Solvay Brussels School
Economics & Management

Course GESTS403 & GESTH510
IT Governance
Digital Agility

Frédéric Hoffmann – Jérôme Grandbastien

EQUIS ACCREDITED Association of MBAs ULB

Frédéric Hoffmann

- With more than 20 years business experience, including 12 years of consulting, Frederic acquired a broad expertise in Project & Portfolio Management (PPM) by implementing PPM best practices, methods & tools for major international private & public institutions.
- He is now part of the Transformation Office of BNP Paribas Fortis where he's responsible for traditional and Agile project & portfolio management tools.
- He is also professor and executive coach at the Solvay Brussels School of Economics & Management for the Masters, Executive Programs and for the MBA in Brussels and in Vietnam (Saigon and Hanoi).
- He holds a master degree in Economics and a postgraduate master degree in Management of Information Technology.

Frédéric Hoffmann
frederic.hoffmann@solvay.edu

Mobile: +32 496 57 44 79

Jérôme Grandbastien

- Jerome Grandbastien is an eager and enthusiastic Agile certified coach at BNP Paribas Fortis.
- Jerome has more than 10 years experience in IT where he filled a number of roles from developer to project manager and team leader.
- Over the last years, his primary focus was the Agile mindset and the Scrum framework.
- He loves sharing experience with others and to see how to improve things !.

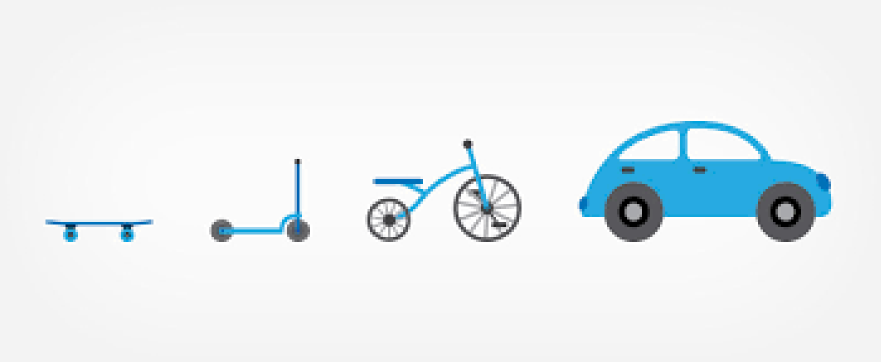
Jerome Grandbastien
Jerome.grandbastien@gmail.com

Mobile: +32 477 48 19 14

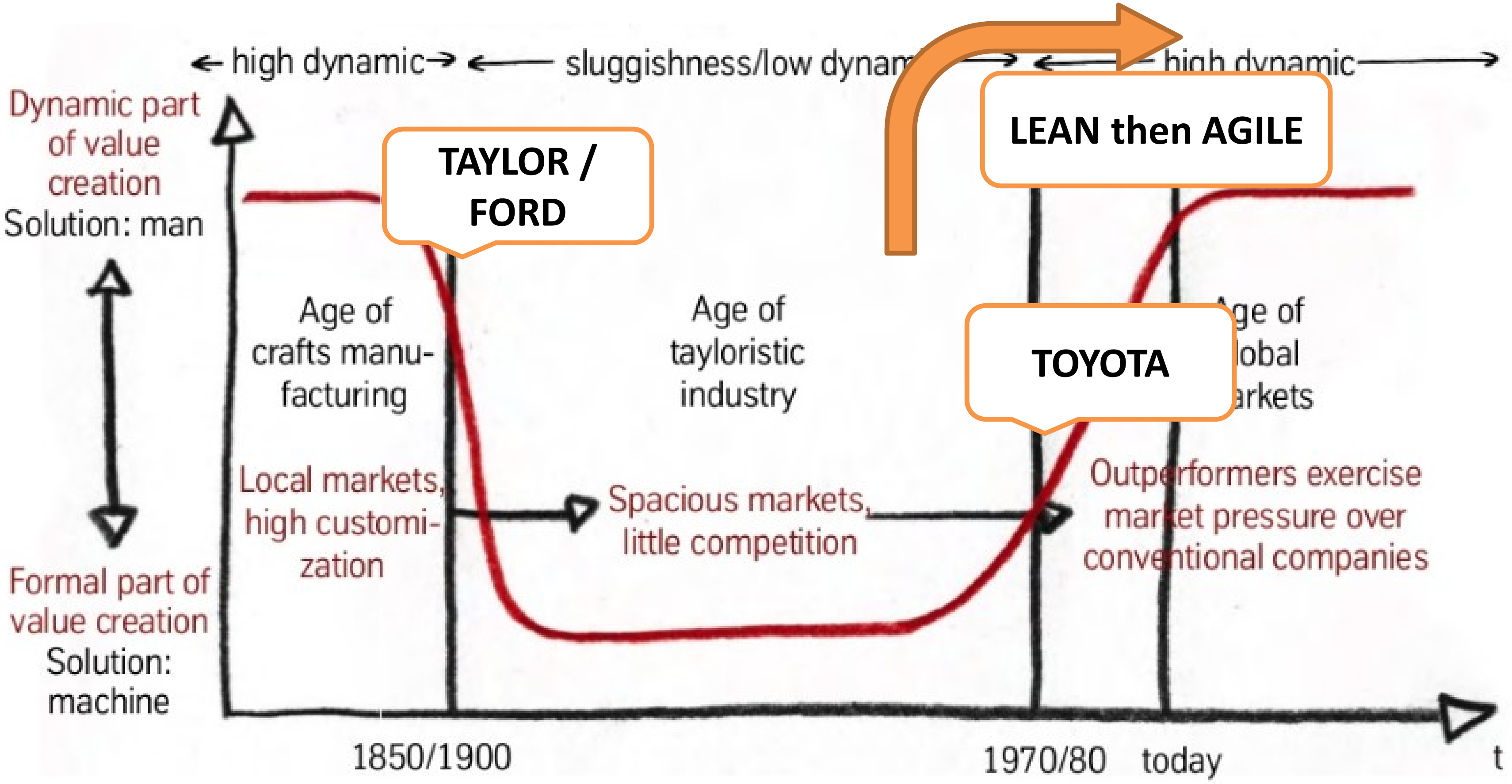
Agenda

- **WHY Agile and WHAT is Agile**
  - Organization complexity (self organized)
  - Lean introduction
  - Agile Manifesto
  - Empirical and incremental
- **Agile in today's industry**
  - Case studies : with Zappos, Spotify, Tesla, Sony, …
- **Lean StartUp basics**
  - How to develop a product (software product) with Agile / Lean practices.

WHY Agile and WHAT is Agile



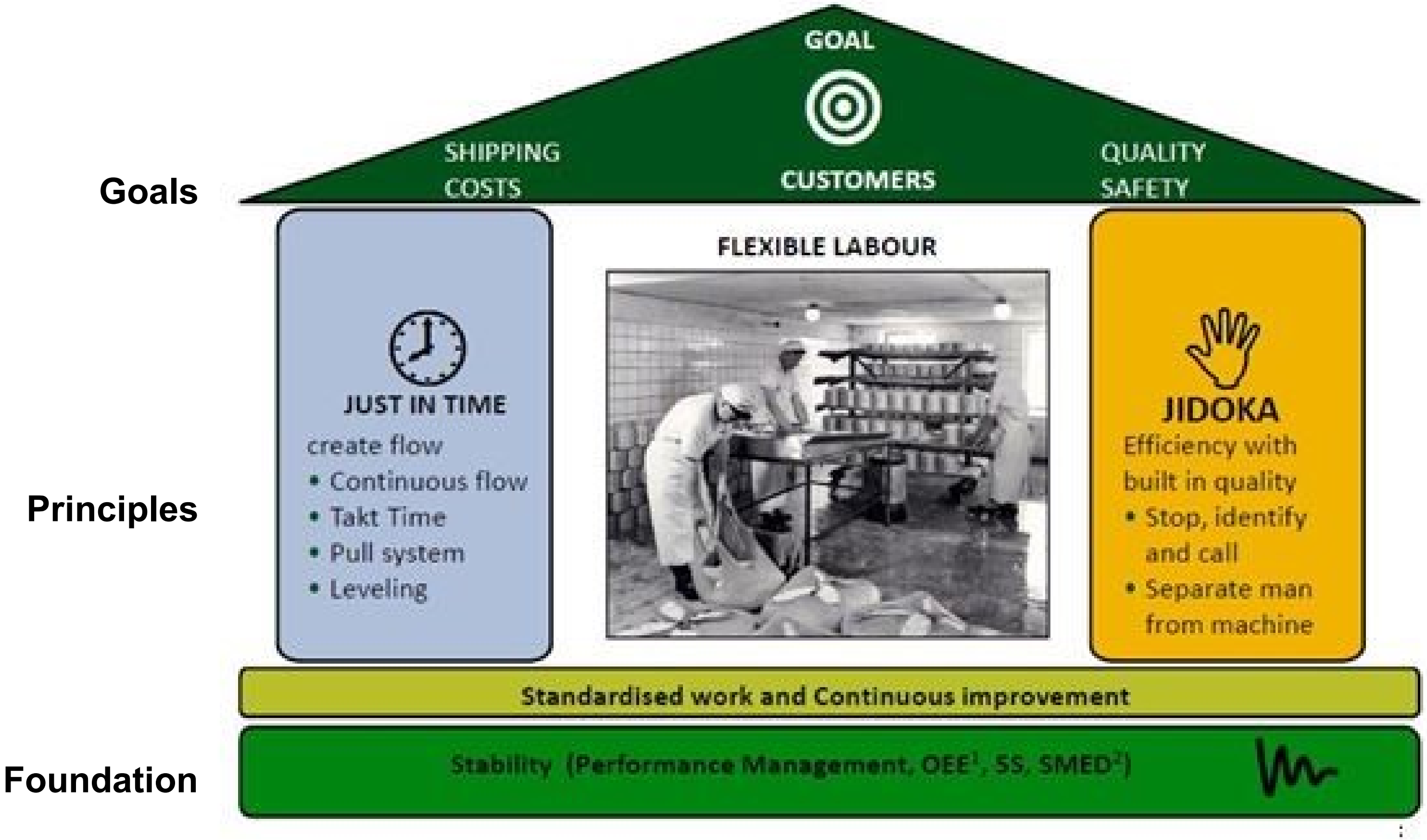
Organization complexity



Source: BetaCodex Network Associates, "Organize for complexity", BetaCodex Network White Paper 12 & 13

