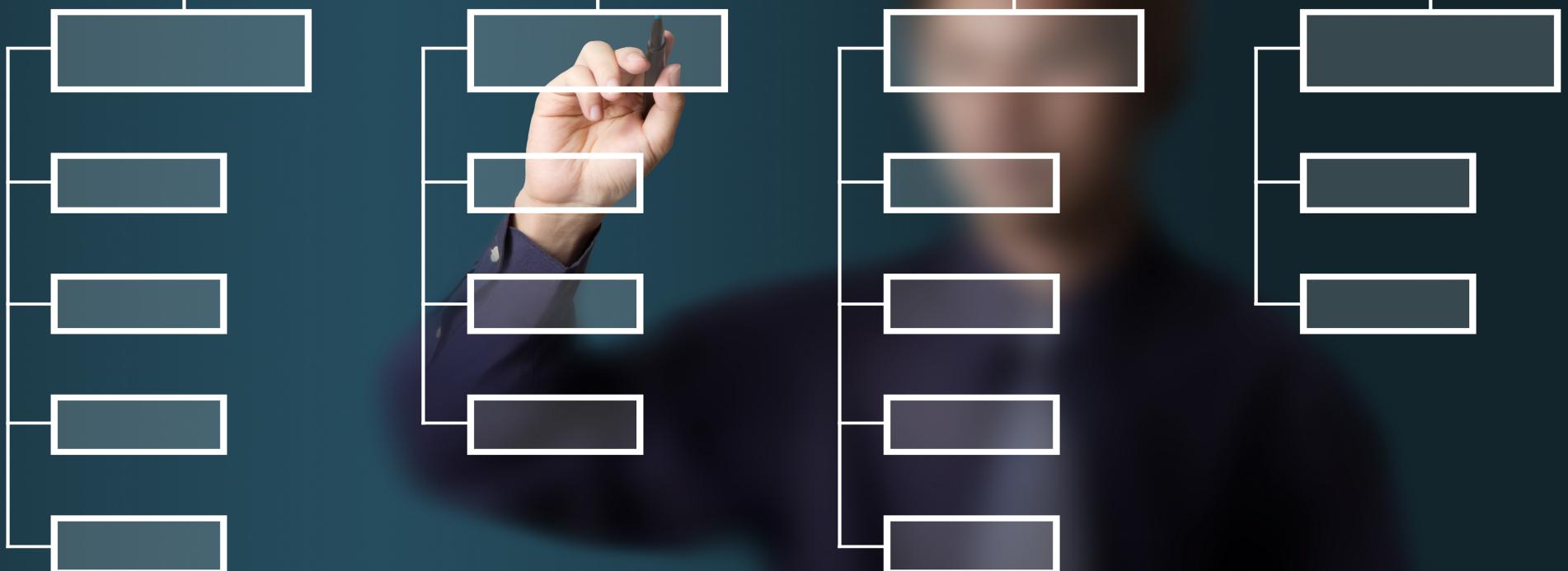
2.2 Customer Validation

2.3 Re-adapt the product

2.4 Get new Customers

3 Wrap-up and Take Aways

1. The Lean Startup

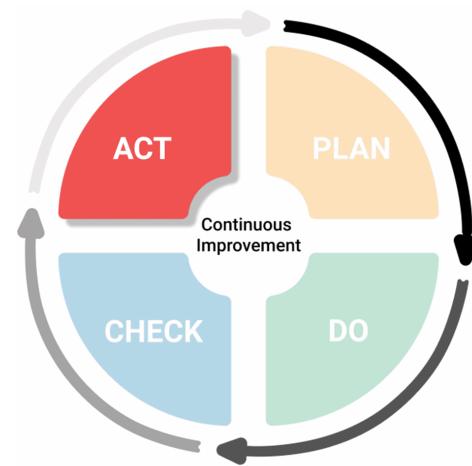


TPS - Toyota Production System (1930 – 1980)

- > Previously a textile company, Toyota moved into building automobiles in 1934.
- > Kiichiro Toyoda, founder of Toyota Motor Corporation, directed the engine casting work and discovered many problems in their manufacturing, with wasted resources on repair of poor-quality castings. Toyota engaged in intense study of each stage of the process.
- > In 1936, when Toyota won its first truck contract with the Japanese government, the processes encountered new problems, to which Toyota responded by developing the "Kaizen" methodology.
- > In the postwar economy, the demands were low. The previous "Sales target / production cost production driving" needed to be replaced by "Actual sales production driving" - Push Model to Pull Model
- > Then, in the 60's and 7's, the TPS - Toyota production system developed its thought school further and emerge with the Kanban methodology.

Kaizen

- > Making changes and monitoring results, then adjusting.
 - + Large-scale pre-planning and extensive project scheduling are replaced by smaller experiments, which can be rapidly adapted as new improvements are suggested.
- > Kaizen cycle : Plan -> do -> Check -> Act (adapt)
- > Small steps improvements approach
- > Kaizen blitz (event) : experiment or task force to address a particular issue over the course of a (few) week(s)
- > Humanized approach to workers and to increasing productivity (human centric)
- > Root cause Analysis - 5 why's
- > Small groups
- > Standardization culture
- > “*You can't manage what you can't measure*” (production KPIs)



Kanban

- > Flow oriented model (vs. task oriented model)
- > From Push model to Pull Model
- > Traditional production relies on heavy and long process requiring a push approach and very strong demand-forcasting models. Kanban, by contrast, is part of an approach where the pull comes from demand and products are made to order. Requires short lead times and strong production processes.
- > “*You can't manage what you can't visualize*”



Lean Movement (1990)

TPS is translated into "Lean" 1988 by John Krafcik

John Krafcik coined the term "Lean" in his 1988 article, "Triumph of the Lean Production System".

The article states:

- (a) Lean manufacturing plants have higher levels of productivity/quality than non-Lean and
- (b) "The level of plant technology seems to have little effect on operating performance".

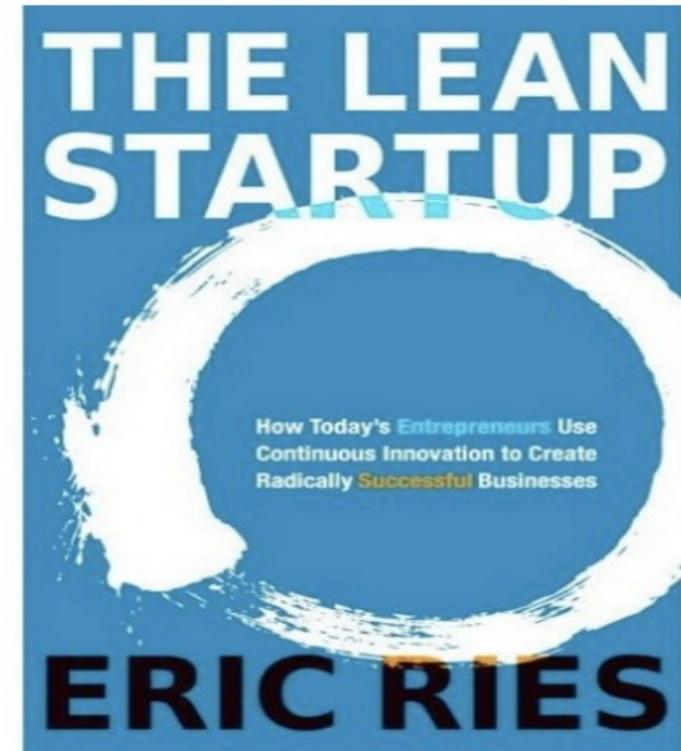
He established the foundation aimed at applying Lean and Agility throughout the Industry.



- > **Lean thinking** is a **business methodology** that aims to provide a new way to think about how to organize human activities to deliver more benefits to society and value to individuals while **eliminating waste**.
- > Lean thinking is a **new way of thinking any activity** and seeing the waste inadvertently generated by the way the process is organized
- > The aim of lean thinking is to create a **lean enterprise**, one that **sustains growth** by aligning customer satisfaction with employee satisfaction, and that **offers innovative products** or services profitably while **minimizing unnecessary over-costs** to customers, suppliers and the environment.
- > The Lean Movement finds its roots in Toyotism and values **performance** and **continuous improvement**.

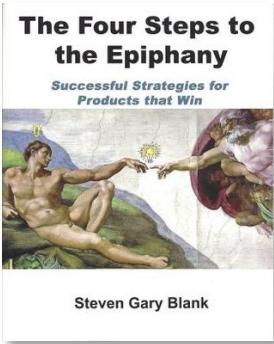
“Apply
Lean Thinking
to
Innovation
Process”

(2011)



- > **Just in Time!**
 - + Only implement minimum requirements, only at the time it is actually required!
 - + Pull model
- > **Measure driven**
 - + Each and every implementation is measurable and measured
 - + Don't believe, know !
 - + **Make measurable predictions !**
 - o An action whose effect cannot be measured is useless.
- > **Speed up development cycles**
 - + Deploy and implement all quality practices (XP, Agile, DevOps) to enable development cycles to be as short as possible

Lean Startup

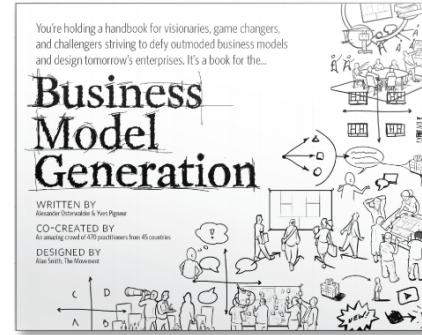
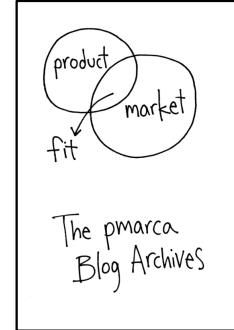


Steve Blank

2005

2007

Marc Andreessen
(Andy Rachleff)

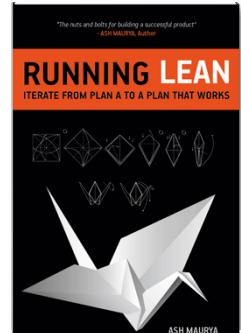
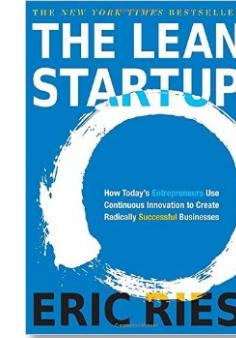


Alex Osterwalder

2011

2011

Eric Ries



Ash Maurya

Key Principles

> Entrepreneurs are everywhere

- + You don't have to work in a garage to be in a startup. The concept of entrepreneurship includes anyone who works within Eric Ries' definition of a startup :
 - o **A startup is a human institution designed to create new products and services under conditions of extreme uncertainty.**
- + That means entrepreneurs are everywhere and the Lean Startup approach can work in any size company, even a very large enterprise, in any sector or industry.

> Entrepreneurship is management

- + A startup is an institution, not just a product, and so it requires a new kind of management specifically geared to its context of extreme uncertainty.
In fact, Ries believes "*entrepreneur*" should be considered a job title in all modern companies that depend on innovation for their future growth

> Validated learnings

- + Startups exist not just to make stuff, make money, or even serve customers. They exist to learn how to build a sustainable business. This learning can be validated scientifically by running frequent experiments that allow entrepreneurs to test each element of their vision.

> Innovation accounting

- + To improve entrepreneurial outcomes and hold innovators accountable, we need to focus on the boring stuff: how to measure progress, how to set up milestones, and how to prioritize work. This requires a new kind of accounting designed for startups- and the people who hold them accountable.

> Build-Measure-Learn

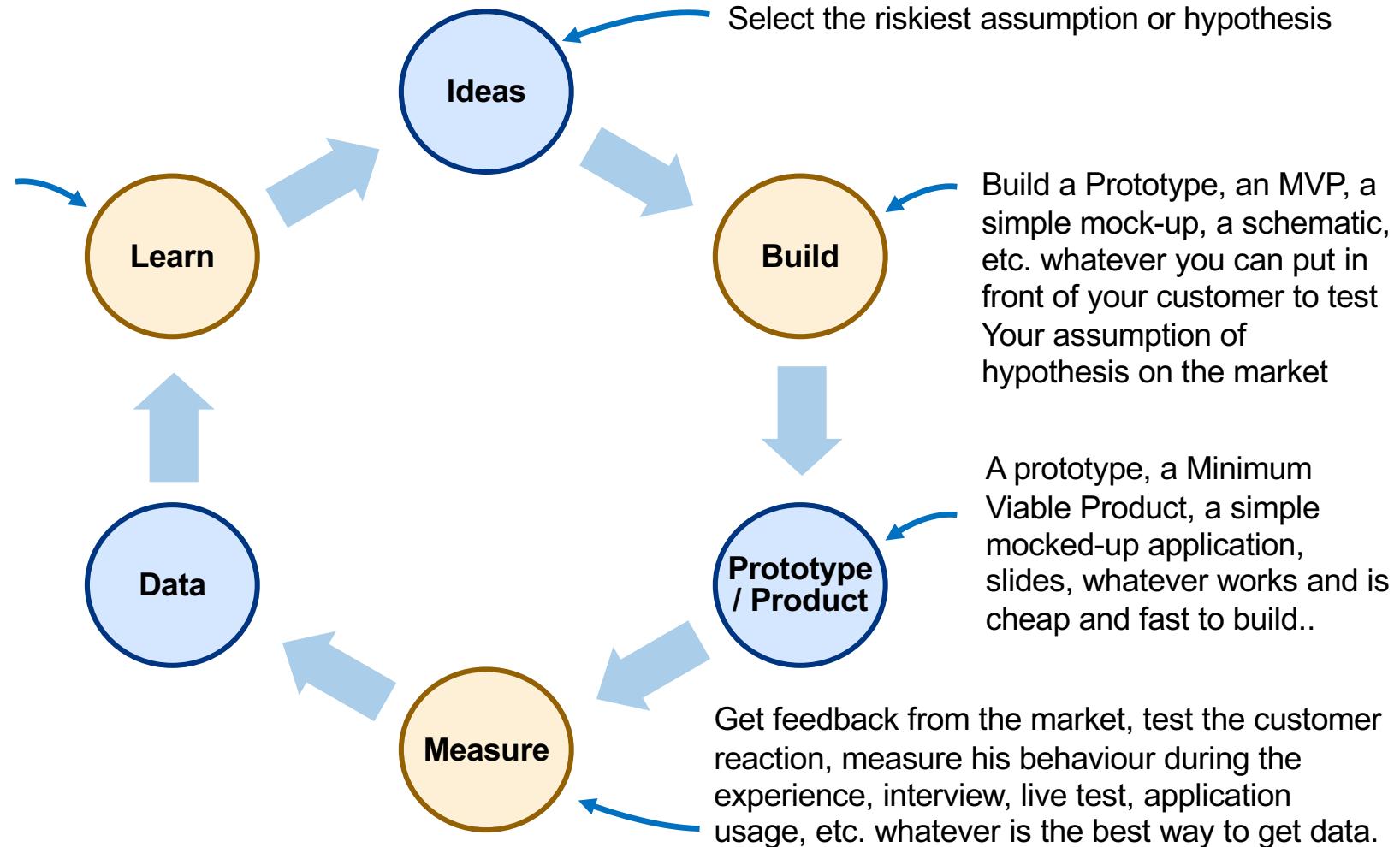
- + The fundamental activity of a startup is to turn ideas into products, measure how customers respond, and then learn whether to **pivot or persevere**. All successful startup processes should be geared to accelerate that **feedback loop**.

