

# **BUSINESS MODELS, PIVOTS, ARCHITECTURE AND AGILITY**

---

Keep Austin Agile 2015

May 5, 2015



# Overview

---

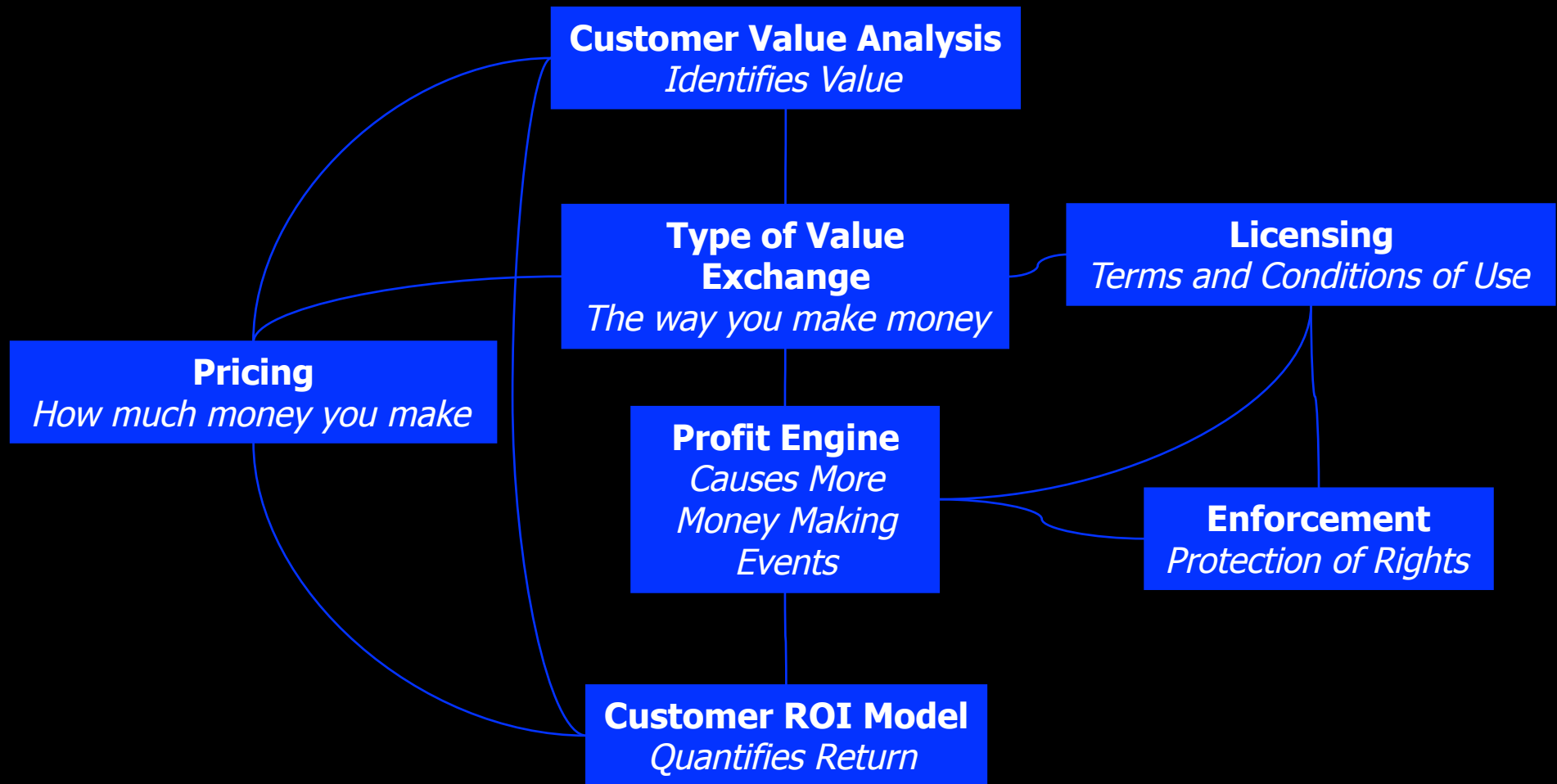
- › Great Products Make Money
- › 7 Ways Software Generates Revenue
- › Choices Affect Architecture
- › (Just enough) Lean Startup Principles (for this session)
- › Agility and Architecture
- › Case Study