Lean introduction

A lean production system has clear goals, 2 principles (pillars), and a solid foundation



Goals

GOAL

SHIPPING
COSTS

CUSTOMERS

QUALITY
SAFETY

Principles

JUST IN TIME
create flow
- Continuous flow
- Takt Time
- Pull system
- Leveling

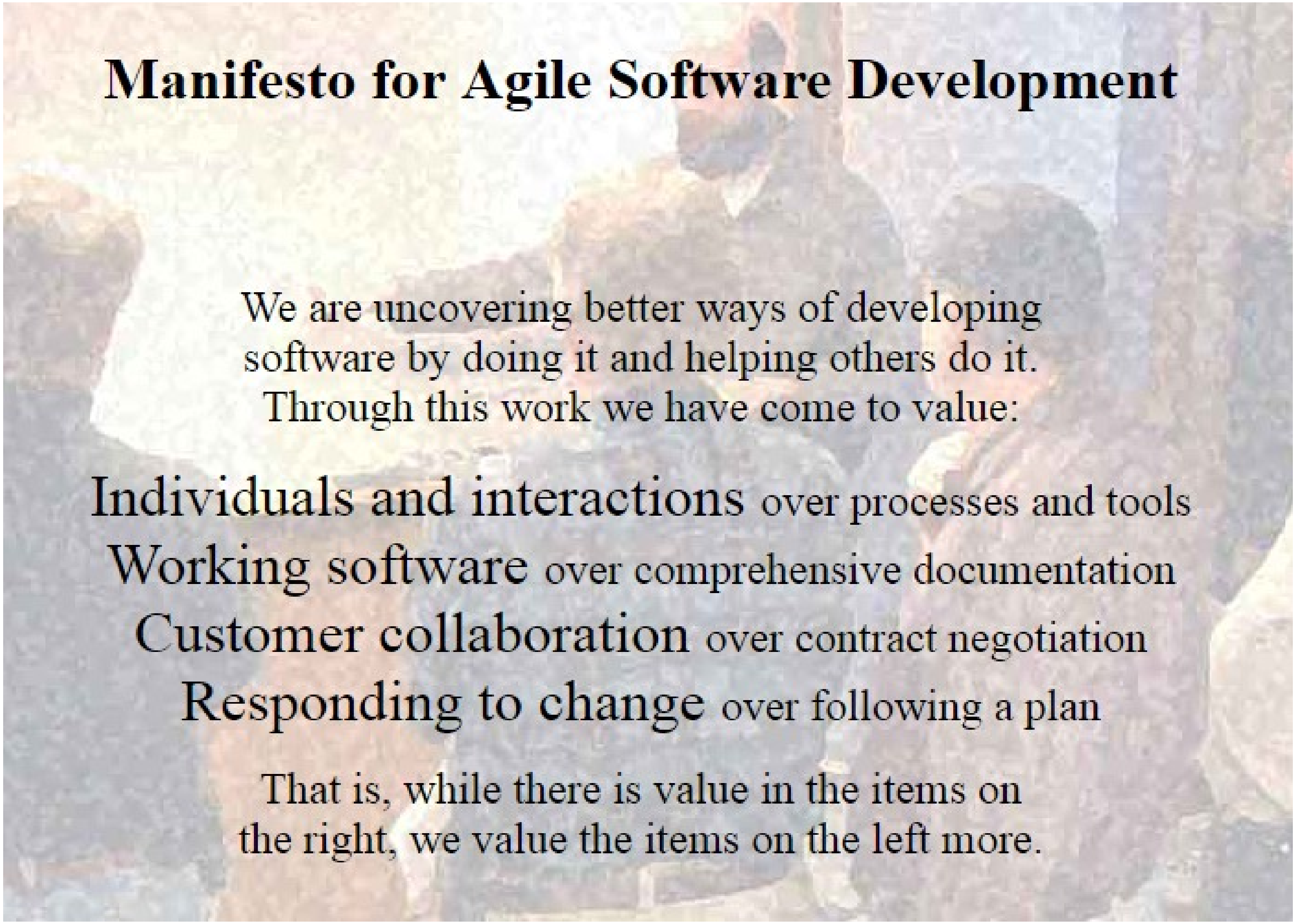
FLEXIBLE LABOUR

JIDOKA
Efficiency with built in quality
- Stop, identify and call
- Separate man from machine

Foundation

Standardised work and Continuous improvement

Stability (Performance Management, OEE[1], 5S, SMED[2])

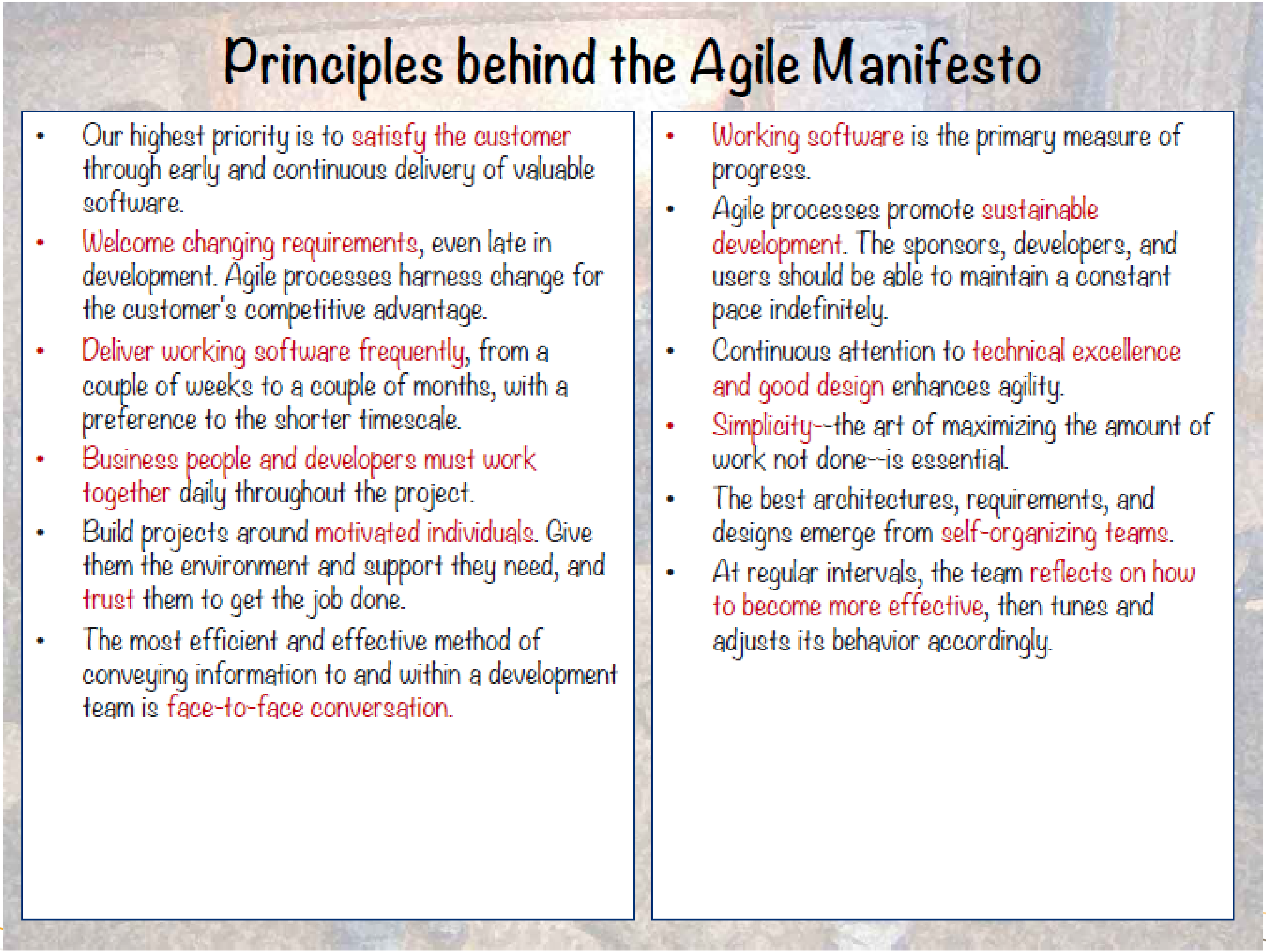


Manifesto for Agile Software Development

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 software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

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



Principles behind the Agile Manifesto

- Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- Business people and developers must work together daily throughout the project.
- Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
- Working software is the primary measure of progress.
- Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
- Continuous attention to technical excellence and good design enhances agility.
- Simplicity--the art of maximizing the amount of work not done--is essential.
- The best architectures, requirements, and designs emerge from self-organizing teams.
- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Agile Manifesto – 12 Principles

@ http://agilemanifesto.org/principles.html

1. **Satisfaction and delivery**
2. **Welcome changing**
3. **Deliver frequently**
4. **Communication is the key**
5. **Environment and trust**
6. **Face to face communication**
7. **Measuring progress**
8. **Sustainable development**
9. **Technical excellence**
10. **The power of less**
11. **Self-organizing team**
12. **Adjusting at regular intervals**

Why?