The feedback loop

Understand the feedback and learn if your assumption is confirmed or contradicted, if your hypothesis is valid or wrong.

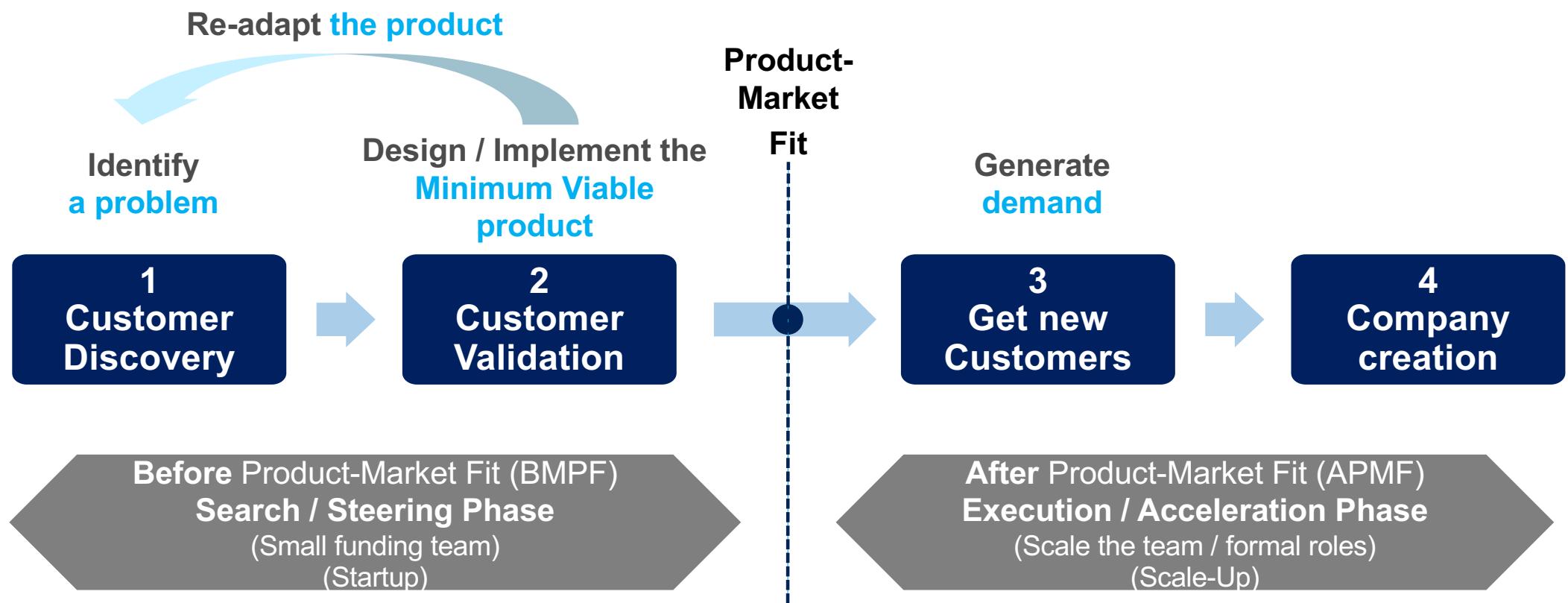
If it is confirmed, move on to the next assumption or hypothesis to be tested.

If it is contradicted, review your hypothesis, adapt your plan and come up with something different.



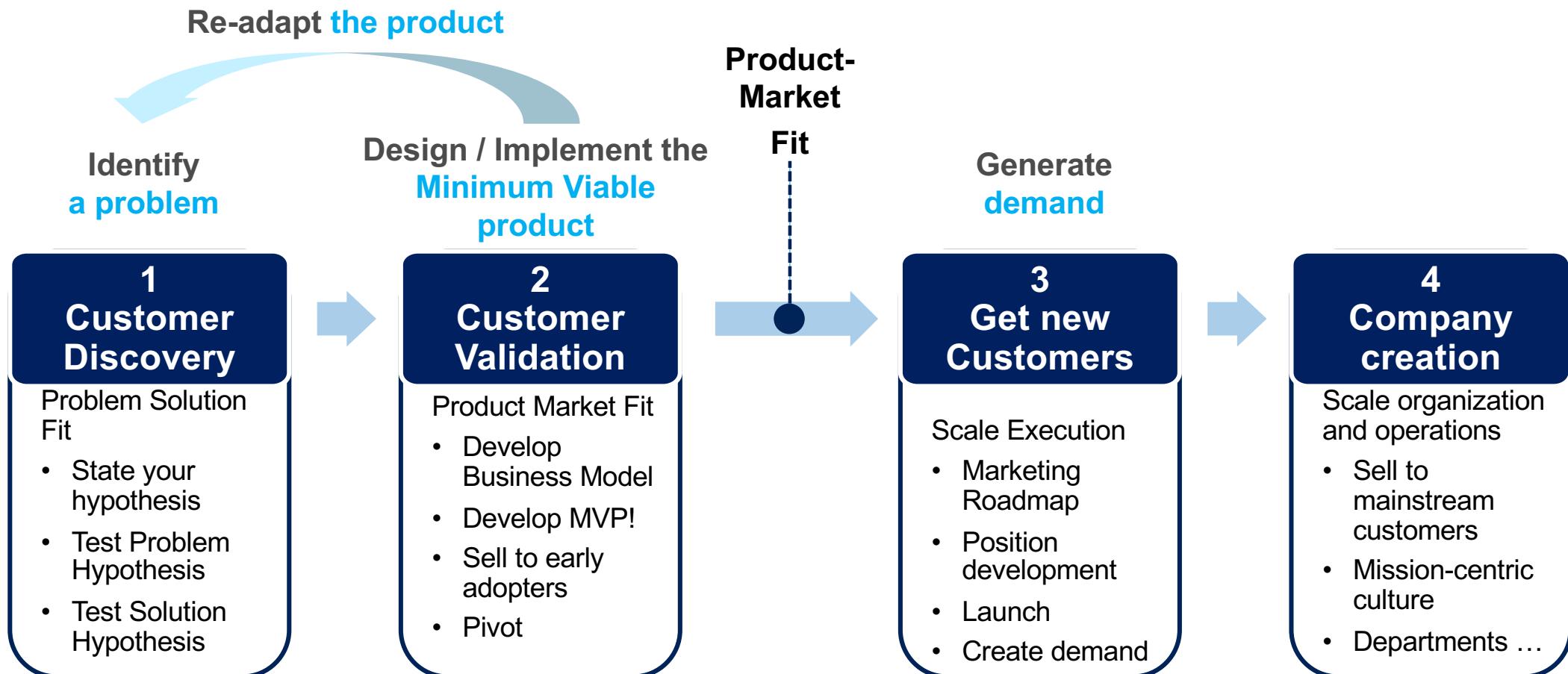
Four steps to Epiphany: customer development

Steve Blank studied the inherent patterns empirically validated in successful and failed startups and offers key insights to improve the probability of finding Product-Market Fit



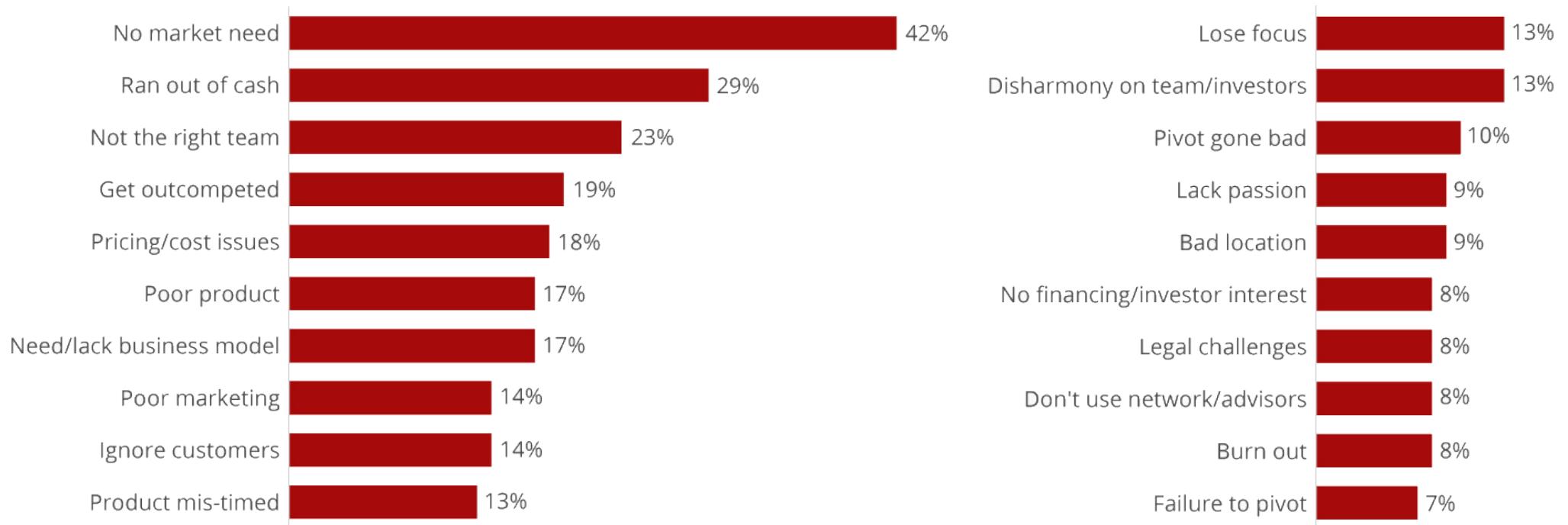
Source : Steve Blank (various blog posts)

Four steps to Epiphany: customer development



The top reasons why startup fail

Most frequently cited reasons for startup failures



Source : CB Insights - <https://www.statista.com/chart/11690/the-top-reasons-startups-fail/>

- > **Customer Development instead of Product Development**
 - + More startup fail from a *lack of customers* rather than from a failure of Product Development.
- > The **Customer Development model** delineates all the customer-related activities in the early stage of a company into their own processes and groups them into four easy-to-understand steps
- > It focuses on developing customers for the product or service your startup is building. Customer Development is eventually just as much about **finding a product for your market** than finding a market for your product

2. The 4 steps to the epiphany



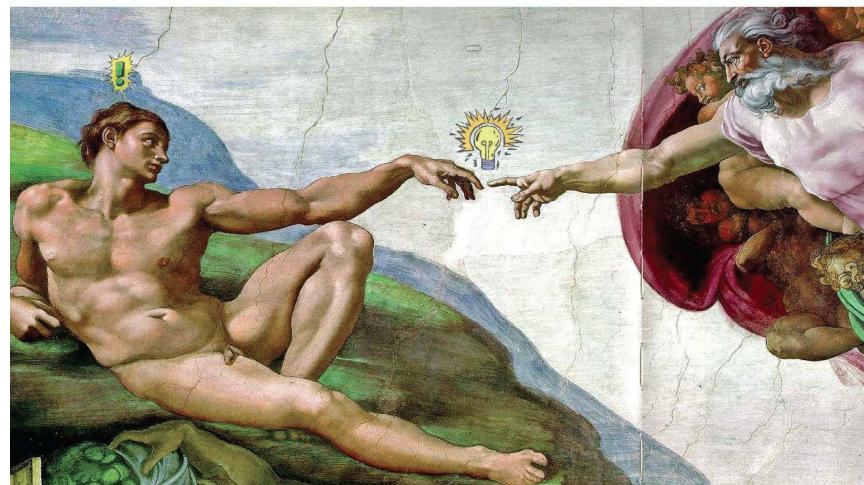
The Path to Disaster: The Product Development Model

- > The traditional product development model has four stages:
 1. concept/seed,
 2. product development,
 3. beta test,
 4. and launch.
- > That product development model, when applied to startups, suffers from a lot of flaws.
They basically boil down to:
 - + Customers were nowhere in that flow chart
 - + The flow chart was strictly linear
 - + Emphasis on execution over learning
 - + Lack of meaningful milestones for sales/marketing
 - + Treating all startups alike