# Business Model Defined

---

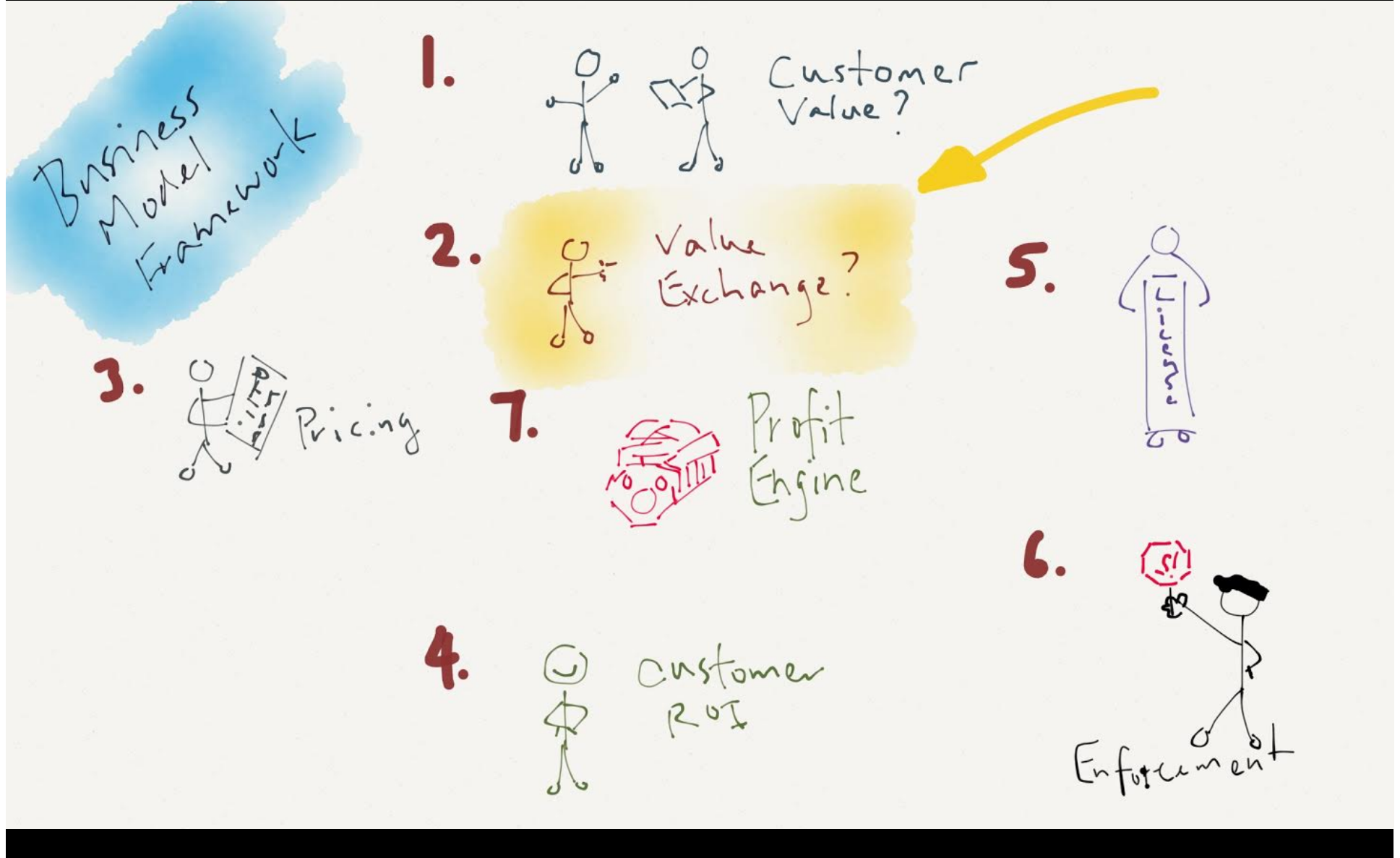
- › Interrelated choices that define the offerings you create, *how you make money* and how you structure your market relationships to generate maximum value.