How?

Agile is...

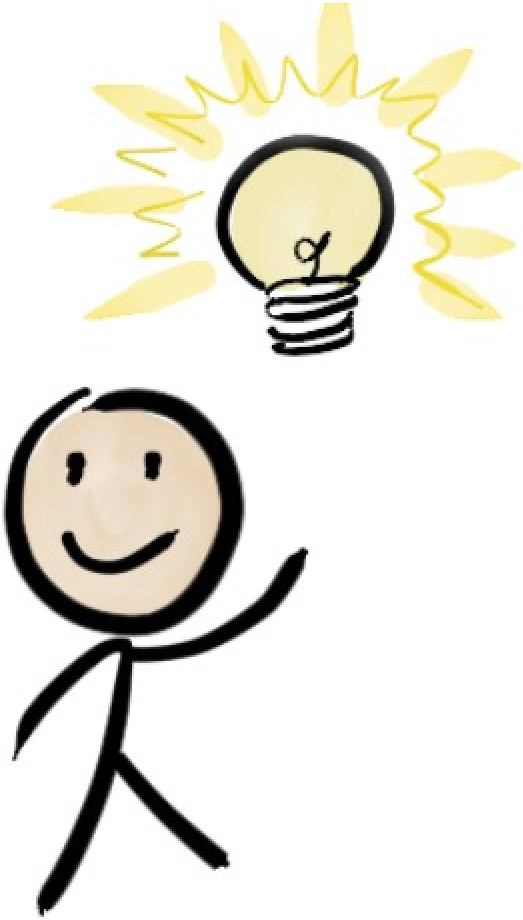
Early delivery of business value

Less bureaucracy

Henrik Kniberg

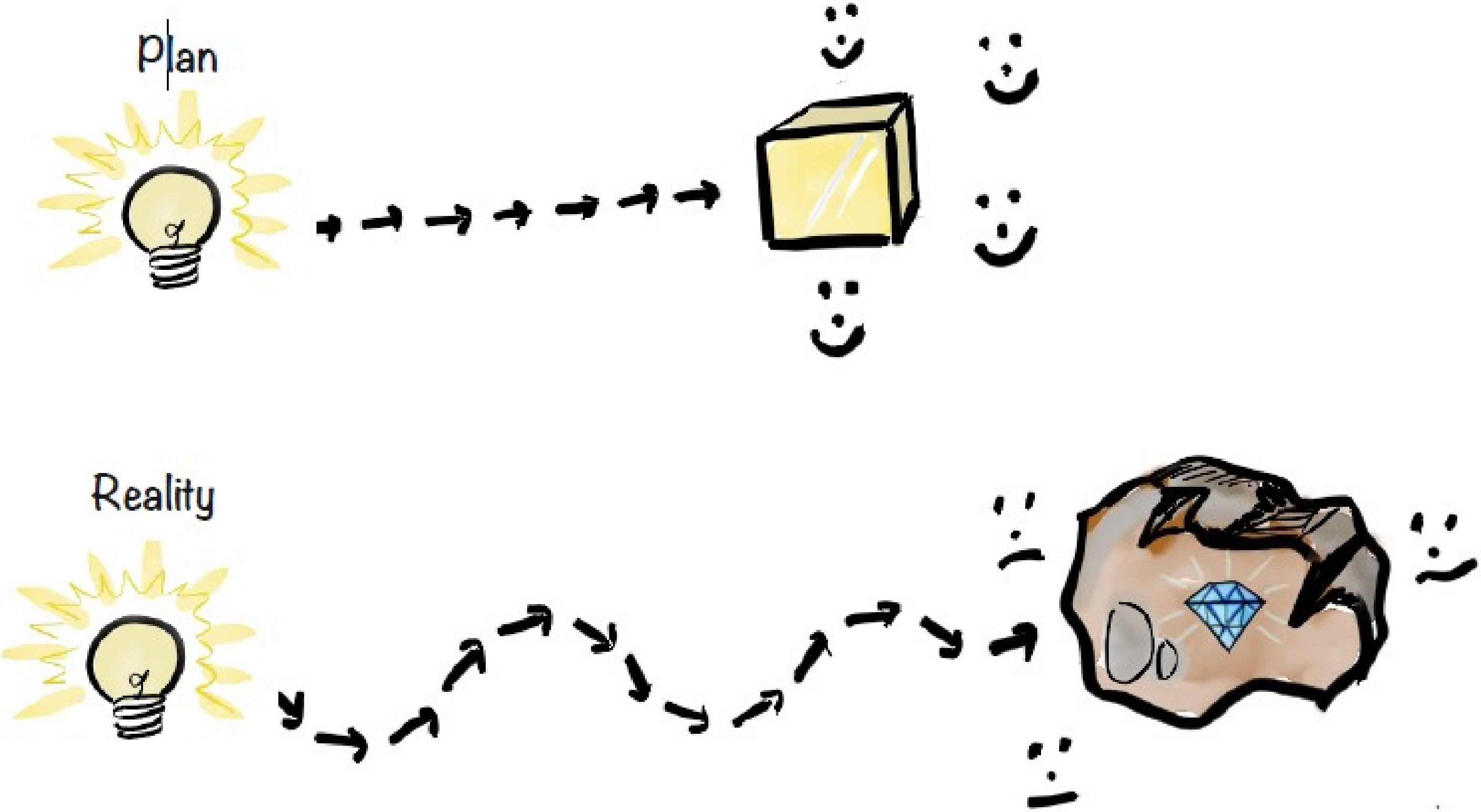
(Thanks Alistair Cockburn for this simplified definition of Agile)

All products / features start with a Great Idea!