The Path to Epiphany: The Customer Development Model

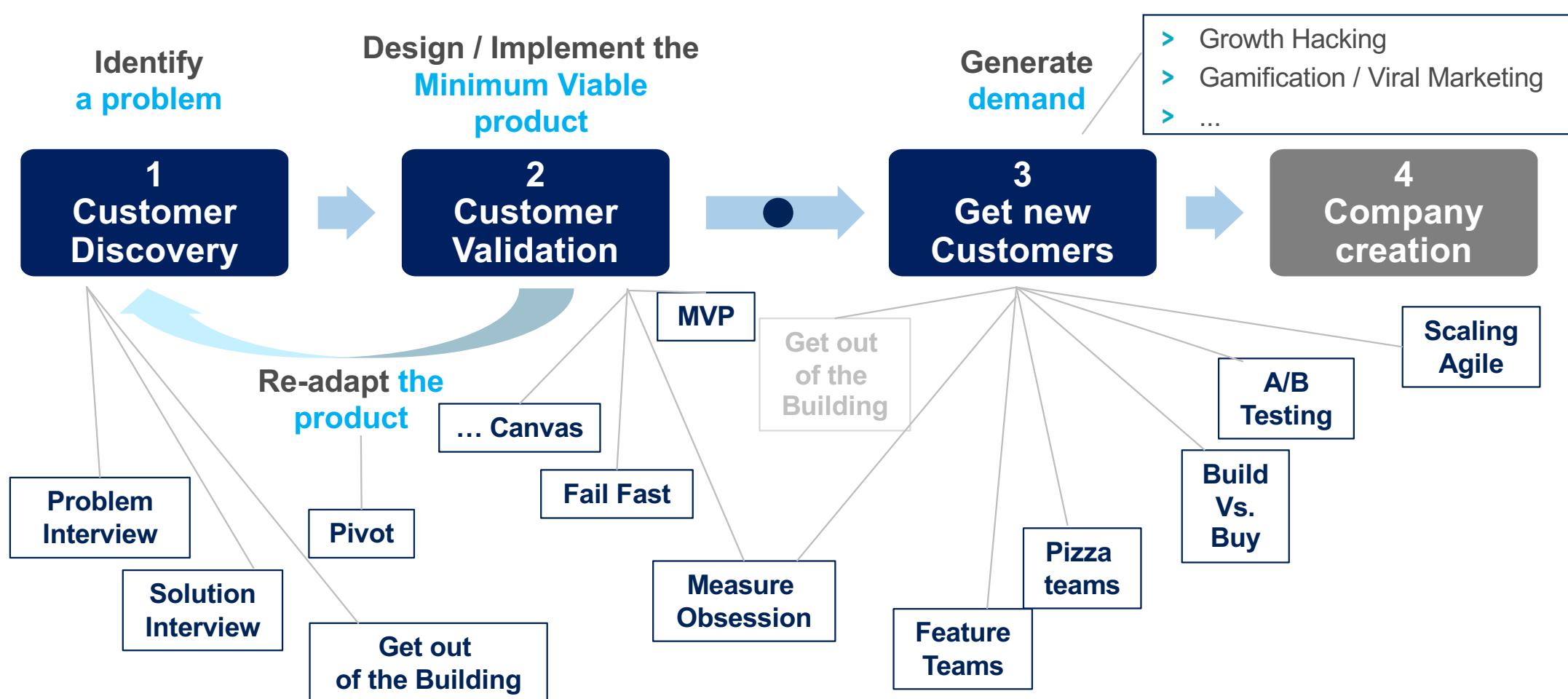
The Four Steps to the Epiphany

***Successful Strategies for
Products that Win***

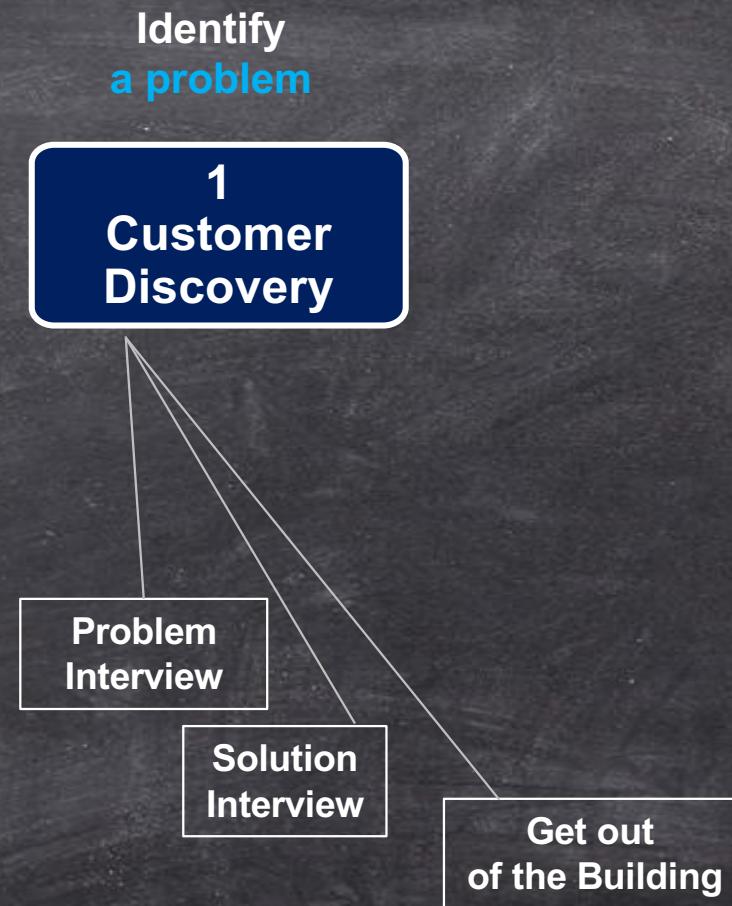


Steve Blank

A focus on practices



2.1 – Customer Delivery





Get out of the Building !

- > “If you’re not Getting out of the Building, you’re not doing Customer Development and Lean Startups”

- > Pre-Product/Market Fit :
“one single good customer development interview is better for learning about your customers / product / problem / solution / market than five surveys with 10,000 statistically significant responses.”

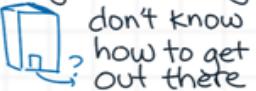
“There are no facts inside the building. **Get out of the building.**” - Steve Blank



10 ideas

#getoutofthebuilding

Common challenge: validating your assumptions 1 on 1



Ex: Kodable

- Bring people in
- Oh, these kids don't have the fine motor skills...

opportunity
Cost:
your next
best idea

① Don't ask your uncle.

Good job
You rock
It's a winner

Friends & Family

② Set up a booth,
do a public demo

Ex: Rooibie
Red Tea

③ Interview potential
customers



④ Put your office where
your customers are
ex: Over

⑤ Throw a
party

find testers,
prospects
RRR

⑥ Talk to experts

⑦ Find decision-makers



⑧ Listen to demand



⑨ Pre-order/landing/analytics

⑩ Ask for the
introduction

Problem Interview

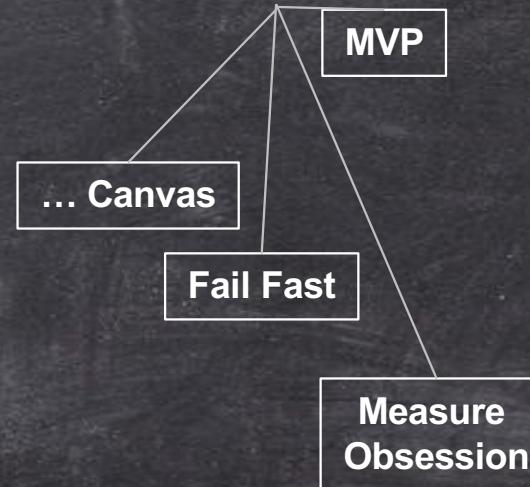




2.2 – Customer Validation

Design / Implement the
Minimum Viable
product

2 Customer Validation



The Lean Canvas

Product		Solution	Market
PROBLEM List your top 1-3 problems 1	SOLUTION Outline a possible solution for each problem. Top 1-3 valuable features or even epics 3	UNIQUE VALUE PROPOSITION Single, clear, compelling message that states why your solution is better, more valuable, different in comparison to competitors 2	UNFAIR ADVANTAGE Something that cannot easily be bought or copied 7
EXISTING ALTERNATIVES List how these problems are solved today	KEY METRICS What does success look like ? How to measure ? List the key numbers that tell you how your business is doing 6	HIGH-LEVEL CONCEPT List your X for Y analogy, e.g. Youtube=Flickr for videos	CHANNELS List your path to customers (inbound or outbound) 4
COST STRUCTURE List your fixed and variable costs - Software licensing - FTEs		REVENUE STREAMS List your sources of revenue - revenue model - lifetime value 5	CUSTOMER SEGMENTS List your target customers and users Should go even further: list the main 3 to 6 personas we'll be targeting 1

THE LEAN CANVAS

For new products, the initial battle is about how to get noticed ? How will you get the customer's attention ?

Limits and Difficulties

PROBLEM

List your top 1-3 problems

SOLUTION

Outline a possible solution for each problem.
Top 1-3 valuable features or even epics

UNIQUE VALUE PROPOSITION

One of the riskiest item on your canvas !
Start testing from day 1 !

KEY METRICS

What does success look like ? How to measure ?
List the key numbers that tell you how your business is doing

- Social networks
- Newsletter
- Ads
- Friends
- Events
- SEO
- Etc.

DEL)

Distinguish between customers and users, customers pay you, users don't.

UNFAIR ADVANTAGE

Something that cannot easily be bought or copied

CUSTOMER SEGMENTS

Early adopters will help you test and refine your product AND BUSINESS MODEL

Examples:

- Insider Information
- Personal authority
- A dream team
- Existing customers
- "Right" celebrity endorsement
- Large network effect
- Community
- SEO ranking
- Patents
- Core values

RE

List your fixed and variable costs
- Software licensing
- FTEs

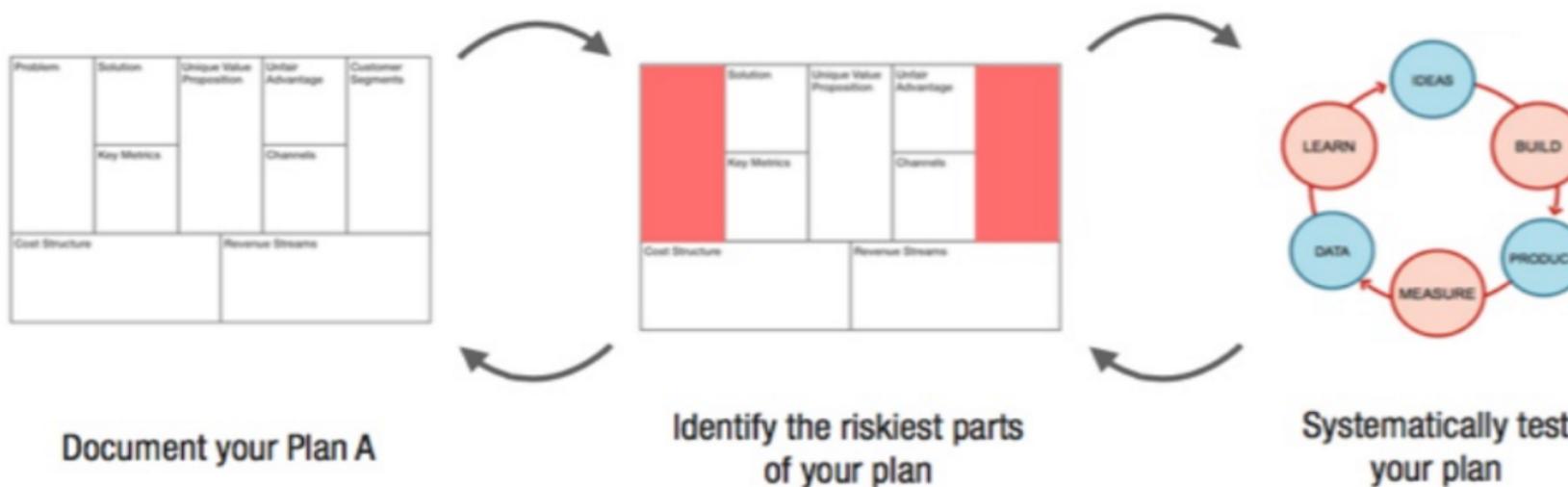
VENUE STREAMS

Calculate your break-even point

Sources of revenue

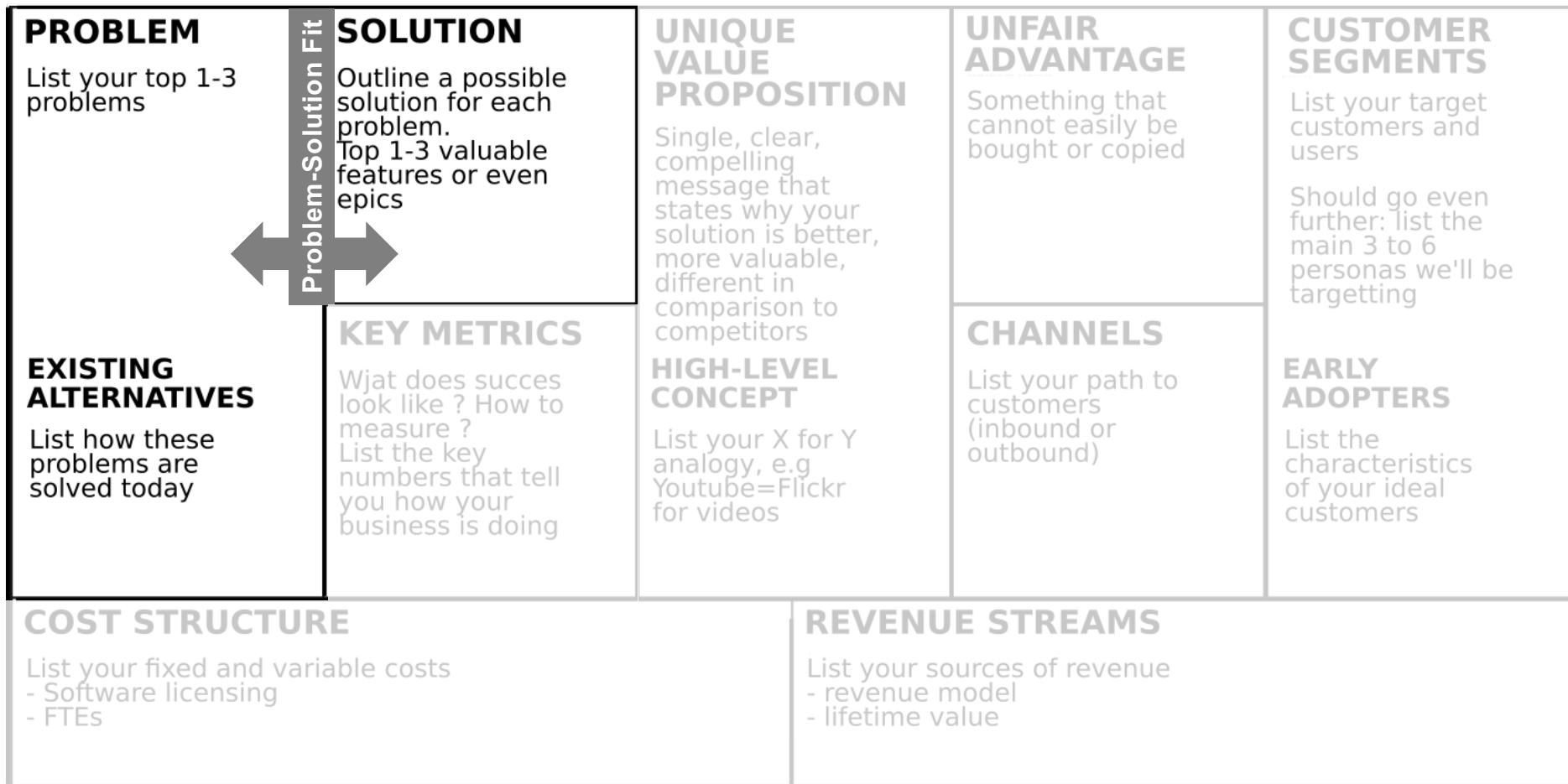
Test your plan !