# Business Model Framework

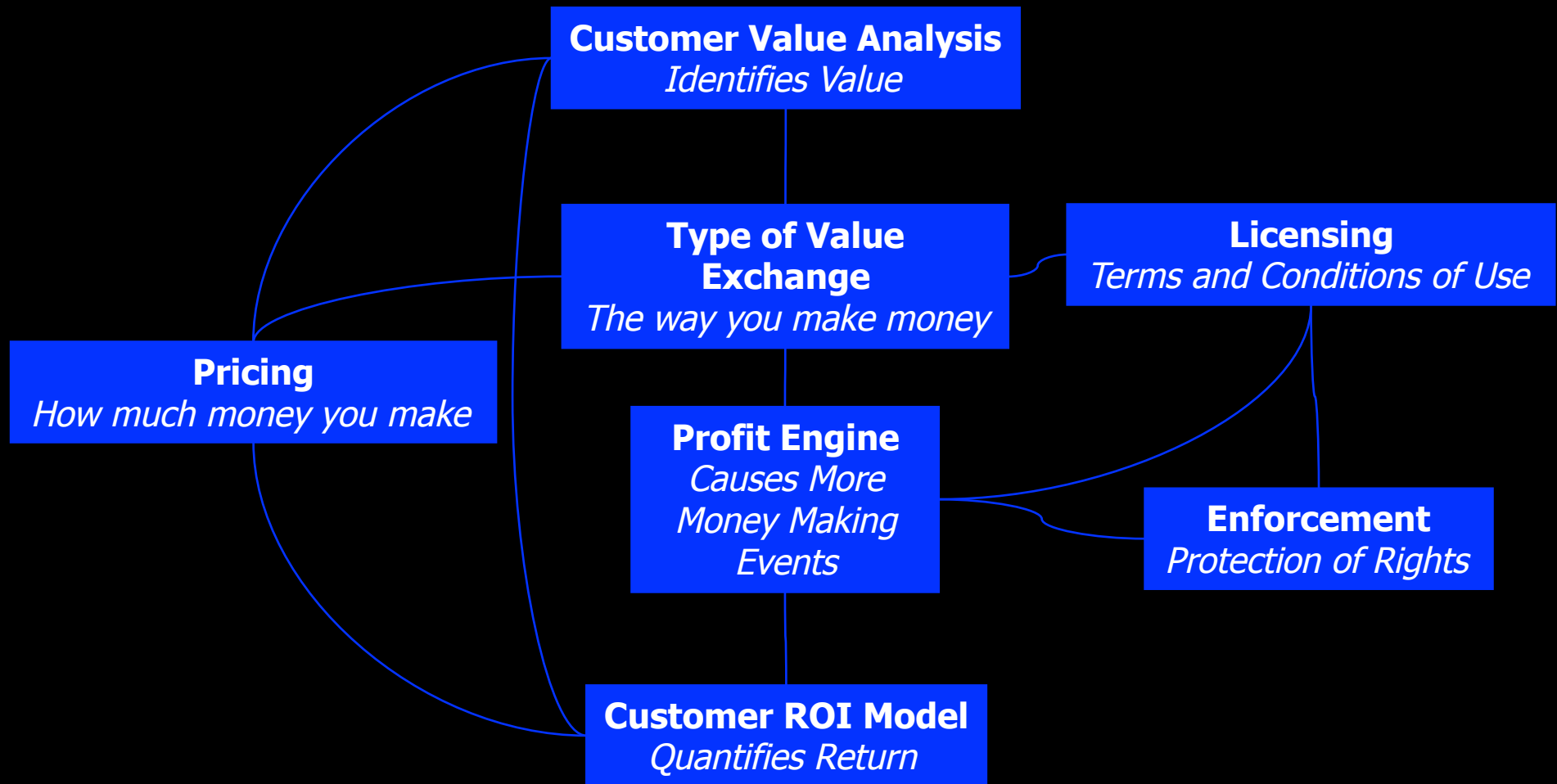


Notes: Multiple paths. Iterative. Directly connected to architecture. Our focus today is on Value Exchange.