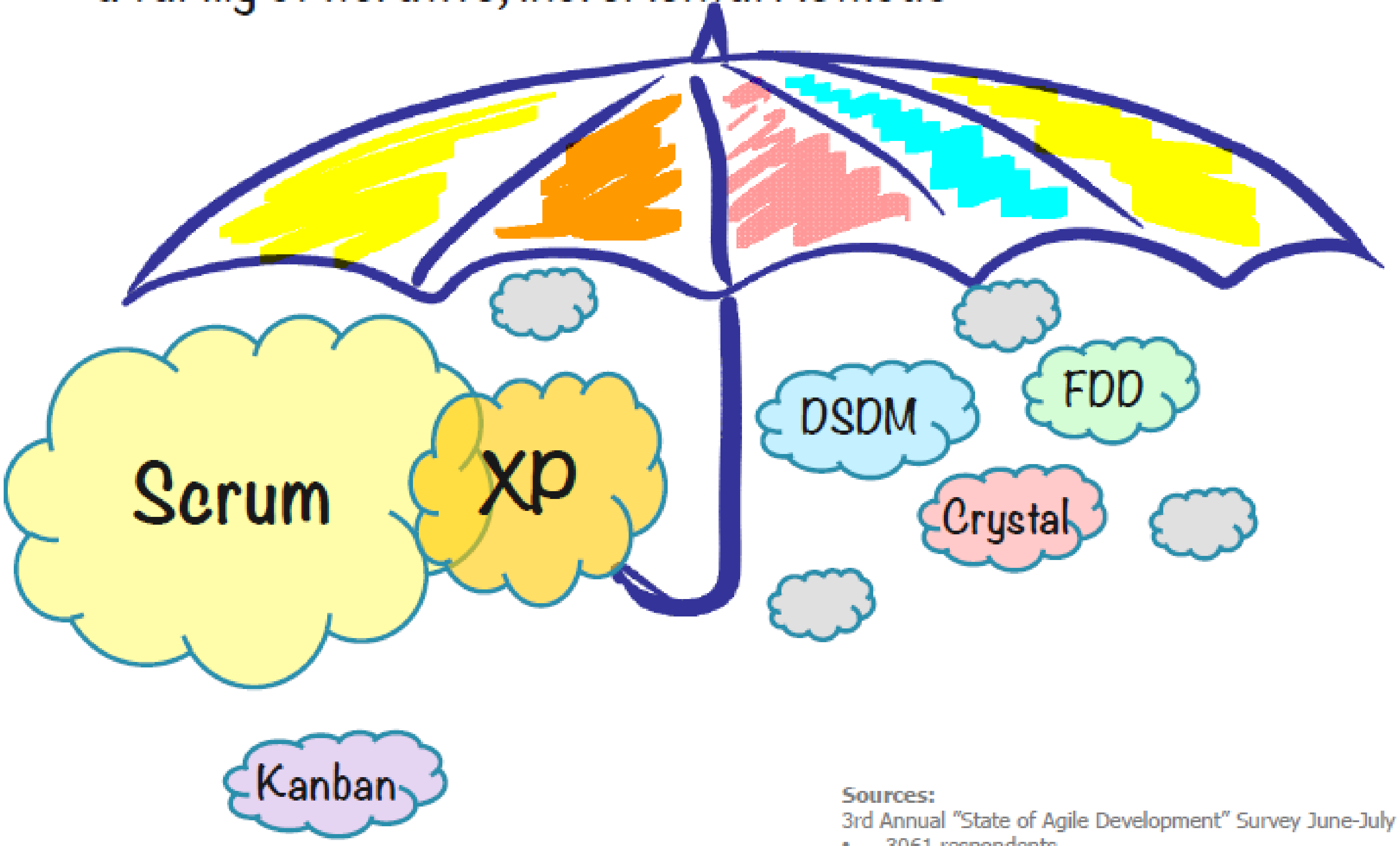
Henrik Kniberg

Unfortunately..... it is likely to fail



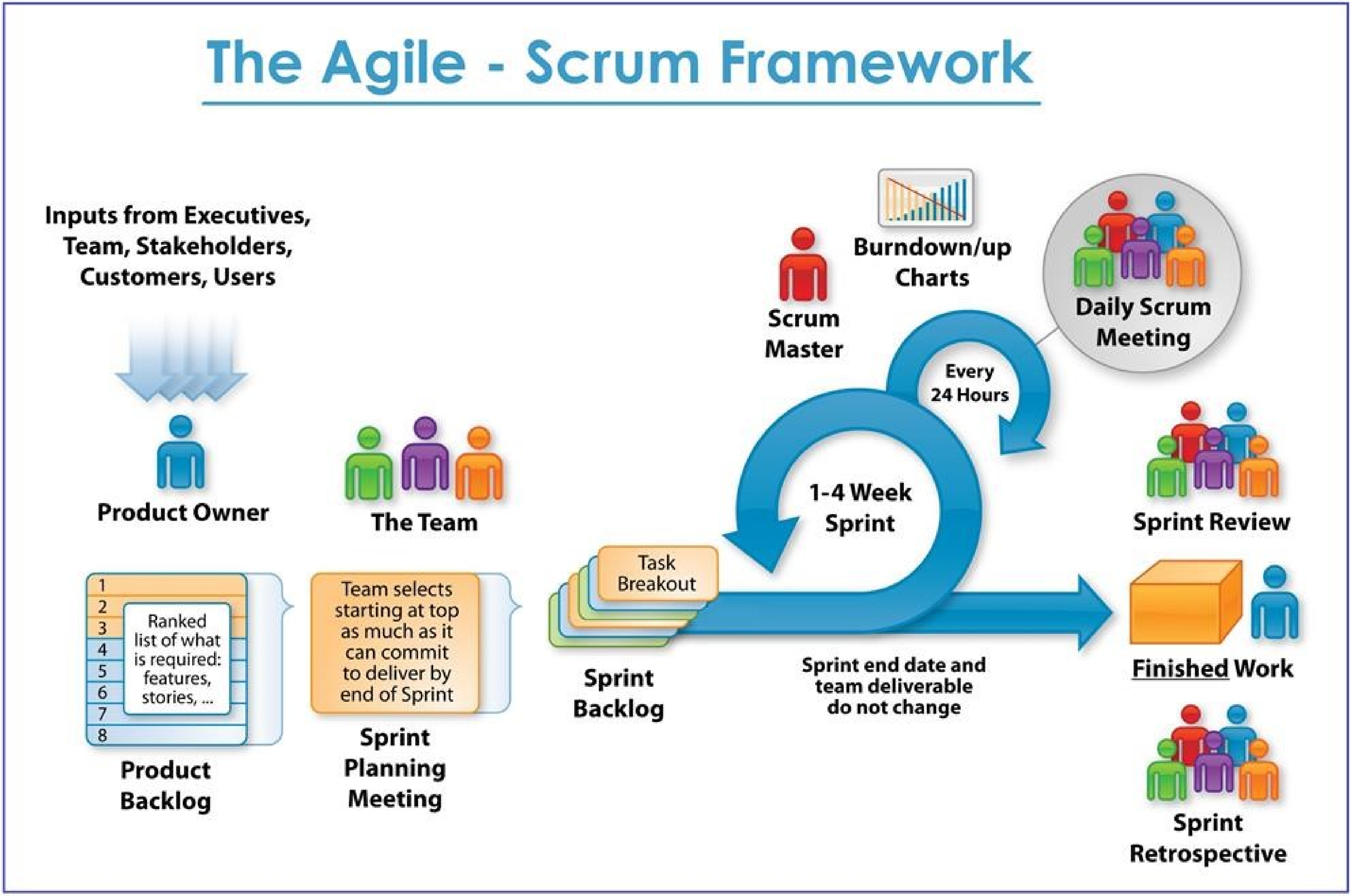
Henrik Kniberg

Agile "umbrella" –
a family of iterative, incremental methods



Sources:
3rd Annual "State of Agile Development" Survey June-July 2008
- 3061 respondents
- 80 countries

Scrum Framework



The Agile - Scrum Framework

Inputs from Executives, Team, Stakeholders, Customers, Users

Product Owner

The Team

Scrum Master

Burndown/up Charts

Daily Scrum Meeting

Every 24 Hours

1-4 Week Sprint

Ranked list of what is required: features, stories, ...

Product Backlog

Team selects starting at top as much as it can commit to deliver by end of Sprint

Sprint Planning Meeting

Task Breakout

Sprint Backlog

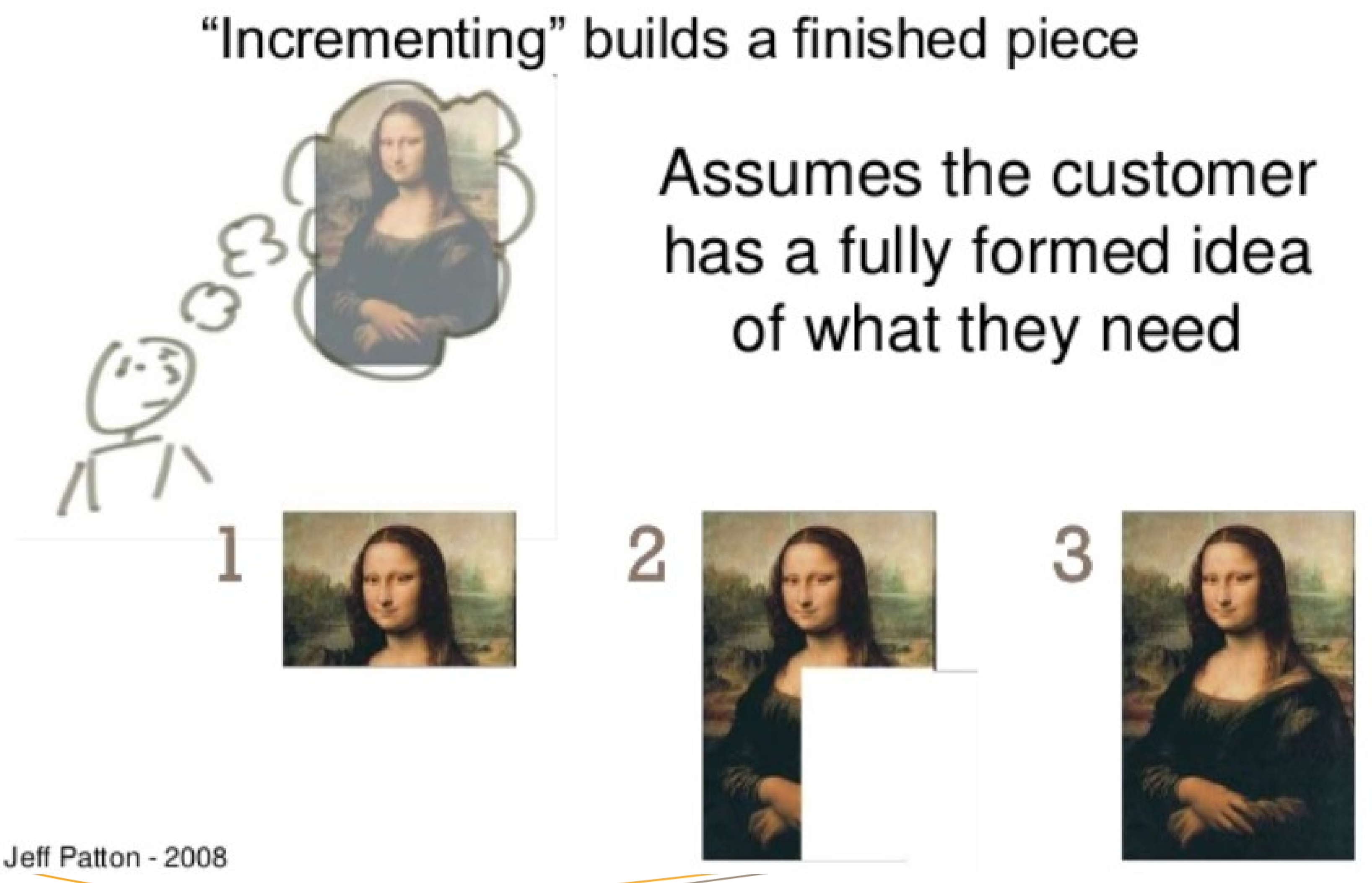
Sprint end date and team deliverable do not change