- > **Lean Startup : test your plan !**
- > Using the "Build - Measure - Learn" diagram, the question then becomes, "What hypotheses should I test?".
 - + This is precisely the purpose of the initial Lean Canvas,



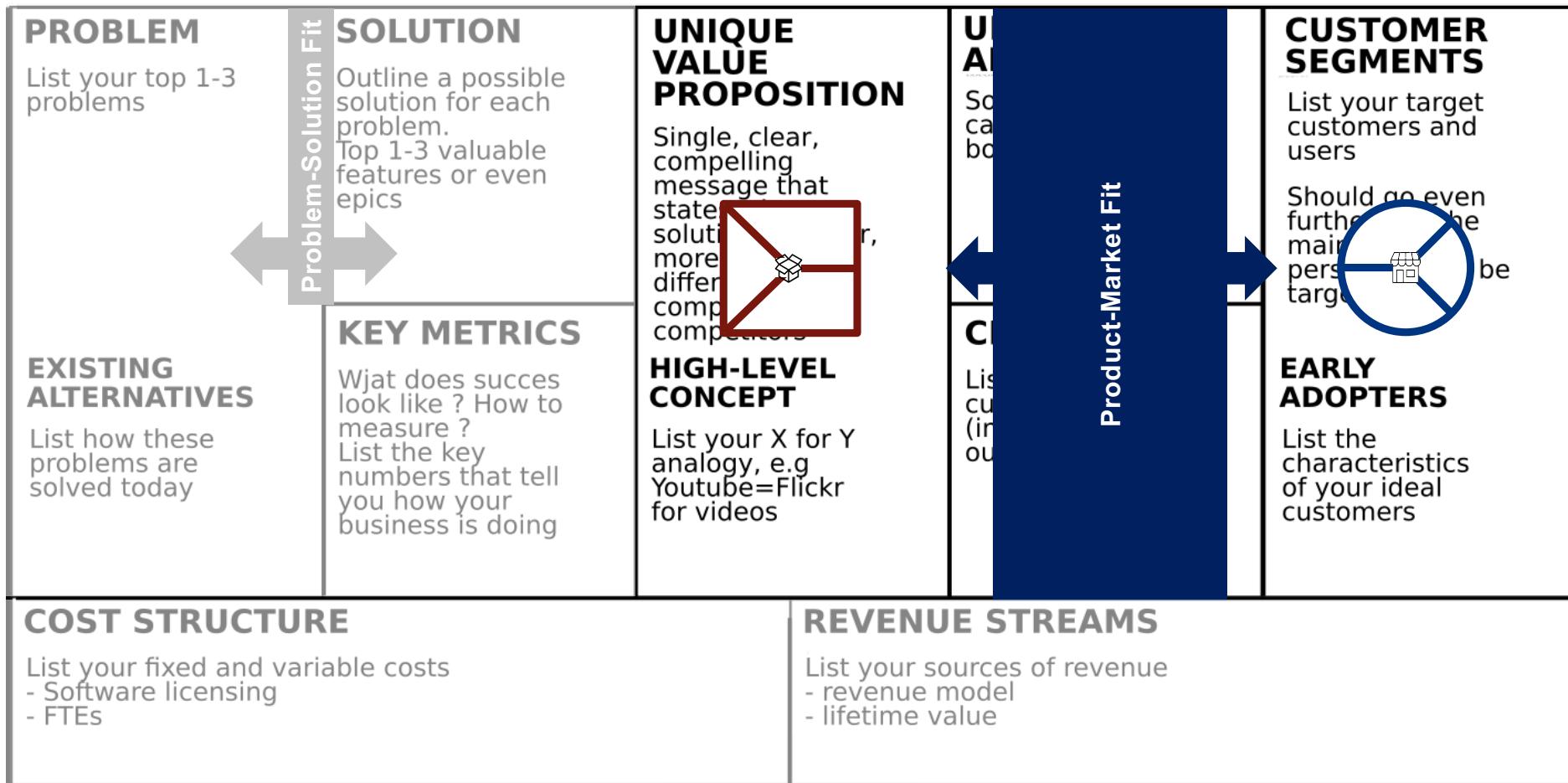
Problem Solution Fit

THE LEAN CANVAS (BUSINESS MODEL)

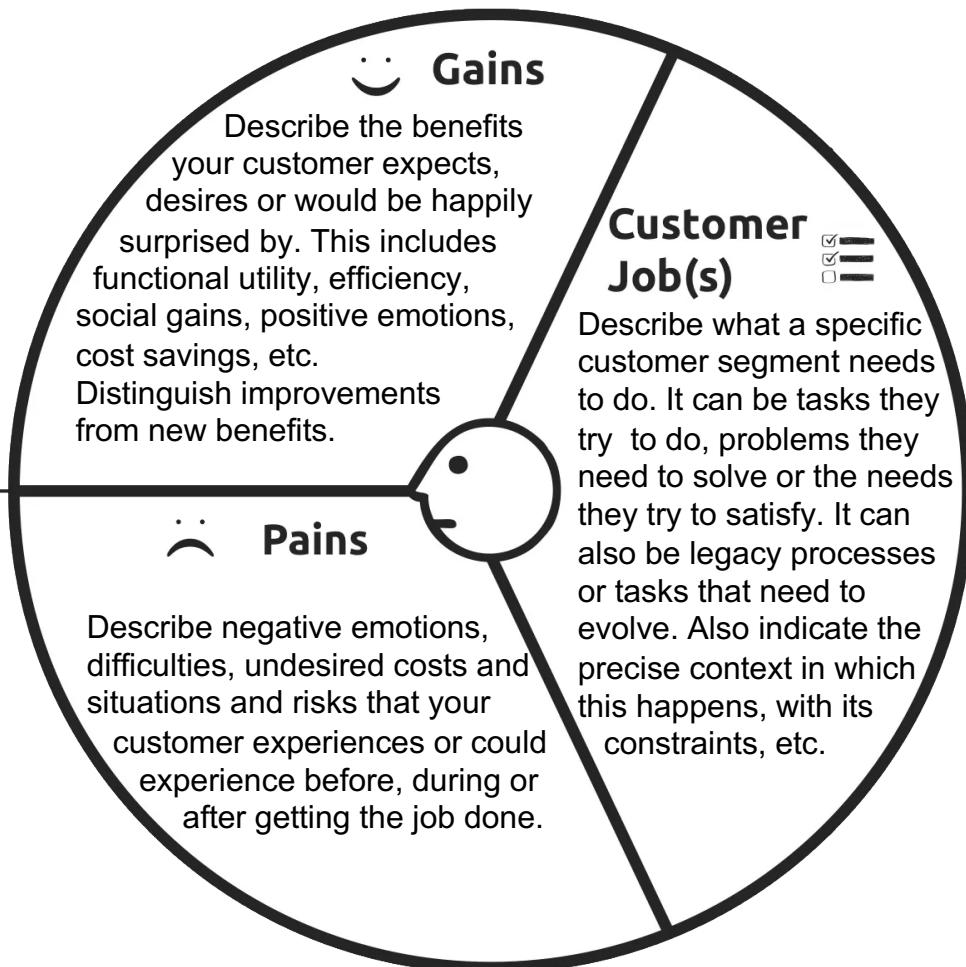
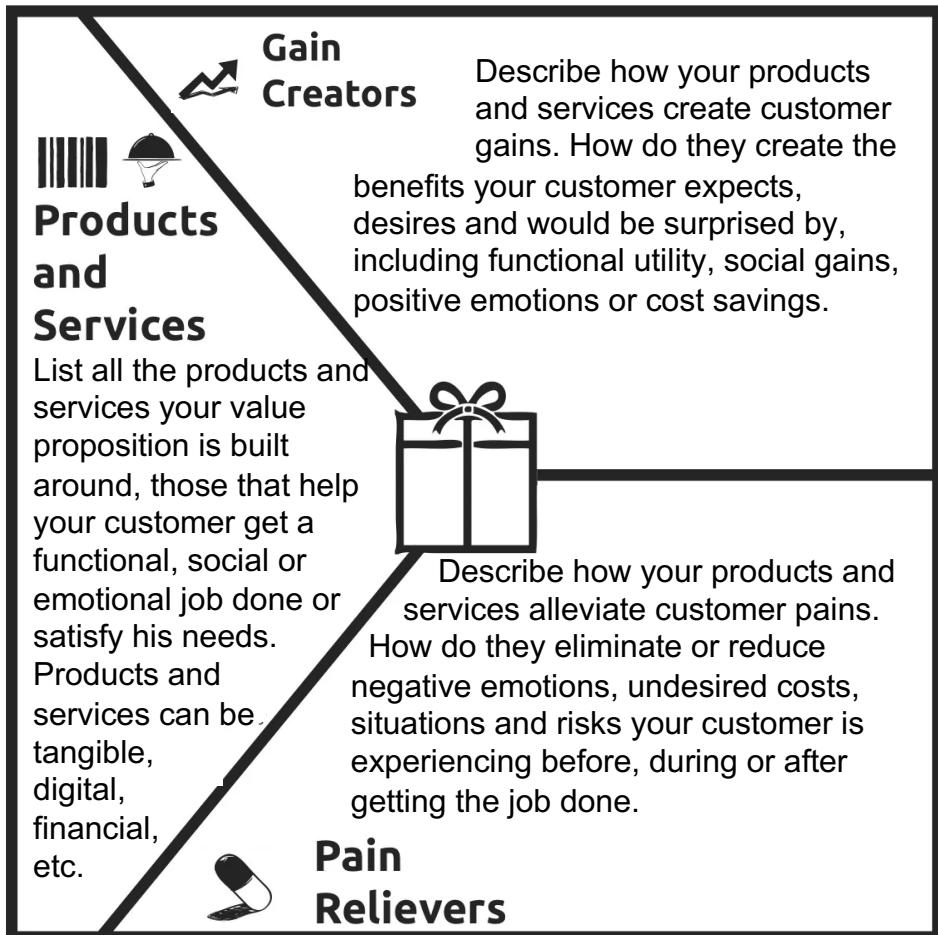


Product Market Fit on the Lean Canvas

THE LEAN CANVAS (BUSINESS MODEL)



The Value Proposition Canvas

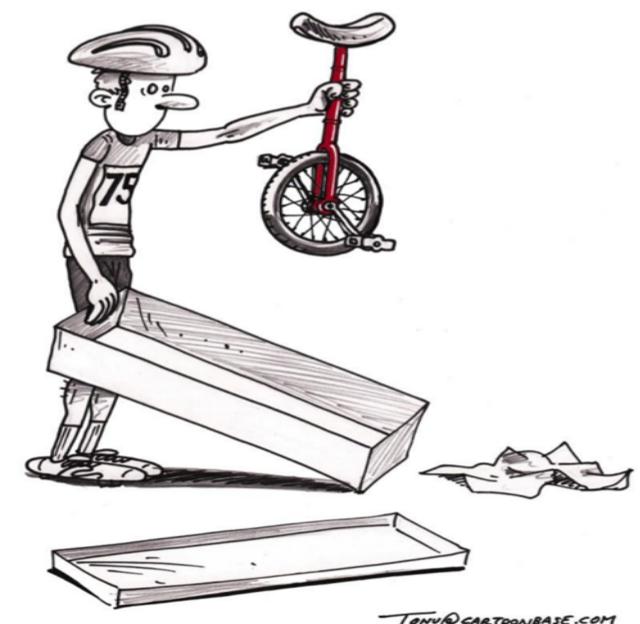


- > The **Minimum Viable Product** is an engineering product with just the set of features required to gather *validated learnings* about it - or some of its features / aspects - and its continuous development.
 - + The *Minimum Viable Product* should have just that **set of initial features** strictly required to have a valid product, usable for its **very initial intent**, and nothing more.
 - + In addition these features should be as minimalist as possible but without compromising the overall *User Experience*. A car should move, a balloon should be round and bounce, etc.
- > Eric Ries defines the MVP as:

"The Minimum Viable Product is that version of a new product a team uses to collect the maximum amount of validated learning about customers with the least effort."