# A Different View of the Framework



# Business Model Framework



Notes: Multiple paths. Iterative. Directly connected to architecture. Our focus today is on Value Exchange.

# Software Value Exchange Models

---

- › Time-based access
- › Transaction
- › Meter
- › Hardware
- › Service
- › Percentage of revenue gained/costs saved
- › Data/Content

# Time-Based Access

---

- › Grant “right to use” for a defined period of use (even if you don’t actually use product)
  - Perpetual (Microsoft Windows)
  - Annual (ERP)
  - Rental
  - Subscription
- › Pay after use





# Transaction

---

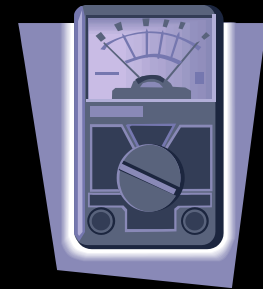
- › Defined and measurable units of work
- › Exchanged of money always tied to the transaction but customer value (and therefore price) often associated with an attribute of transaction
  - *Duration* of a phone call
  - *Time of day* call is made
  - *To whom* call is made



# Meter

---

- › Constraining a well-defined, identifiable resource
  - Hardware (e.g. 4 CPUs)
  - Named user
- › Consuming a well-defined resource
  - Concurrent (e.g. concurrent user)
  - Absolute value or consumptive (e.g. 100 hours)



# Hardware

---

- › Associate amount charged for software with some element of hardware
  - Software anti-piracy dongles – pay for both dongle and license SDK/run-time
- › Huge challenge – software becomes “free,” especially in embedded software
  - Home appliances: microwave ovens
  - Information appliances: router
  - Consumer electronics: LED TV



# Service

---

- › The exchange of money is tied to a service; software is required to provide the service or is intimately related to the service
  - Symantec: anti-virus updates
  - Red Hat: Linux-related *services* such as support or upgrades
- › Often associated with subscription pricing



# % of Revenue Gained/Costs Saved

---

- › Charge based on revenue obtained or costs saved, most often in terms of percentages
  - Retail yield management
  - HR, payroll, insurance

“The latest software company valued at over \$1B doesn’t make a dime selling software... raised \$500M...the round values the company at \$4.5B...After only 2 years, Zenefits is now valued at more than...Box and Zendesk....Zenefits gives all its software away free of charge...It generates revenue by...taking a percentage each time a user signs up for insurance.

Wall Street Journal, May 7, 2015



# Data/Content

---

- › The software creates unique data/content
- › The exchange of value provides access to these data
  - FICO scores
  - Re-processed government data (patents, TIGER/Line files)
- › Often associated with subscription pricing



Exercise: What's your Value Exchange Model?

# Software Value Exchange Models

---

- › **Time-based access** Right to use over time - perpetual, annual, subscription
- › **Transaction** Defined and measurable units of work
- › **Meter** Constraining/consuming a well-defined, identifiable resource - hardware, named user, absolute value
- › **Hardware** Embedded software – pay for software with hardware
- › **Service** Software required to provide service, often subscription – anti-virus
- › **Percentage of revenue gained/costs saved**
- › **Data/Content** Software creates unique data, often subscription - FICO

# Potential Architectural Impact

---

- › Each value exchange model is likely to affect your architecture. Value exchange model choices intertwine with technical choices
- › Your architecture will either constrain or support alternative business models
  - Does your architecture fully support your model?
  - Are you (perhaps unexpectedly) constrained by business model choices?
- › Let's review two examples



# Effects of time-based access on architecture

---

- › When term expires, what should I do?
- › One option, disallow use
  - What does disallow mean?
    - Retain data?
    - Access to data?
  - How?
  - Notifications?
  - Sequence to resume service?
- › What about multiple computers?
- › Rental/other models require both protection and backend systems to support