Sprint Review

Finished Work

Sprint Retrospective

Iterative and Incremental



"Incrementing" builds a finished piece

Assumes the customer has a fully formed idea of what they need

Jeff Patton - 2008

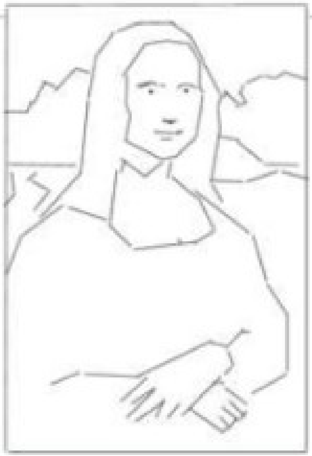
Iterative and Incremental

“Iterating” builds something imperfect to validate and refine

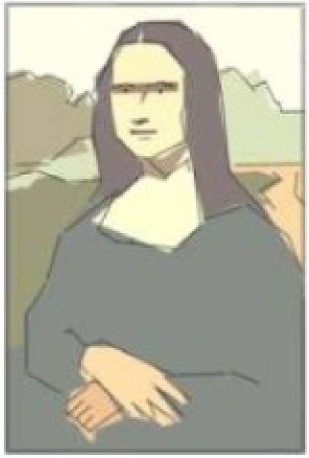
woman
in pastoral
setting

Iterating allows for less developed ideas to evolve and be refined

1



2



3

Jeff Patton - 2008

Agile in today's industry

Agile success stories

Sony Interactive Entertainment

- Delivered double the value compared to before practicing Agile
- initial planning time cut by 28 %

KANTAR RETAIL

- 60% faster time to market
- 28-41% decrease in attrition rate

Zappos.com

- Halocracy: self organized and managed company (no more permanent managers)

American Airlines

- Business Agility :abality to embrace market and operational changes as a matter of routine

Spotify

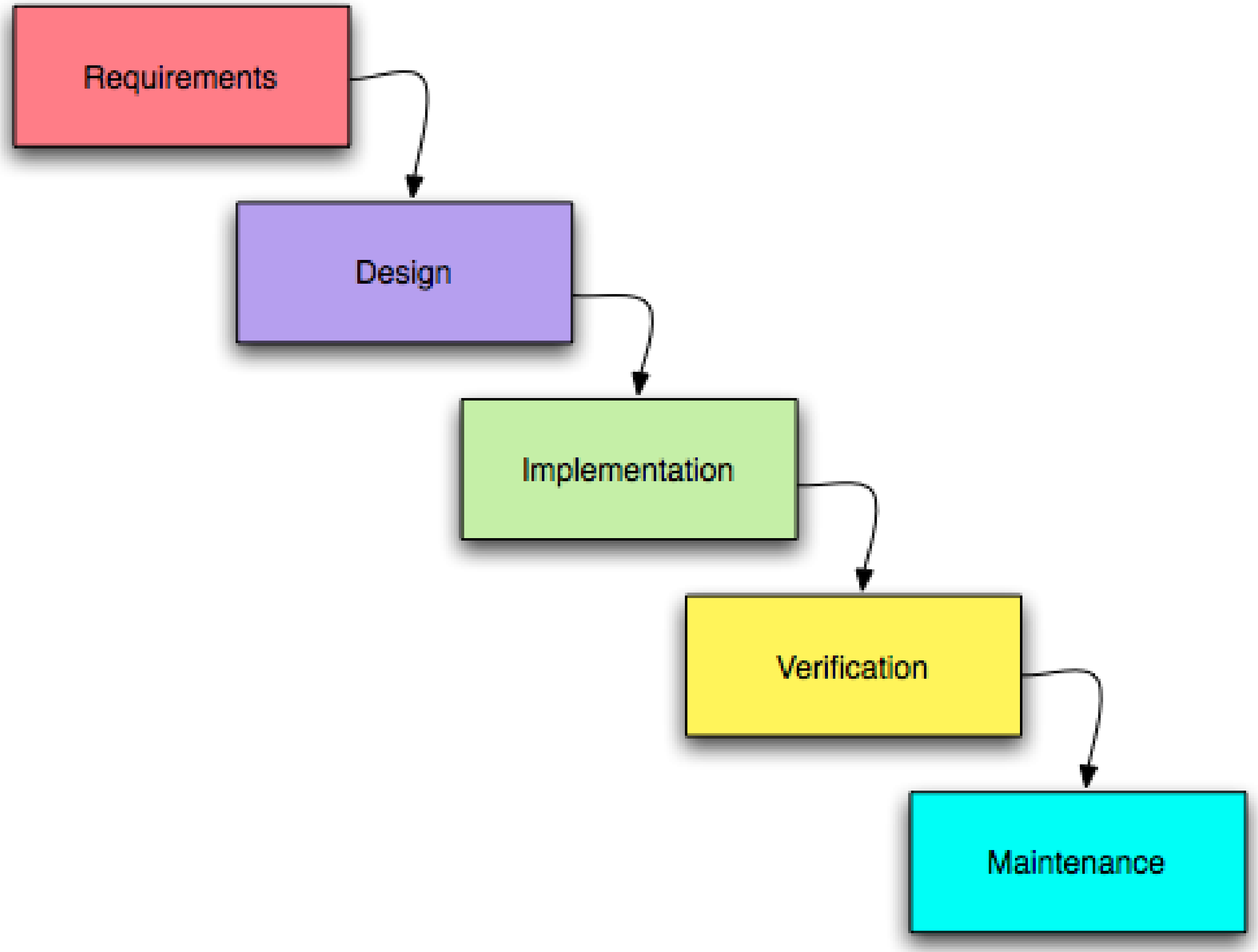
- Scaled Agile framework invented by Spotify
- Self organized company