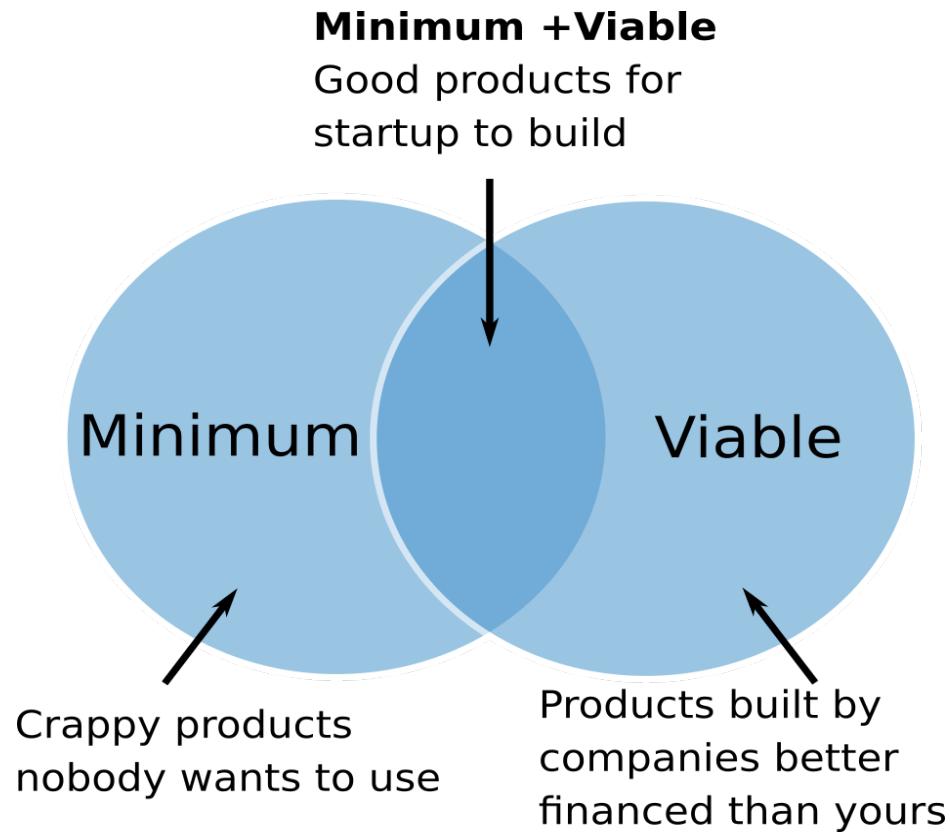
(Source : Les pratiques des géants du web / Stephen Perrin – OCTO Technology –
<http://fr.slideshare.net/sperin/les-pratiques-des-geants-du-web>)

MINIMUM VIABLE PRODUCT



Tony@CARTOONBASE.COM

Minimum and Viable ?



HOW NOT TO BUILD A MINIMUM VIABLE PRODUCT



1



2



3



4

ALSO HOW NOT TO BUILD A MINIMUM VIABLE PRODUCT



1



2



3

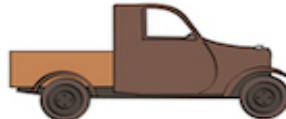


4

HOW TO BUILD A MINIMUM VIABLE PRODUCT



1



2

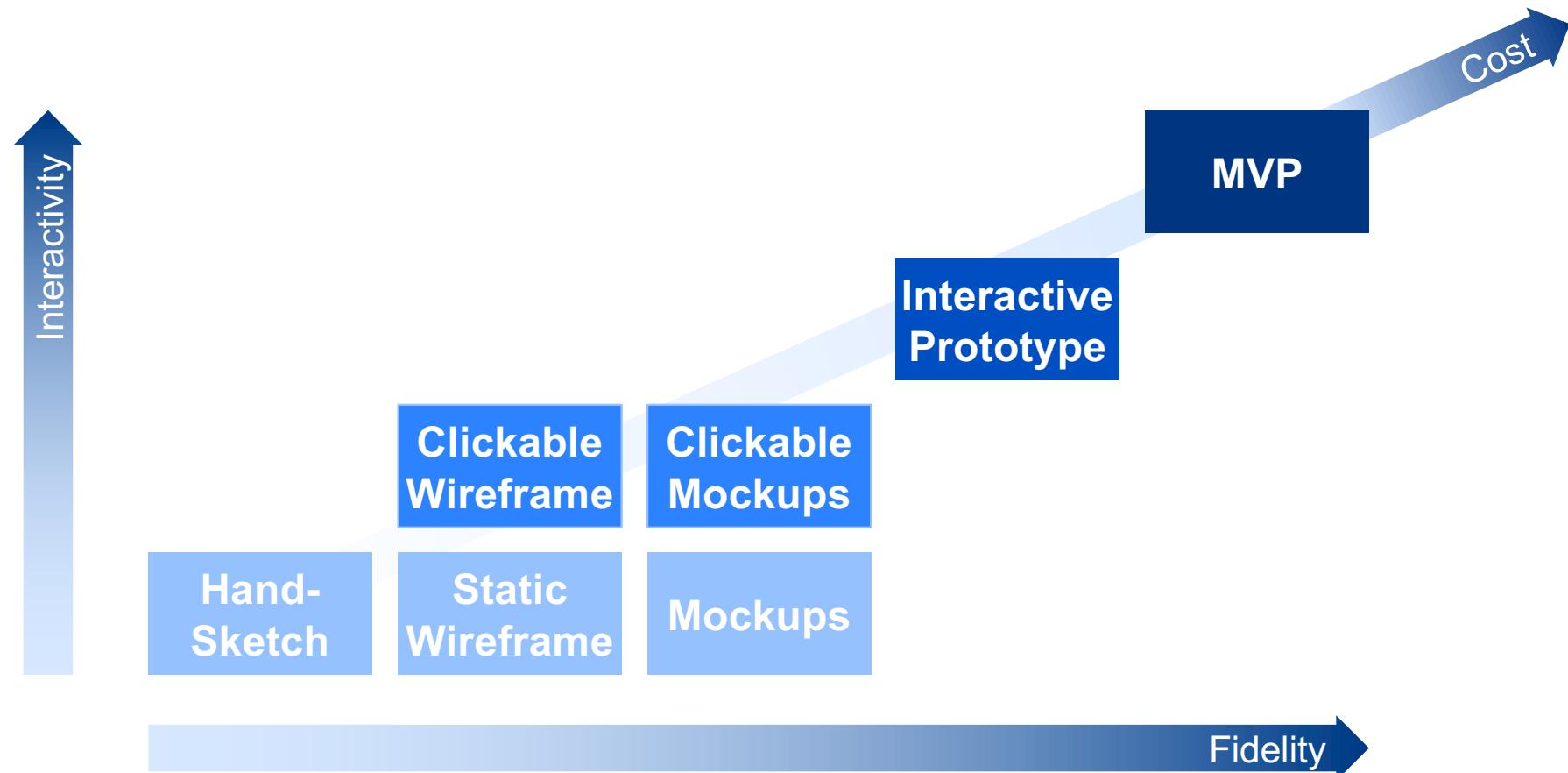


3



4

Sidenote : Product Design artifacts



Source : <https://www.slideshare.net/LeanStartupConf/a-playbook-for-achieving-productmarket-fit>

Fail Fast

- > **Fail fast means getting out of planning mode and into testing mode**, eventually for every component, every single feature, every idea around your product or model of change and drop things that are not working as fast and cheap as possible.
- > *It's totally OK to fail, as long as you fail fast.*

Fail fast,
Learn faster,
Succeed sooner !

Success

Success



what people think
it looks like

what it really
looks like

- > So how do you know when to turn, when to drop an approach and adapt your solution?
How can you know it's not too soon?
Measure, measure, measure of course!
- > (Fail fast ≠ Fail often)

Repeat a thousand times:

I will never again say
“I think that ...”

but always

“I will measure that ...”

“In god we trust, all others must bring data “



(Source : Les pratiques des géants du web / Stephen Perrin – OCTO Technology – <http://fr.slideshare.net/sperin/les-pratiques-des-geants-du-web>)

How to choose good metrics ?

- > Honestly there is no magic silver bullet and it can in fact be pretty difficult to pick up the right metric that would be most helpful to validate a certain hypothesis.
However, metrics should at all cost respect the three A's. Good metrics
 - + are **actionable**,
 - + can be **audited**
 - + are **accessible**
- > An **actionable metric** is one that ties specific and repeatable actions to observed results.
The *actionable* property of picked up metrics is important since it prevents the entrepreneur from distorting the reality to his own vision.
 - + We speak of *Actionable vs. Vanity Metrics*.
 - + Meaningless metrics such as "How many visitors ?", "How many followers ?" are vanity metrics and are useless.
- > Ultimately, your metrics should be useful to **measure progress against your own questions**.



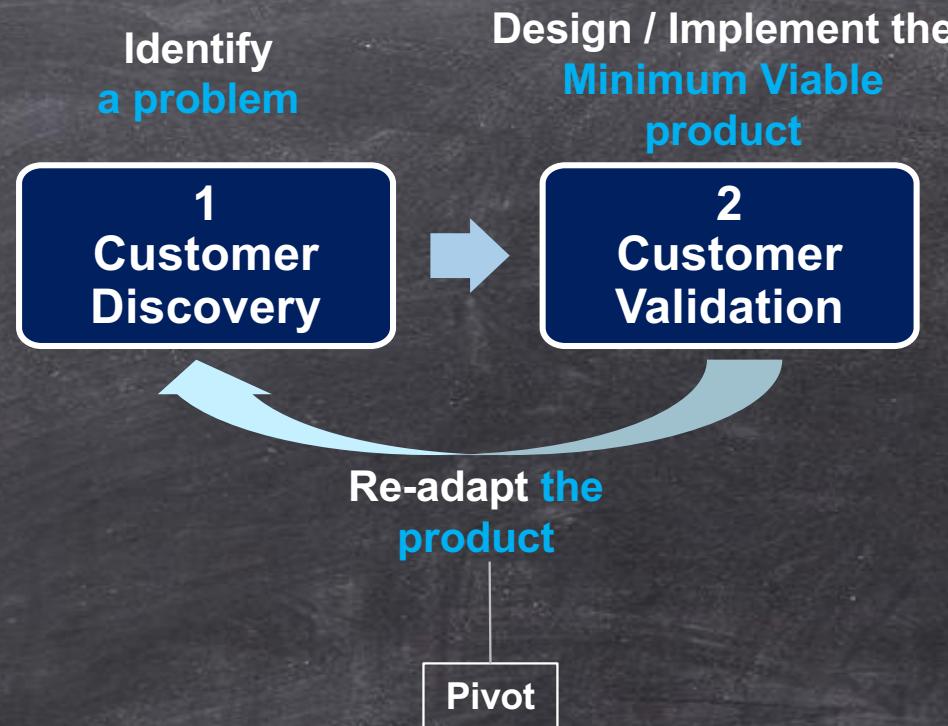
Some examples

1. NPS - Net Promoter Score
2. CLV to CAC Ratio
3. Retention Ratio
4. Growth Rate

There are many others!

(Hint : search for “Pirate Metrics” by Dave McClure)

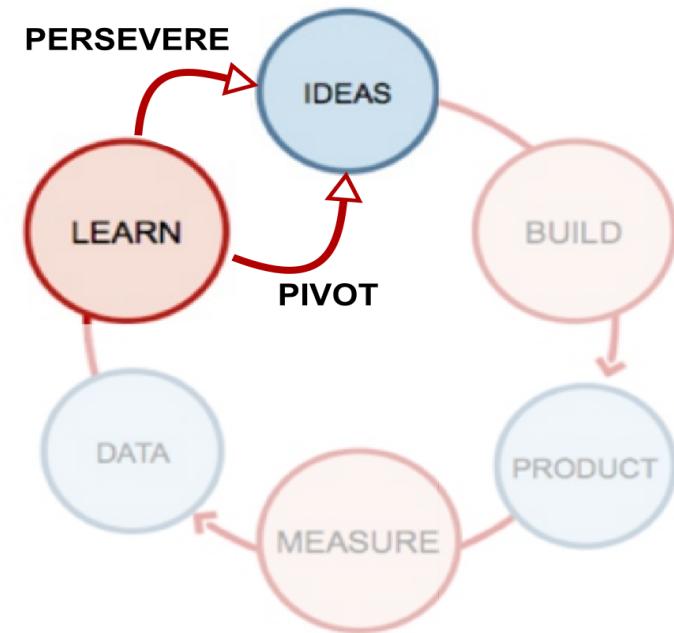
2.3 - Re-Adapt the product



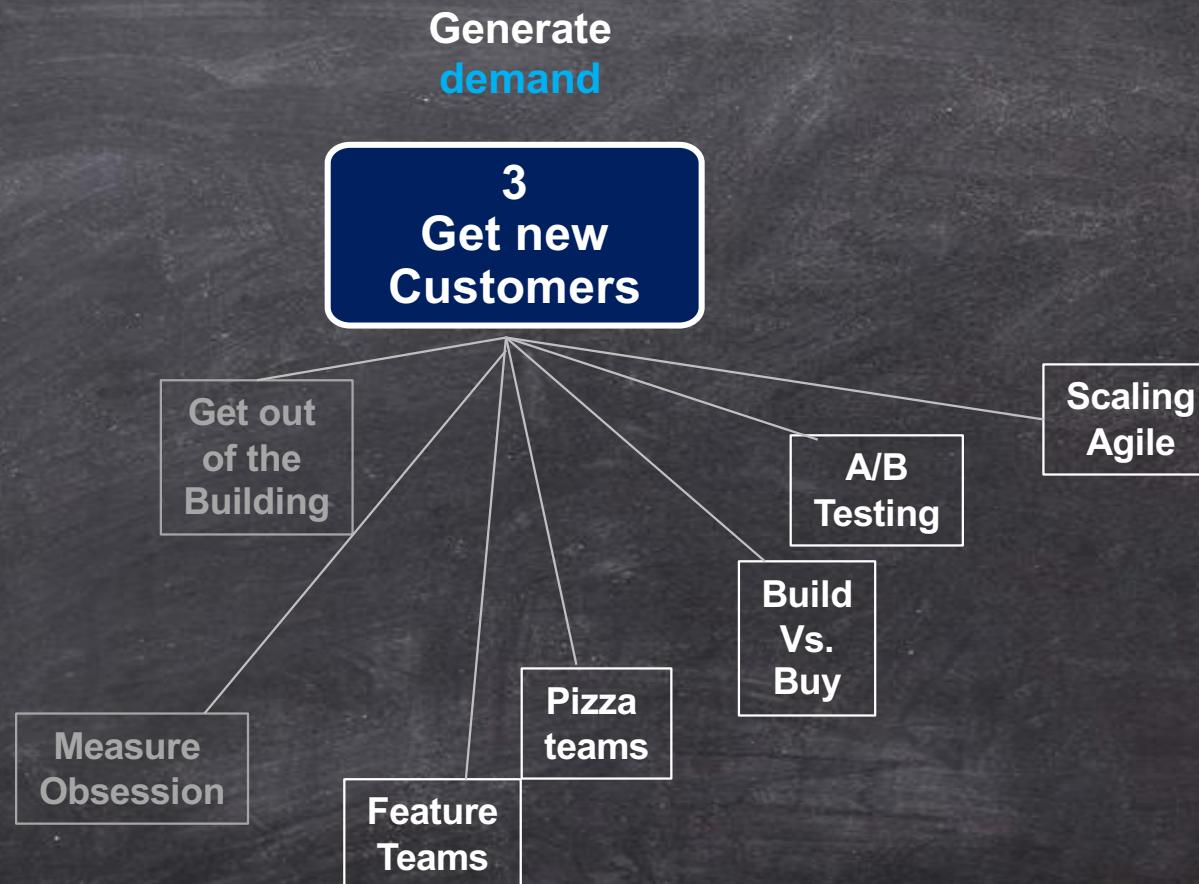
“A startup **pivots** when
it **changes its product**
to answer a
new or different need
identified by
customer / market feedback
in its learning (search) phase”

Pivot or Persevere

- > When the startup changes its product to meet either this new need or the former need in a different way, it is said to have performed a **Pivot**. A startup can *pivot* several times during its existence.
- > A *pivot* is ultimately a **change in strategy** without a *change in vision*. It is defined as a structured course correction designed to **test a new fundamental hypothesis** about the product, business model and engine of growth.