# Effects of transactions on architecture

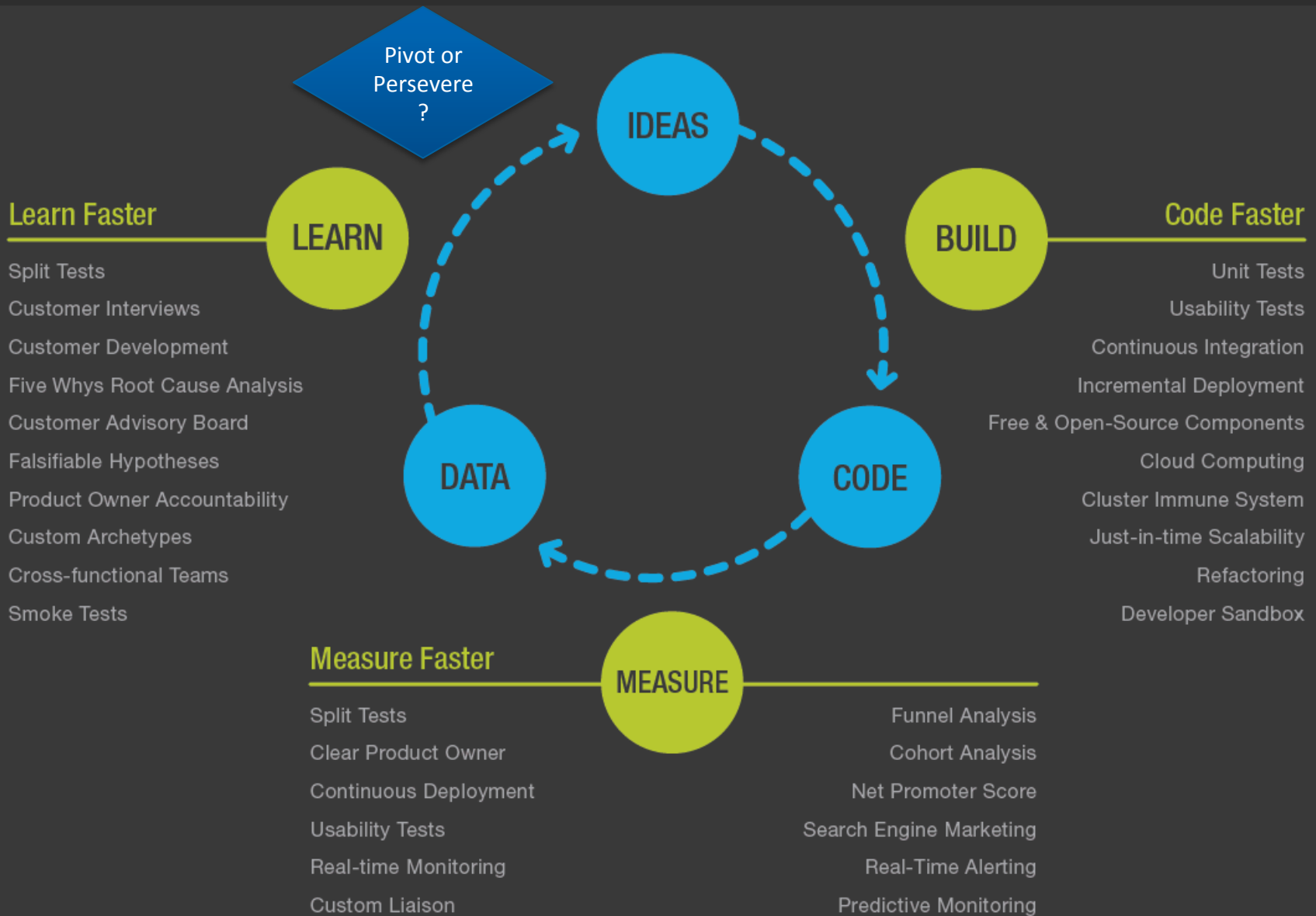
---

- › Do legal and technical definitions match?
- › Uniquely identify & understand entire life
- › Capturing the necessary data (schema)
  - Direct: System captures and manages all data necessary to support the business model
  - Indirect: System must be integrated with other systems to capture/manage necessary data
- › Reporting/remittance requirements
- › Format? Security? Auditability?