TESLA

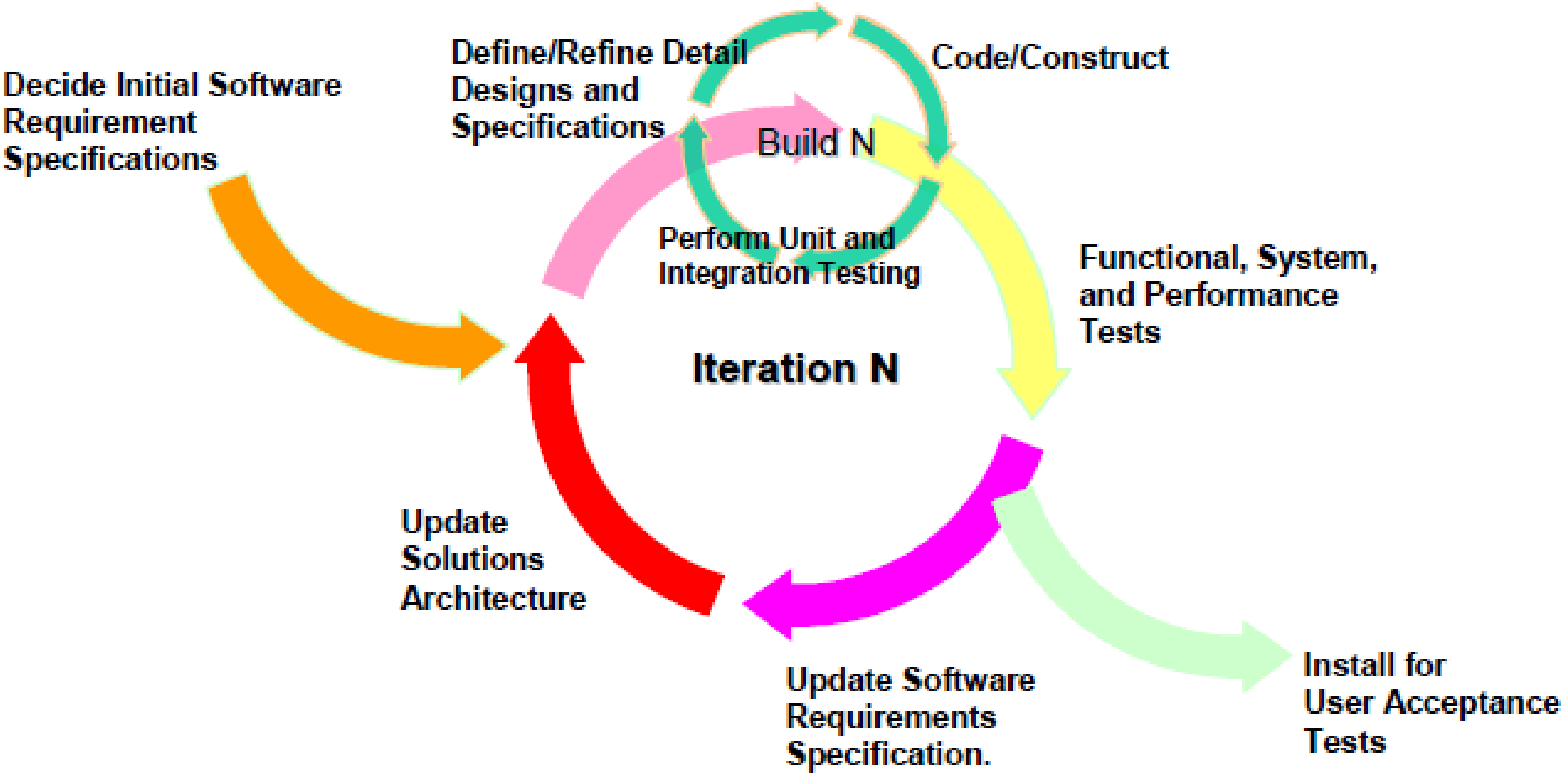
- Tesla: The agile automobile

Project Time Management
Lifecycles Models

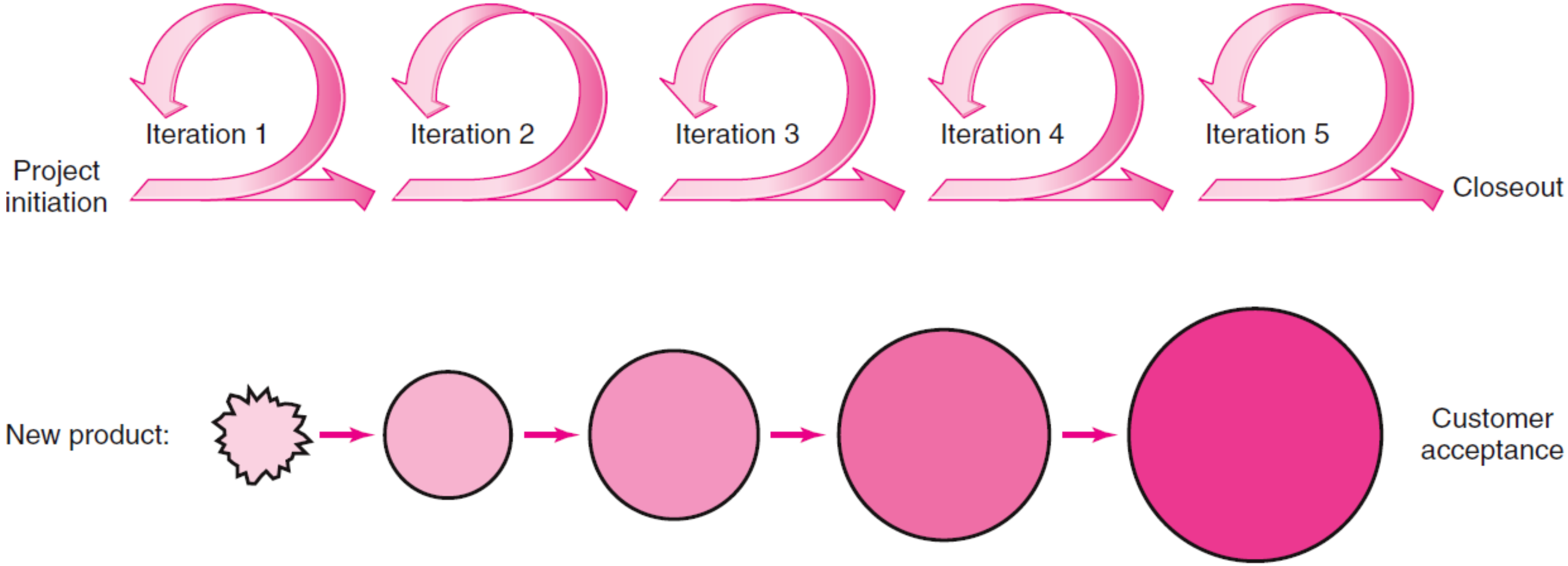
Waterfall Model



Iterative and Incremental Model (e.g. RUP, Agile)


Decide Initial Software Requirement Specifications
Define/Refine Detail Designs and Specifications
Code/Construct
Build N
Perform Unit and Integration Testing
Functional, System, and Performance Tests
Iteration N
Update Solutions Architecture
Update Software Requirements Specification.
Install for User Acceptance Tests

Agile: Iterative, Incremental Product Development



Traditional PM versus Agile Methods (1)

Advantages of Agile PM:

- Useful in developing critical breakthrough technology or defining essential features
- Continuous integration, verification, and validation of the evolving product.
- Frequent demonstration of progress to increase the likelihood that the end product will satisfy customer needs.
- Early detection of defects and problems.

Focus on customer value

Iterative and incremental delivery

Experimentation and adaptation

Self-organization

Continuous improvement

Traditional PM Approach

- Concentrates on thorough, upfront planning of the entire project.
- Requires a high degree of predictability to be effective.

Traditional PM versus Agile Methods (2)

Traditional	Agile
Design up front	Continuous design
Fixed scope	Flexible
Deliverables	Features/requirements
Freeze design as early as possible	Freeze design as late as possible
Low uncertainty	High uncertainty
Avoid change	Embrace change
Low customer interaction	High customer interaction
Conventional project teams	Self-organized project teams

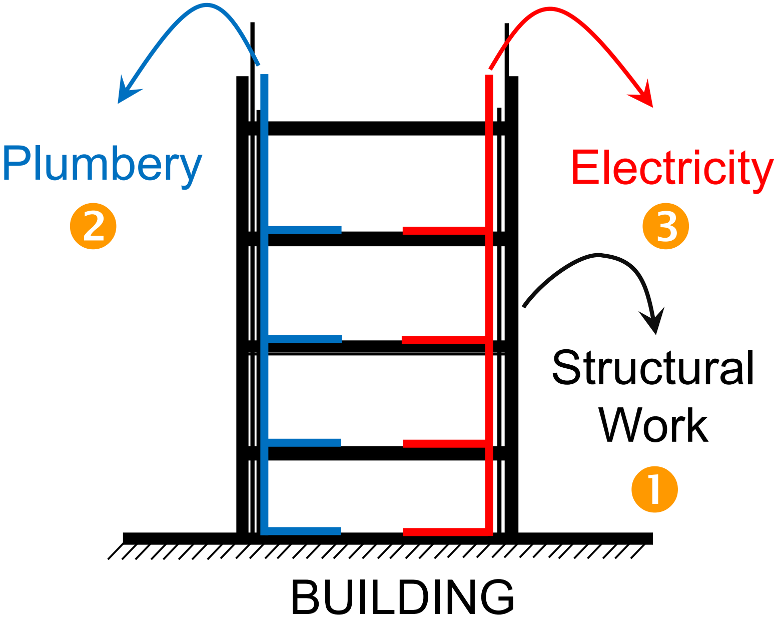
Illustration of the different lifecycle models

Project Approach:

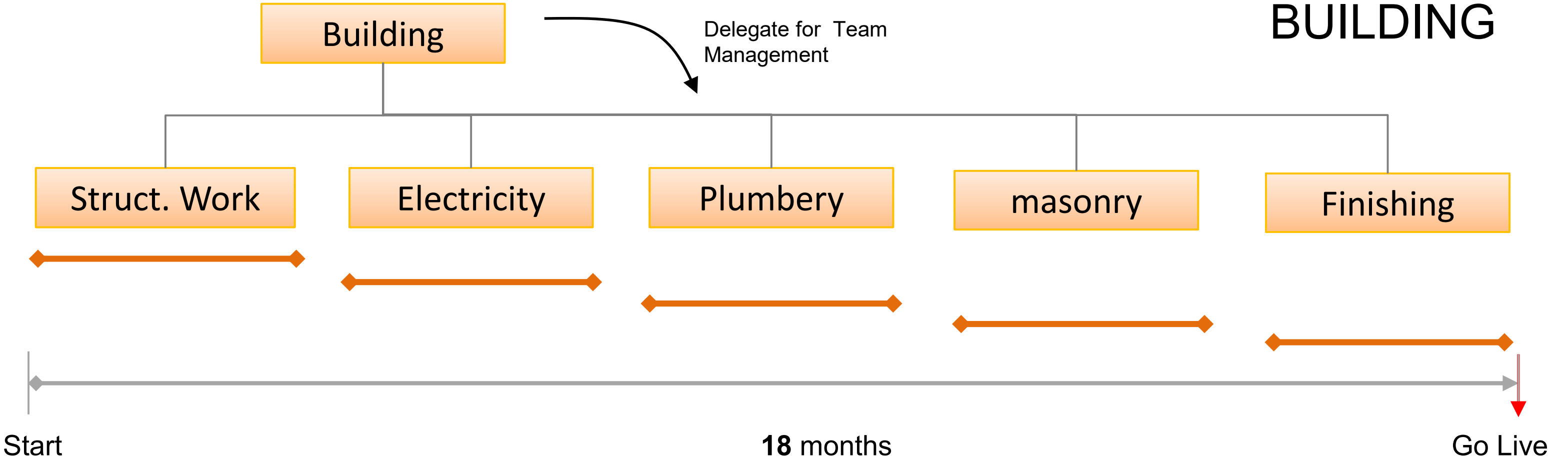
Example 1: Optimize costs and available capacity

•Objectives:

- Maximize capacity / skill usage
- Optimize costs
- Reduce overall planning duration



Project Structure & Planning:



Building

Delegate for Team Management

Struct. Work

Electricity

Plumbery

masonry

Finishing

Start

18 months

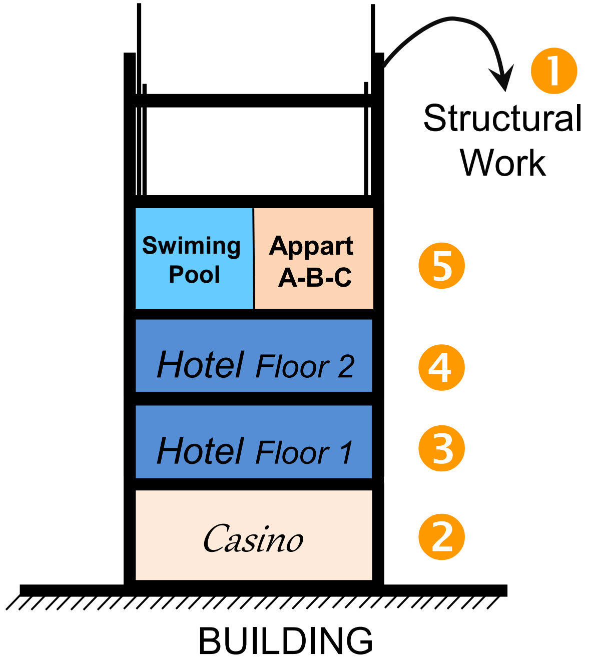
Go Live

Project Approach:

Example 2: Maximize ROI and Time to Market

•Objectives:

- Maximize ROI
- Optimize product value
- Optimize time to market



1 Structural Work

Swiming Pool

Appart A-B-C

5

Hotel Floor 2

4

Hotel Floor 1

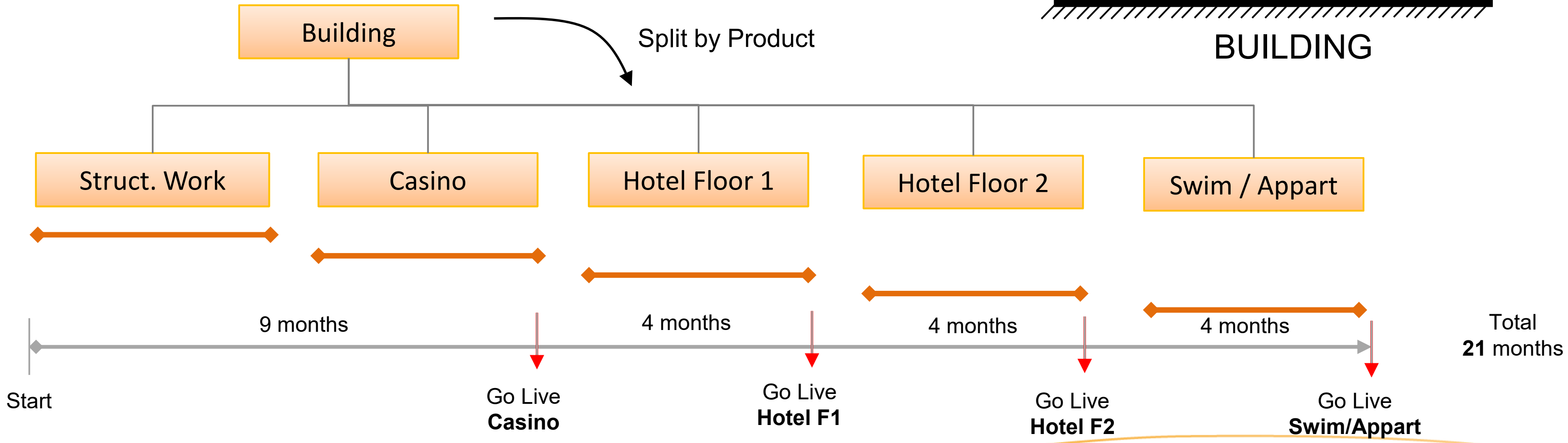
3

Casino

2

BUILDING

Project Structure & Planning:



Building

Split by Product

Struct. Work

Casino

Hotel Floor 1

Hotel Floor 2

Swim / Appart

9 months

4 months

4 months

4 months

Total **21** months

Start

Go Live **Casino**

Go Live **Hotel F1**

Go Live **Hotel F2**

Go Live **Swim/Appart**

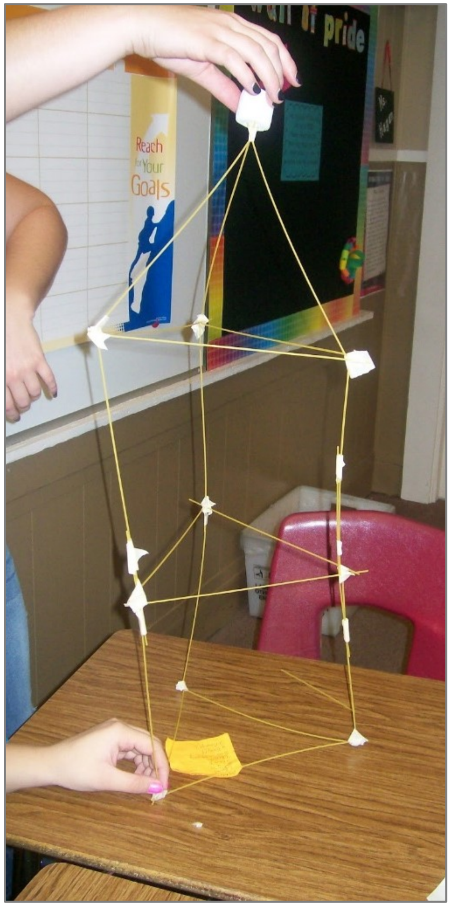
Games

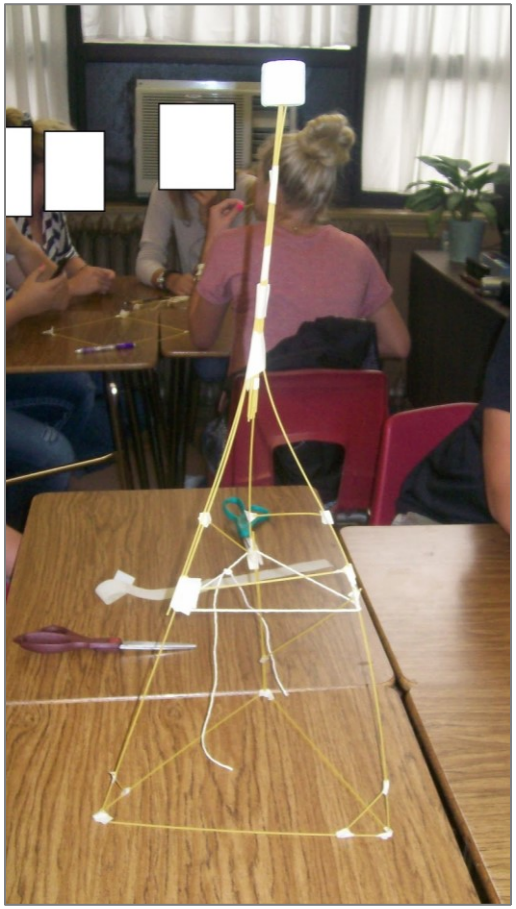
Marshmallow Challenge

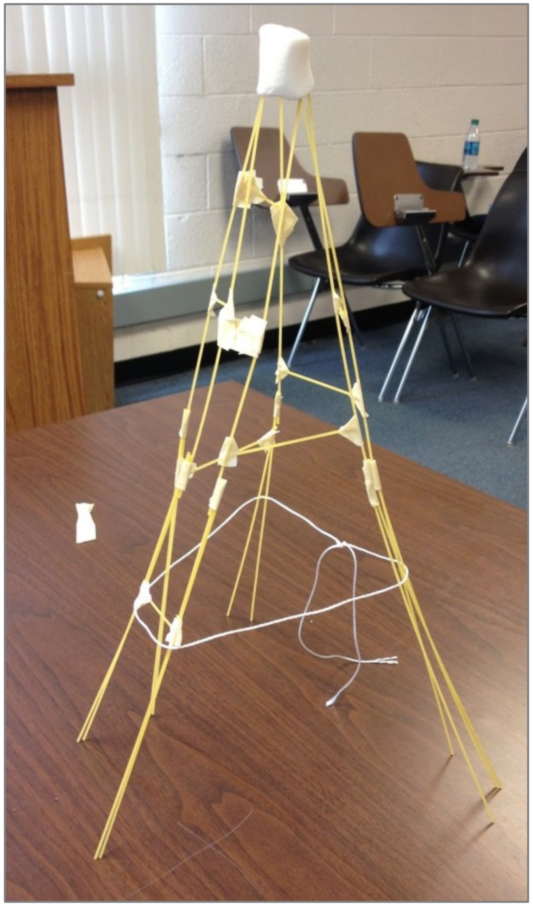
In 18 minutes, build the tallest free-standing structure out of 20 sticks of spaghetti, 3 feet of tape, 3 feet of string, and one marshmallow. The marshmallow must be on top.