2.4 – Get New Customers



Small Teams

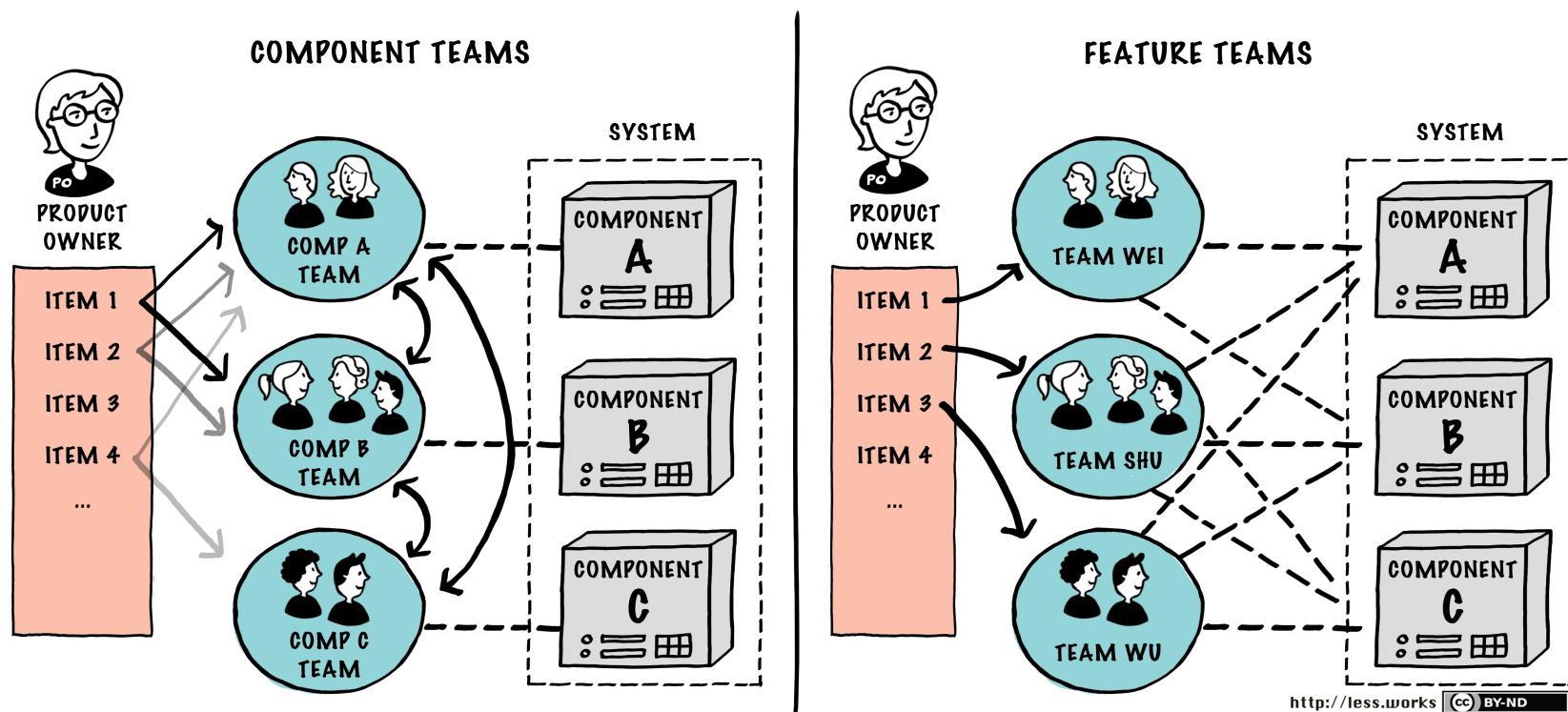


As team size grows, **the amount of one-on-one communication channels tend to explode**, following the formula to compute number of links between people which is $n(n - 1)/2$.
This is $O(n^2)$ (Hello Engineers) and is really a *combinatorial explosion*.



What size for a team ?

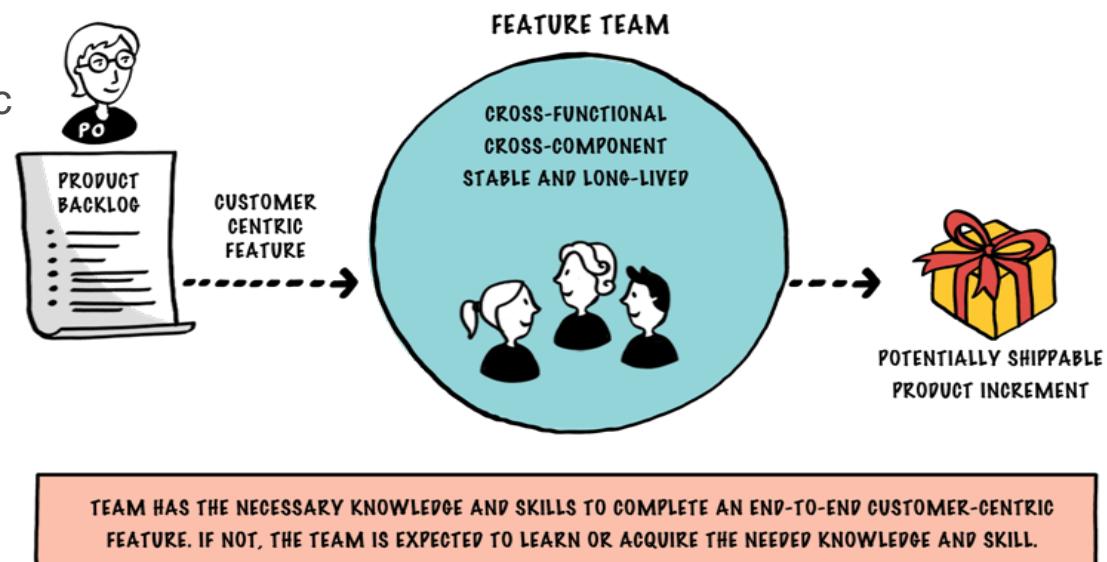
Introducing Feature Teams



(Source : Large Scale Scrum : <http://less.works/>)

➤ The characteristics of a feature team are :

- **long-lived**—the team stays together so that they can ‘jell’ for higher performance; they take on new features over time
- **cross-functional and cross-component**
- ideally, **co-located**
- work on a complete customer-centric feature, across all components and disciplines (analysis, programming, testing, ...)
- composed of generalizing specialists
- in Scrum, **typically 7 ± 2 people**



(Source : Large Scale Scrum : <http://less.works/>)



Component Teams vs Feature Teams

Component Team	Feature Team
optimized for delivering the maximum number of lines of code	optimized for delivering the maximum customer value
focus on increased individual productivity by implementing 'easy' lower-value features	focus on high-value features and system productivity (value throughput)
responsible for only part of a customer-centric feature	responsible for complete customer-centric feature
traditional way of organizing teams — follows Conway's law	'modern' way of organizing teams — avoids Conway's law
leads to 'invented' work and a forever-growing organization	leads to customer focus, visibility, and smaller organizations
dependencies between teams leads to additional planning	minimizes dependencies between teams to increase flexibility
focus on single specialization	focus on multiple specializations
individual/team code ownership	shared product code ownership
clear individual responsibilities	shared team responsibilities
results in 'waterfall' development	supports iterative development
exploits existing expertise; lower level of learning new skills	exploits flexibility; continuous and broad learning
works with sloppy engineering practices—effects are localized	requires skilled engineering practices—effects are broadly visible
contrary to belief, often leads to low-quality code in component	provides a motivation to make code easy to maintain and test
seemingly easy to implement	seemingly difficult to implement

(Source : Large Scale Scrum : <http://less.works/>)

The Spotify Model

Squads

- > Equivalent to a Scrum Team
- > Autonomous as possible

Tribes

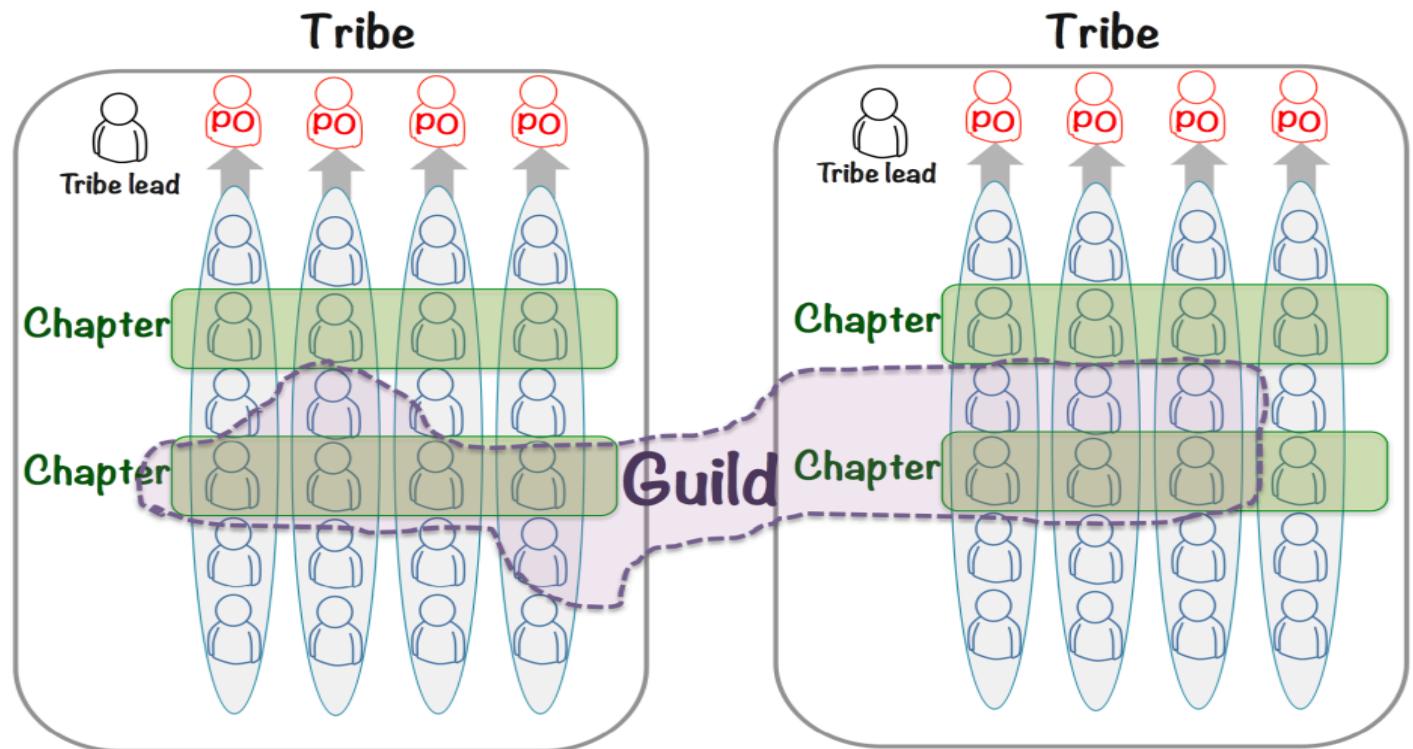
- > Same office < 100 FTE
- > Common area of the system
- > Organized for minimum interdependency

Chapters

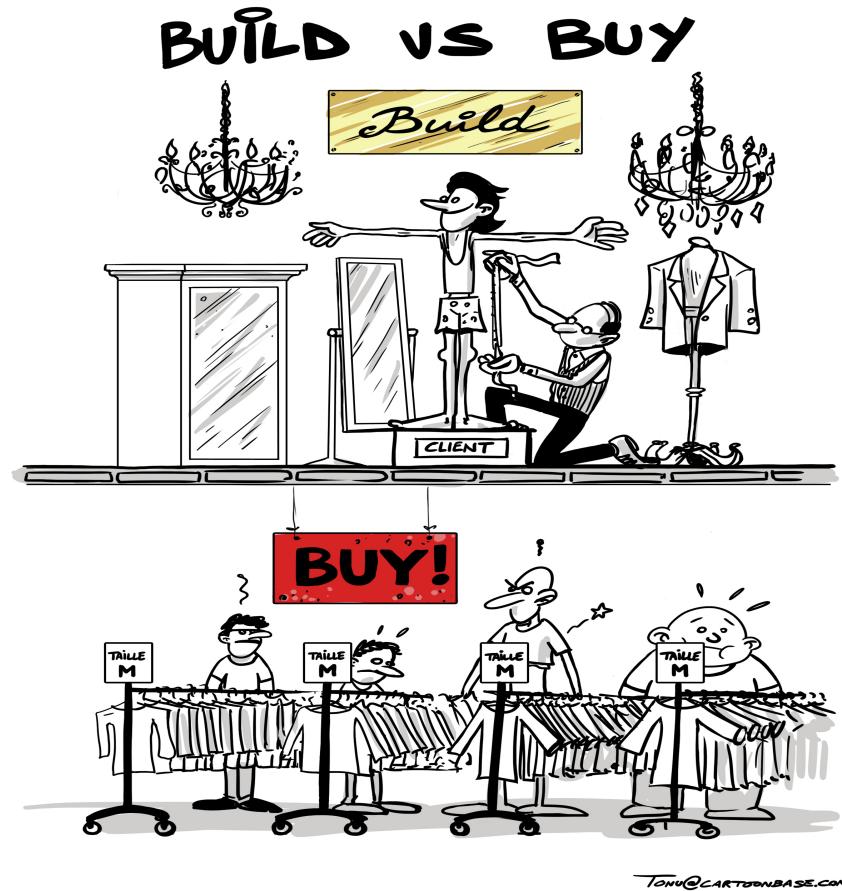
- > Skills community (community of practices)
- > Chapter lead is line manager