# THE LEAN STARTUP

Created by Eric Ries - [startuplessonslearned.blogspot.com](http://startuplessonslearned.blogspot.com)

Designed by  KISSmetrics



# Types of Pivots

---

1. Zoom-In – single feature becomes whole product
2. Zoom-Out – product becomes feature of much larger product
3. Customer Segment – solves problem for different customer
4. Customer Need – solves different problem – reposition or new product
5. Platform – app to platform or reverse
6. Business Architecture – complex systems vs. volume operations
7. Value Capture – change to how company captures value
8. Engine of Growth – change to seek faster or more profitable growth (may include 7.)
9. Channel – different channel with greater effectiveness
10. Technology – same solution, different tech

# Exercise: Pivots and Architecture

Mark architecture impact of each pivot on the scale.

1. **Zoom-In** single feature becomes whole product



2. **Zoom-Out** product becomes feature of much larger product



3. **Customer Segment** solves problem for different customer



4. **Customer Need** solves different problem – reposition or new product



5. **Platform** app to platform or reverse



6. **Business Architecture** complex systems vs. volume operations



7. **Value Capture** change to how company captures value



8. **Engine of Growth** change to seek faster or more profitable growth  
(may include 7.)



9. **Channel** different channel with greater effectiveness



10. **Technology** same solution, different tech



# Agility

---

- › Defined by four values and twelve principles
- › Key principles for the case study
  - Deliver frequently
  - Working software is the primary measure of progress
  - Welcome changing requirements
  - Our highest priority is to satisfy the customer through early and continuous delivery

# Exercise: Pivots and Agility

Mark the impact to Agility of each pivot on the scale.

1. **Zoom-In** single feature becomes whole product

Low High  
←→

2. **Zoom-Out** product becomes feature of much larger product

Low High  
←→

3. **Customer Segment** solves problem for different customer

Low High  
←→

4. **Customer Need** solves different problem – reposition or new product

Low High  
←→

5. **Platform** app to platform or reverse

Low High  
←→

6. **Business Architecture** complex systems vs. volume operations

Low High  
←→

7. **Value Capture** change to how company captures value

Low High  
←→

8. **Engine of Growth** change to seek faster or more profitable growth  
(may include 7.)

Low High  
←→

9. **Channel** different channel with greater effectiveness

Low High  
←→

10. **Technology** same solution, different tech

Low High  
←→

# Case Study – Part 1

---

- › You are the founder of an Agile consulting company that provides market research services to engage customers to collaborate in fun activities to identify unmet needs.
- › One of your biggest clients asks you to build an online version of the activities. You use Scrum to build the initial product and show it to several other clients. Everyone likes it.
- › Question 1: What value exchange model(s) do you consider?



## Case Study – Part 2

---

- › When you ask your clients, all of them say they prefer a Transaction model – pay for each game.
- › You and your team build the product with a transaction model and launch.
- › Almost no one signs up.
- › Question 2: What do you do?

## Case Study – Part 3

---

- › You and your team decide to change the value exchange model from transaction to time-based access.
- › Question 3: How would you describe the potential architectural impact?
- › Question 4: What is the impact of the change to the team's Agility?

# Summary

---



- › Framing choices help us make better decisions
- › Pivots are harder & will take longer than you think
  - Rarely 1 sprint
  - Impacts whole product
  - Impacts not just new customers but existing customers
  - May be  $\frac{1}{4}$  dev,  $\frac{3}{4}$  prod mgmt/prod mktg
- › Nearly impossible to make business models “configurable” but you may have primary/2ndary
- › For a true startup, will you run of \$ before you complete your pivot?

# Questions?

# Thank you!

› Training:  
AppliedFrameworks.eventbrite.com

› The Real Story of the Case  
Study

<http://bit.ly/sgpivotcase>

@jasonbtanner

jtanner@appliedframeworks.com

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

919-740-9678

Deliver the  
**RIGHT PRODUCT**  
To the  
**RIGHT MARKET**  
At the  
**RIGHT TIME**  
With the  
**RIGHT PROCESS**