Guilds

- > Community of interest
- > Cross tribe groups
- > Guild unconferences



Build vs Buy (1/2)

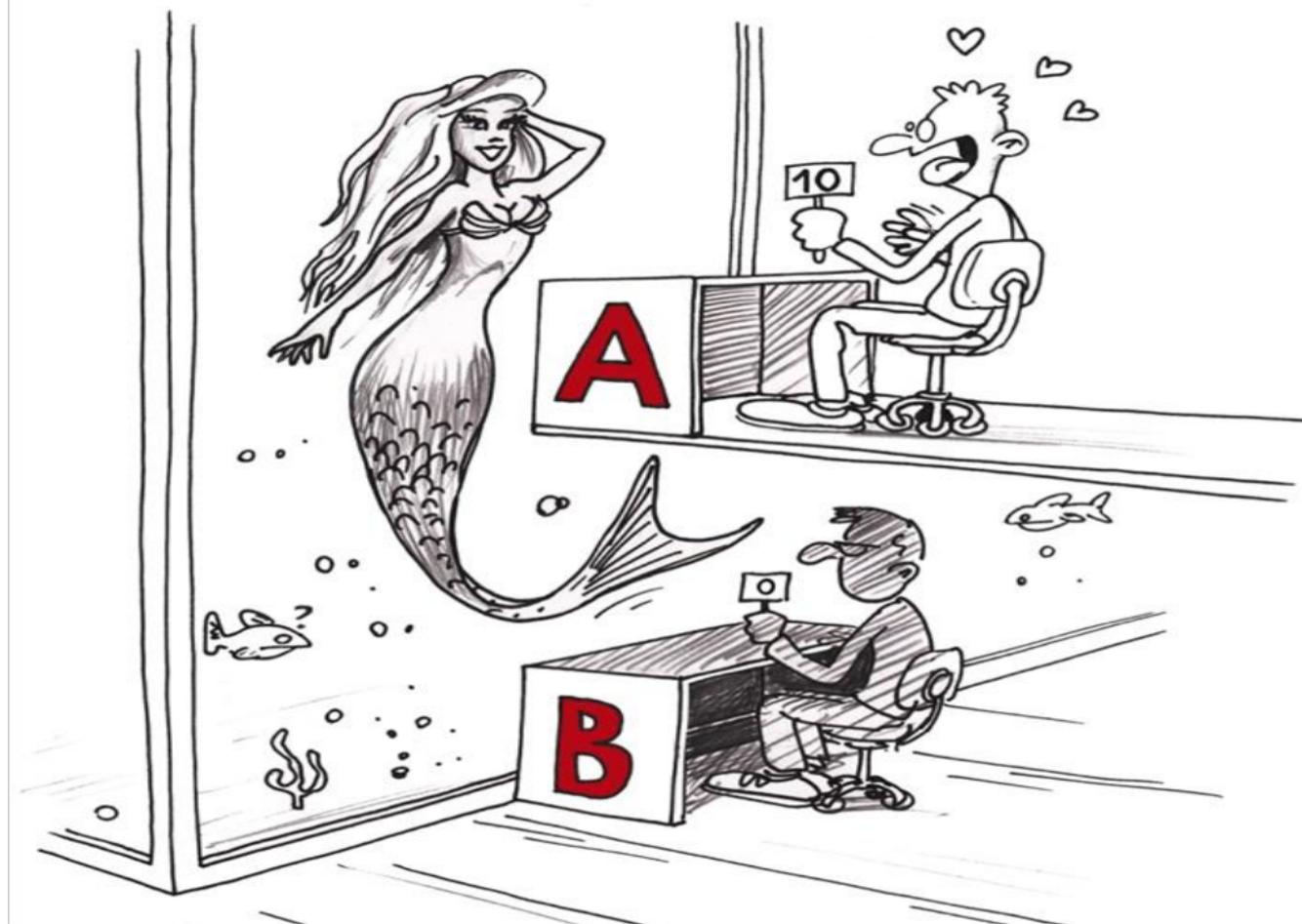


(Source : Les pratiques des géants du web / Stephen Perrin – OCTO Technology –
<http://fr.slideshare.net/sperin/les-pratiques-des-geants-du-web>)

Build vs Buy (2/2)

- > This dilemma is as old as the world of computers: is it better to invest in developing a software that is best suited to your needs or should you rely on a software package or third party product that embed the capitalization and R&D of **another** software editor in order to **-apparently** - speed up your time to market ?
 - + In order to be as efficient as possible on the build-measure-learn loop, it is essential to master your development process.
 - + For this reason, *tailor made* solutions are better because the adoption of a third party software package often requires to invest a lot of resources not in the development of your product, but instead in the development of workarounds, hacks and patches to correct all the points on which the software package is poorly adapted to the specific and precise behavior required by your own product feature.
 - + This cost aspect is particularly critical of course when scaling the solution. When one multiplies the processors and the servers, the invoice climbs very quickly and not necessarily linearly, and the costs become very visible, no matter whether it is a business software package or an infrastructure brick.
- > Most technologies that make the buzz today in the world of high performance architectures are the result of developments made by the Web Giants that have been released as Open Source: Cassandra, developed by Facebook, Hadoop and HBase inspired by Google and developed at Yahoo, Voldemort by LinkedIn, etc.
- > **Open-Source software is cool**
 - + Of course the cost problem doesn't apply to Open-Source and free to use software. In addition, instead of developing workarounds and patches around Open-Source Software, you can instead change its source, fork it and maintain your different baseline while still benefiting from the developments made on the official baseline by merging it frequently.

A/B Testing (1/2)



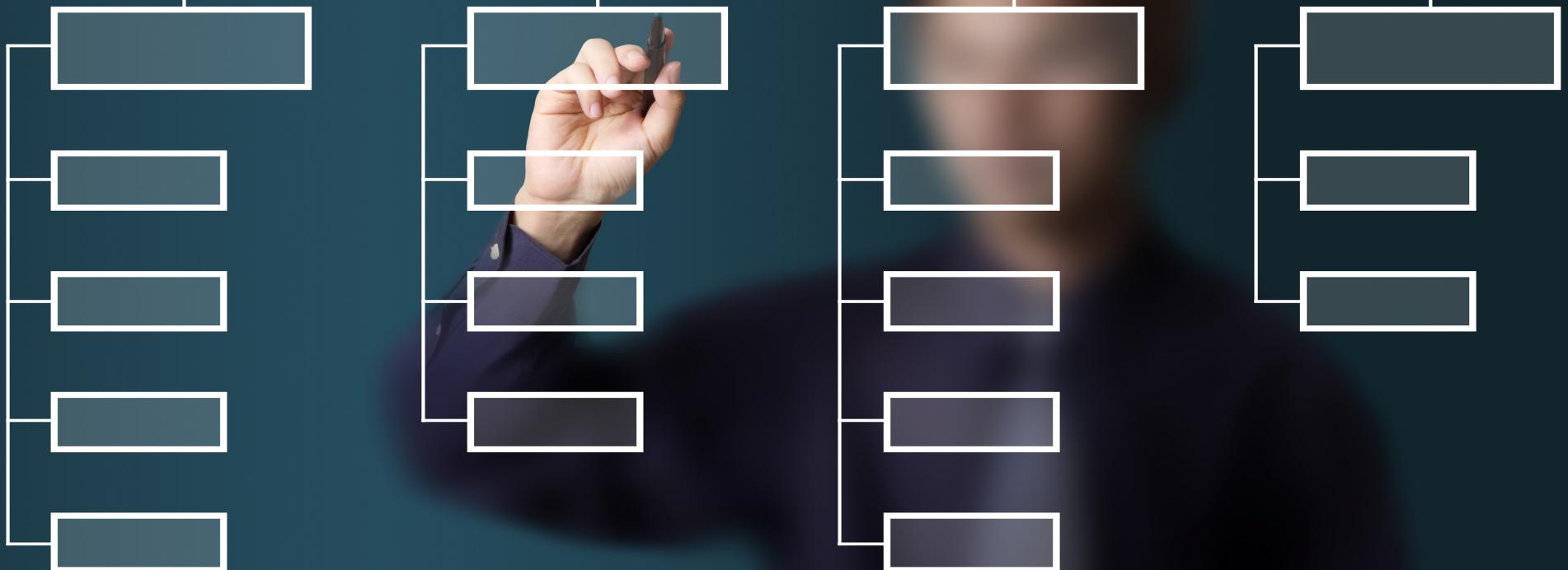
(Source : Les pratiques des géants du web / Stephen Perrin – OCTO Technology –
<http://fr.slideshare.net/sperin/les-pratiques-des-geants-du-web>)

- > A/B testing is a marketing technique that consists in proposing several variants of the same object that differ according to a single criterion (for example, the color of a package) in order to determine the version which lead to the best appreciation and acceptance from consumers.
- > A / B testing is used to qualify all kinds of multivariate tests.
- > An A/B test evaluates the respective performance of one or more partially or totally different versions of the same product or functionality by comparing them to the original version.
- > The idea is to split the visitors into two groups (hence the name A / B) and to present to each group a different version of the functionality or the product.
- > Then, we should follow the path of the two groups, their appreciation of the functionality by means of ad'hoc metrics, and we consider which of the two variants gives the best result with respect to a given objective.



- > Transforming a startup into a company, changing and scaling its organization is a unique, and yet challenging, opportunity to make it an agile organization keeping the *lean* genes on which it has been built.
The *agile* aspect here is essential and the approach here actually has a name: **Scaling Agile**.
- > *Scrum* and *Kanban* are two agile frameworks often used at the team level.
- > Over the past decade, as they gained popularity, the industry has begun to adapt and use Agile in larger companies.
 - + Two methods (amongst others) emerged to facilitate this process:
 - + **LeSS** (Large Scale Scrum) and
 - + **SAFe** (Scaled Agile Framework).
 - + Both are excellent starting points for using Agile on a large scale within a company.
- > Both approaches differ a little but also have a lot in common: they consist of scaling agility first among multiple agile team within the R&D or Engineering department and then around it, by having the whole company organizing its activities in an agile way and centered on the engineering team, the product development team.

3. Wrap-up and Take-Aways



- > Lean Startup is a movement
 - + mostly coming from the Silicon Valley
 - + 4 years old
 - + Initiated by successful serial-entrepreneur and academician
- > It gives keys, practices and advises to startup creators
- > From a purely software engineering perspective
 - + Lean startup practices are inspirational also for big corporations
 - + These practices are key to **scaling agile.**
- > **At the end of the day, Lean-Startup is about reaching Product-Market Fit**

Rachleff's Law of Startup Success:

- > **The #1 company-killer is lack of market.**
 - + When a great team meets a lousy market, market wins.
 - + When a lousy team meets a great market, market wins.
 - + When a great team meets a great market, something special happens.
- > **The market always wins!**

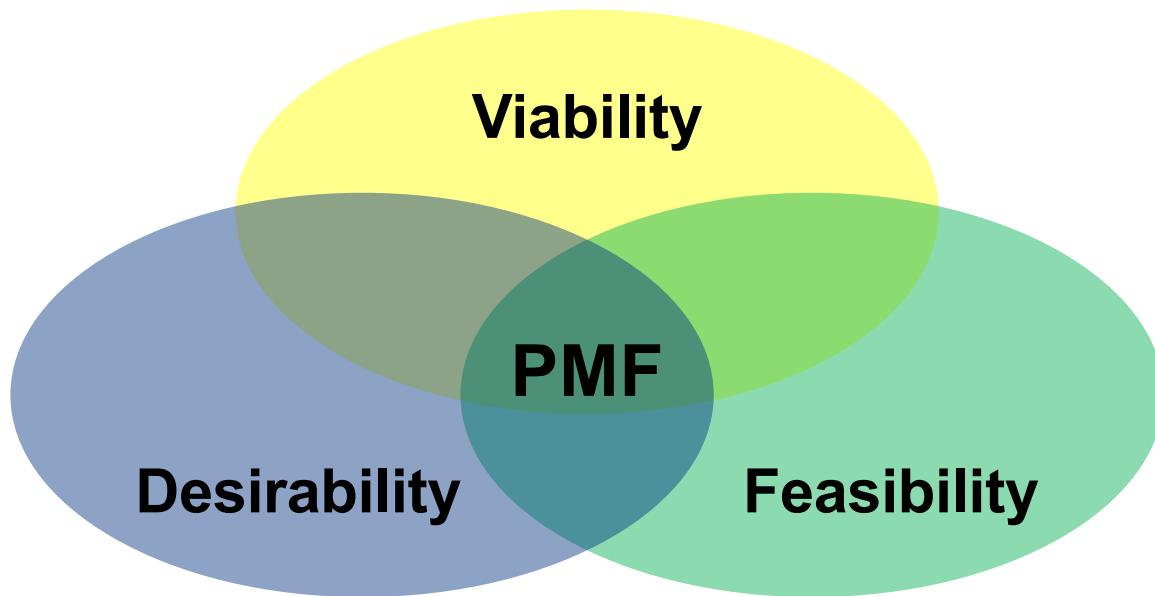
Rachleff's Corollary of Startup Success:

- > The only thing that matters is getting to **product/market fit**.
- > **Product/market fit means being in a good market with a product that can satisfy that market.**

Source : June 25, 2007 / Marc Andreessen - *The only thing that matters* (blog post)

So what is Product-Market Fit ?

A confirmed business model that can make the company real money

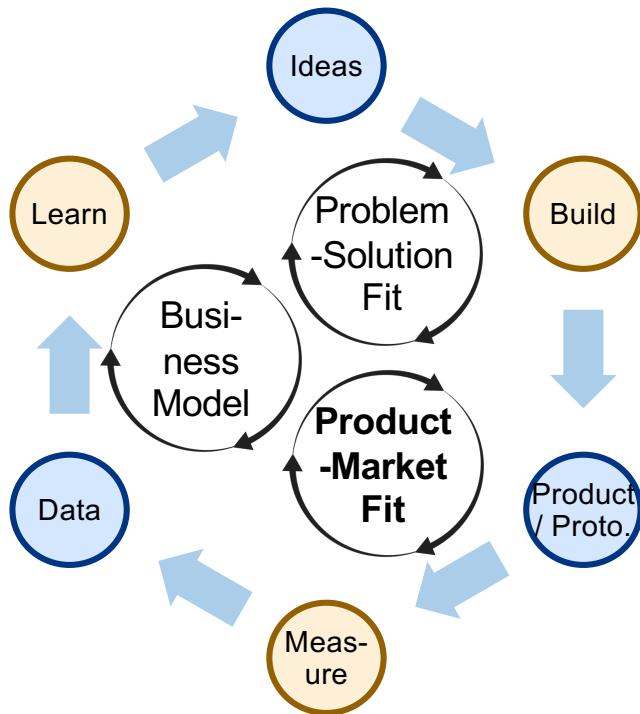


Large market of people who really need this problem solved

A product that you are able to actually build and that properly solved the problem

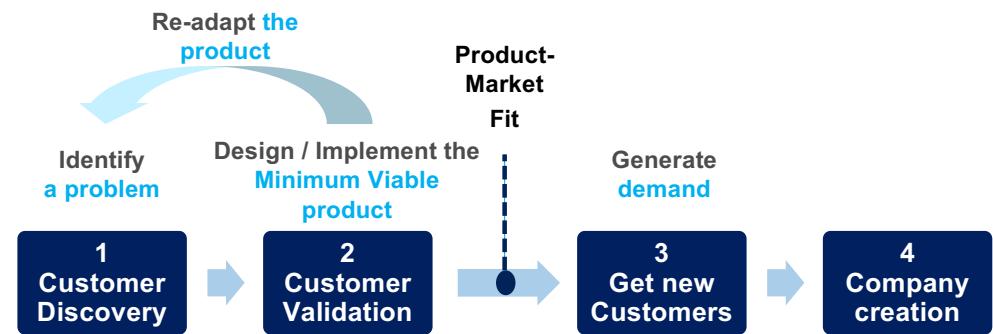
Source : <https://medium.com/@briantod/about-product-market-fit-what-ive-learned-about-the-goal-the-process-and-the-nuance-e7b317740f43>

Searching Product-Market Fit (1/2)



The Lean Startup perspective

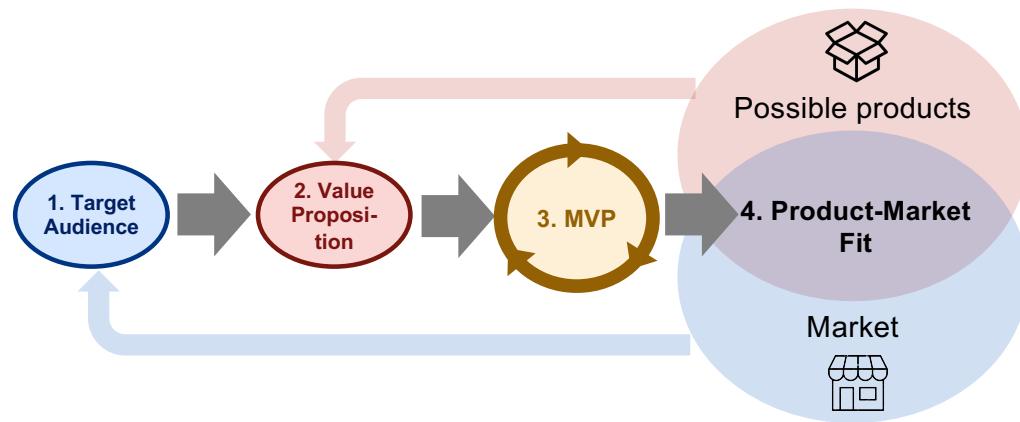
Searching Product-market fit is applying the *Build-Measure-Learn* feedback loop comprehensively throughout the product identification and design lifecycle and the business plan definition to shape a product fulfilling perfectly the customer needs.



The “Four steps to the Epiphany” perspective (also *Lean Startup*)

Product-market fit is the result of the search phase, when the solution to the customer problem is clearly identified along with its feature set, market potential, business plan, foreseen evolutions, etc.

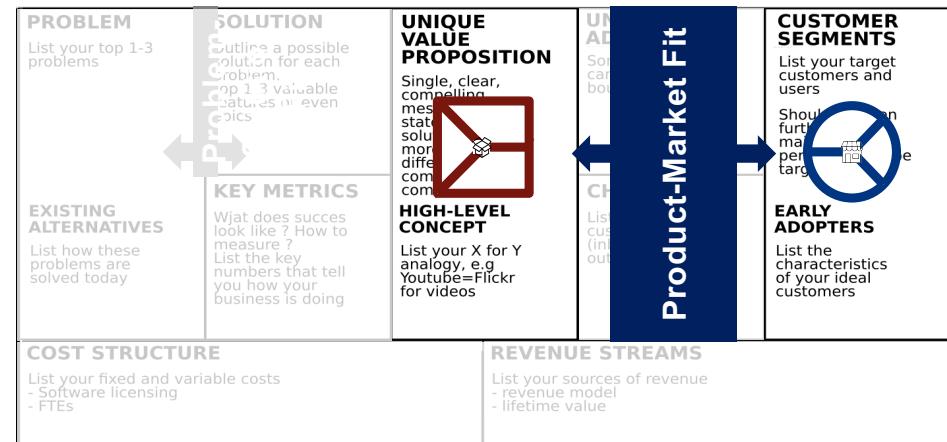
Searching Product-Market Fit (2/2)



The “MVP”-centric perspective *(somewhat again Lean Startup)*

For many, searching product-market fit is iterating around a minimum viable product. It is the result of a process centered around the MVP design iterations, when the MVP and what we learned from it enabled to identify the product fulfilling the market needs.

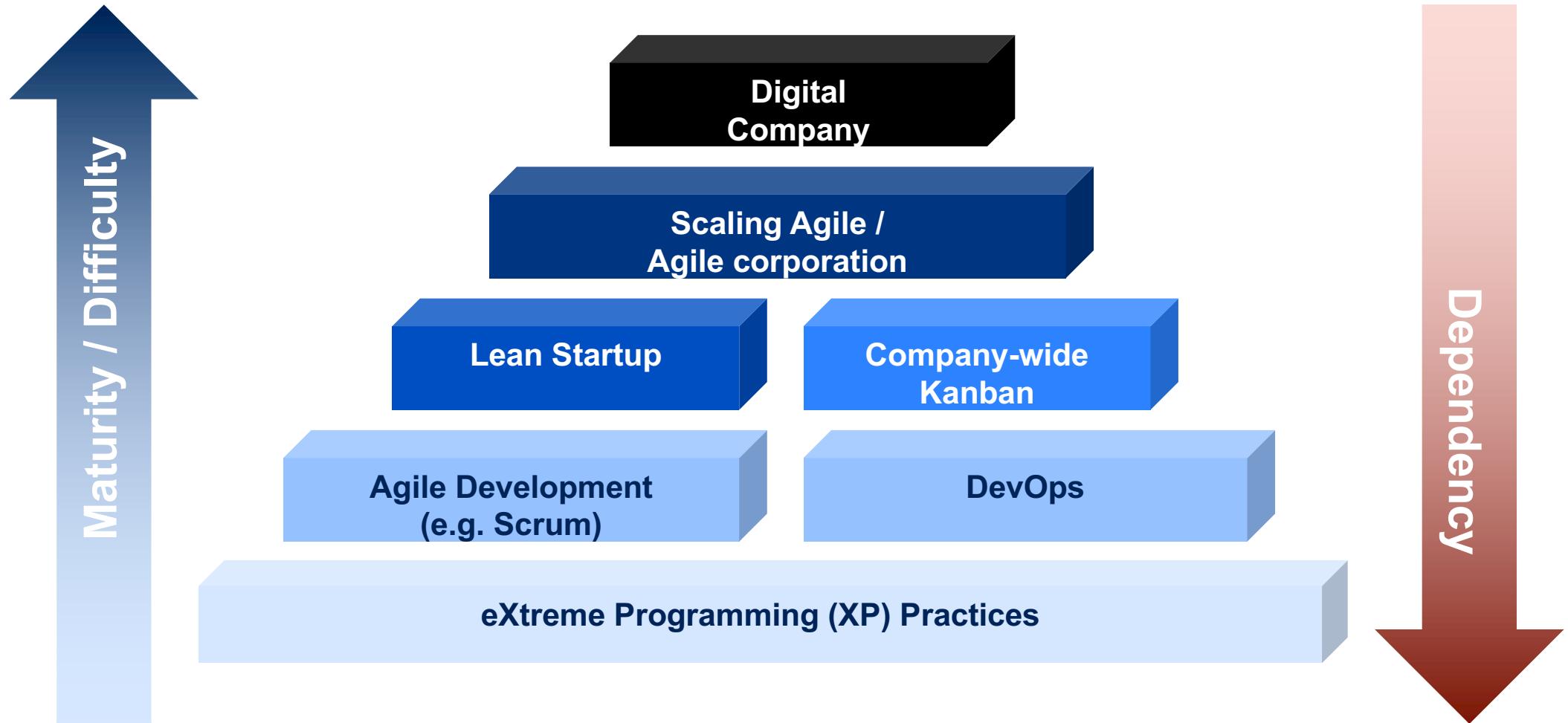
THE LEAN CANVAS (BUSINESS MODEL)



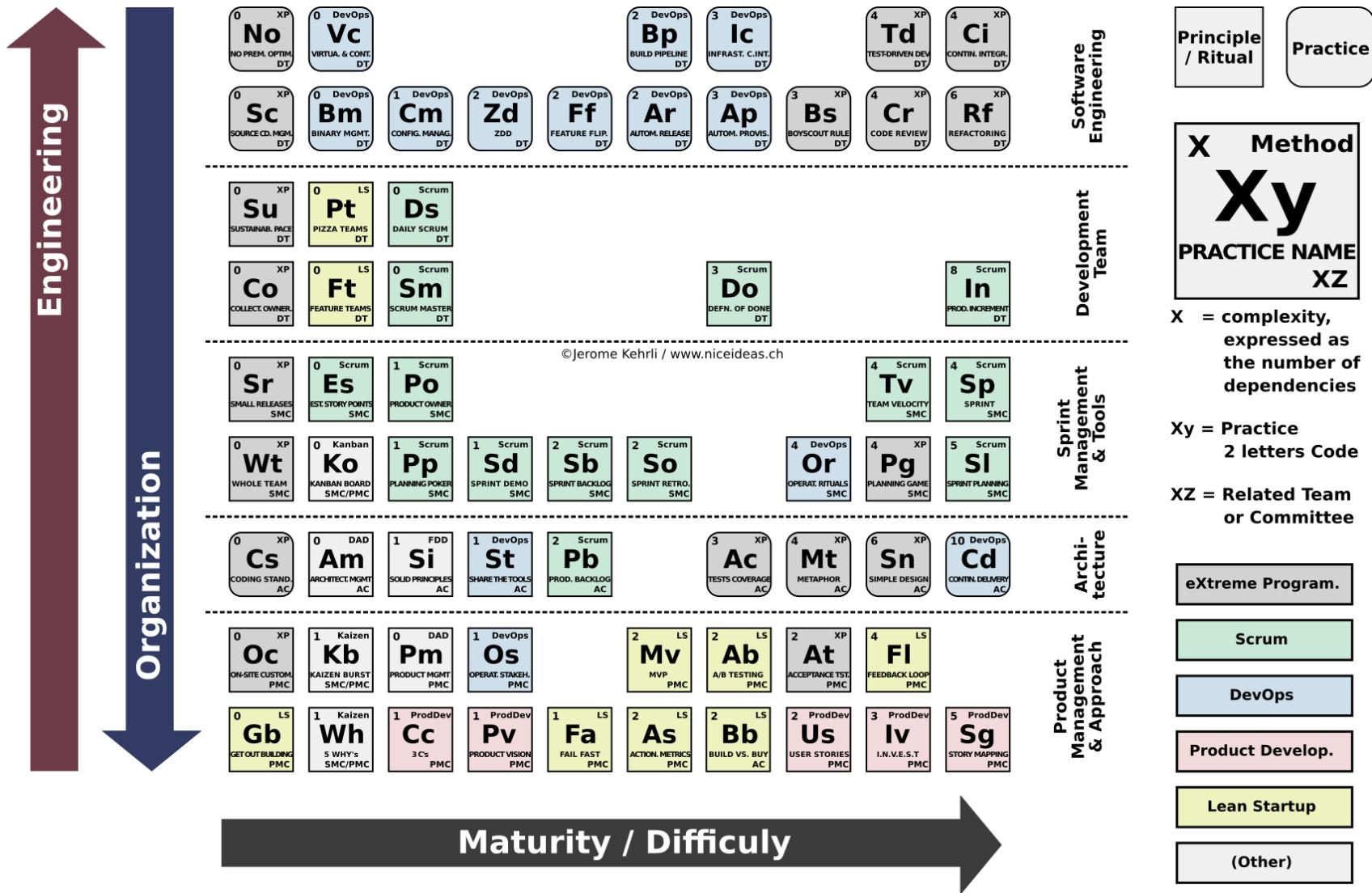
The “Lean Canvas / Value Proposition” perspective *(Did I mention Lean Startup?)*

For others, Product-market fit happens when you succeeded in designing great value propositions that match your customer needs and jobs-to-be-done and helps solve their problems.

DevOps / Lean Startup / Agility— why are these important ?



The periodic table of Agile Principles and Practices



<https://www.niceideas.ch/roller2/badtrash/entry/periodic-table-of-agile-principles>



A photograph of a person's hand holding a black marker, writing the text "Thanks for listening" in large, white, sans-serif letters on a whiteboard. The whiteboard is framed by a thick black border. The background is a dark, blurred indoor setting.

Thanks for listening