20 sticks of spaghetti + one yard tape + one yard string + one marshmallow

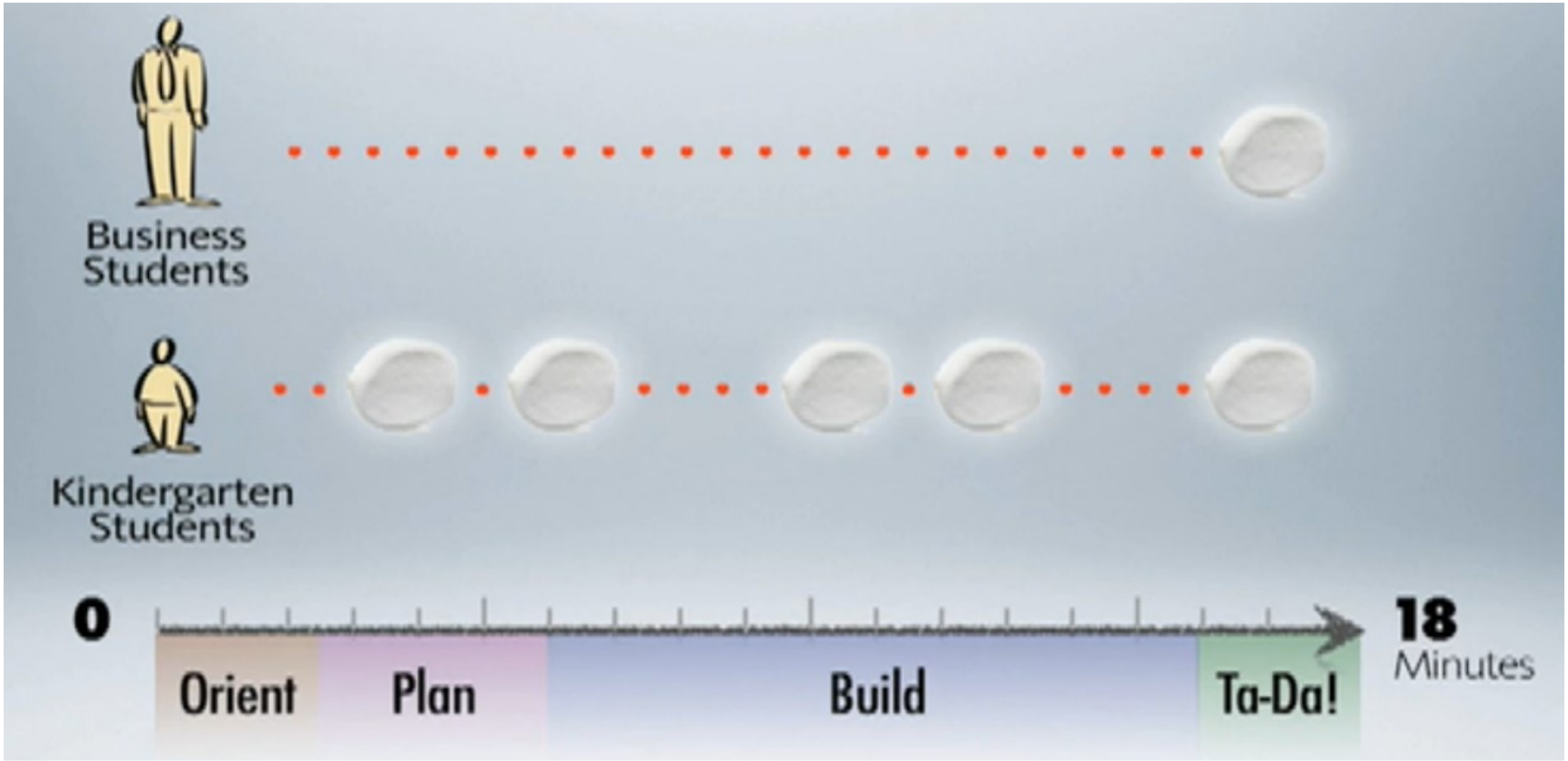
Marshmallow Challenge: in practice







Marshmallow Challenge: Lessons learned

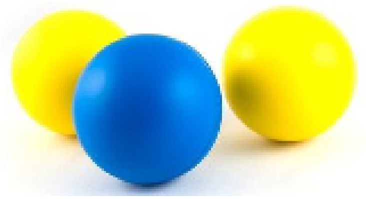


Agile Scrum Ball Point Game

- **Objective:**
  - The point of this game is to pass as many balls as possible through every team member in 2 minutes.
  - The team gets a point for each ball passed through every member of the team provided that the first person to touch that ball is also the last.

- **Rules:**
  - You are one big team
  - Ball must have air-time
  - No ball to your direct neighbor
  - Start Point = End Point
  - Iteration = 2 min
  - In between = 1 min
  - We play 5 iterations

- **Run 5 iterations:**
  - Estimation **1 min**
    (how many balls will you pass?)
  - Iteration **2 min**
  - Record result
  - Team plans improvements **1 min**



Debrief Points (1)

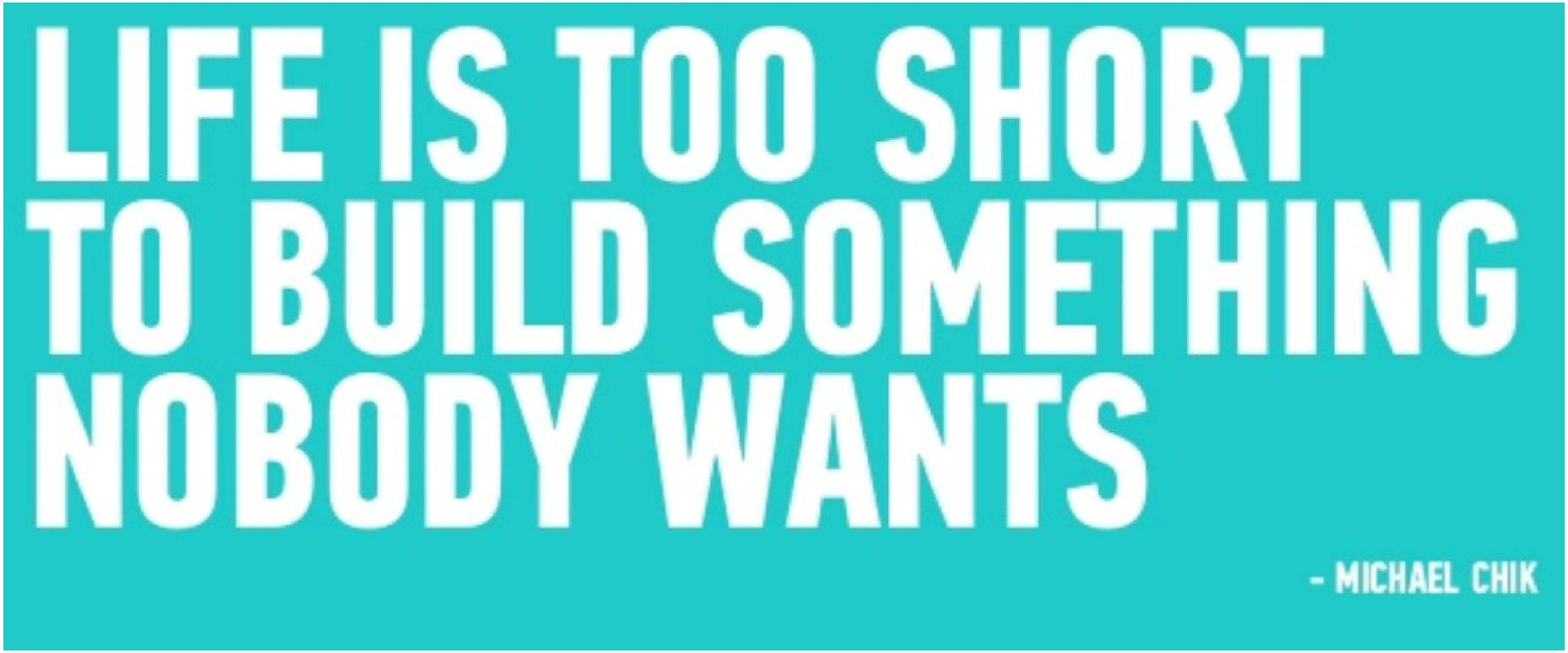
- **What Happened?**
  Ask with an open question and allow people to share their experiences. I then guide the remainder of the debrief around the remaining points depending on what I noticed, team interest, and time.
- **What Iteration Felt the Best?**
- **Ask what made that one feel best? Why?**
- **Natural Velocity**
  Look for an iteration where there was a dramatic improvement due to a reorganization. Ask whether this improvement was due to working faster or harder. Point out that every system has a natural velocity and that to significantly improve the system requires changing the process.
- **Theory of Constraints**
  Ask if there is a bottleneck in the system and if so where it was. Ask how they identified and whether they attempted to address it. Ask if it would be helpful to improve efficiency at a point other than the bottleneck.
- **Experiments Sometimes Fail**
  Often a team will try something that will not result in an improvement and performance may actually drop. Point out that this is ok and expected. Sometimes we try a good idea and it does not work out. We don’t have to get it right every time and we learn more from our failures.
- **Waste**
  If you acted as the manager and this was not challenged then ask why they did not challenge this. Point out that is was not a rule and that we must be careful to challenge constraints and identify waste if we are to achieve maximum performance. Negotiate with your management and customers – we all win!
- **Rhythm and Flow**
  Do you experience a rhythm? Point out that flow will happen if:
    - the challenge is doable
    - people are not disturbed during the iteration
    - the work has meaning

  You may want to point out that agile development has multiple levels of rhythm: daily standups, sprints, releases.

Debrief Points (2)

- **Pull Systems**
  Most teams will put in place a system where the balls are not passed until the downstream person is ready – i.e. a pull system. When balls are pushed it usually results in dropped balls and lack of flow. Point out how a pull system maximizes flow and increases performance.
- **The Power of the Retrospective**
  Ask them if they had 6 minutes (total planning time) to plan for a single 2 minute run do they think they would have achieved the same results?
- **The Power of Face to Face Communications**
  Could the same improvements have been achieved via phone or email? Would it have helped to document the process?
- **Heroes**
  Would it have helped if someone on the team was much better at this game than the rest of the team?
- **Whole Team and Leadership**
  You may notice some good suggestions were made but not followed. Ask the person how that felt. Is there a natural leadership model on the team? Is there a "right" leadership model? A post on Babble burble banter balderdash suggests asking the following questions:
  - Who had all the ideas?
  - What roles did you all take?
  - When something went wrong what did you do?
- **Stretch Goal**
  If you introduce a stretch goal after the third iteration ask how that felt and the impact it had. Often, teams will perform better knowing that improvements are possible. And improvements are always possible!
- **Impossible Goal**
  According to Kane Mar setting an impossible goal often results in decreased performance.
- **How Does this Apply to Us?**
  Close the game by ask the team if there is anything they can take back into their development team. Record on a flipchart.5 iterations:

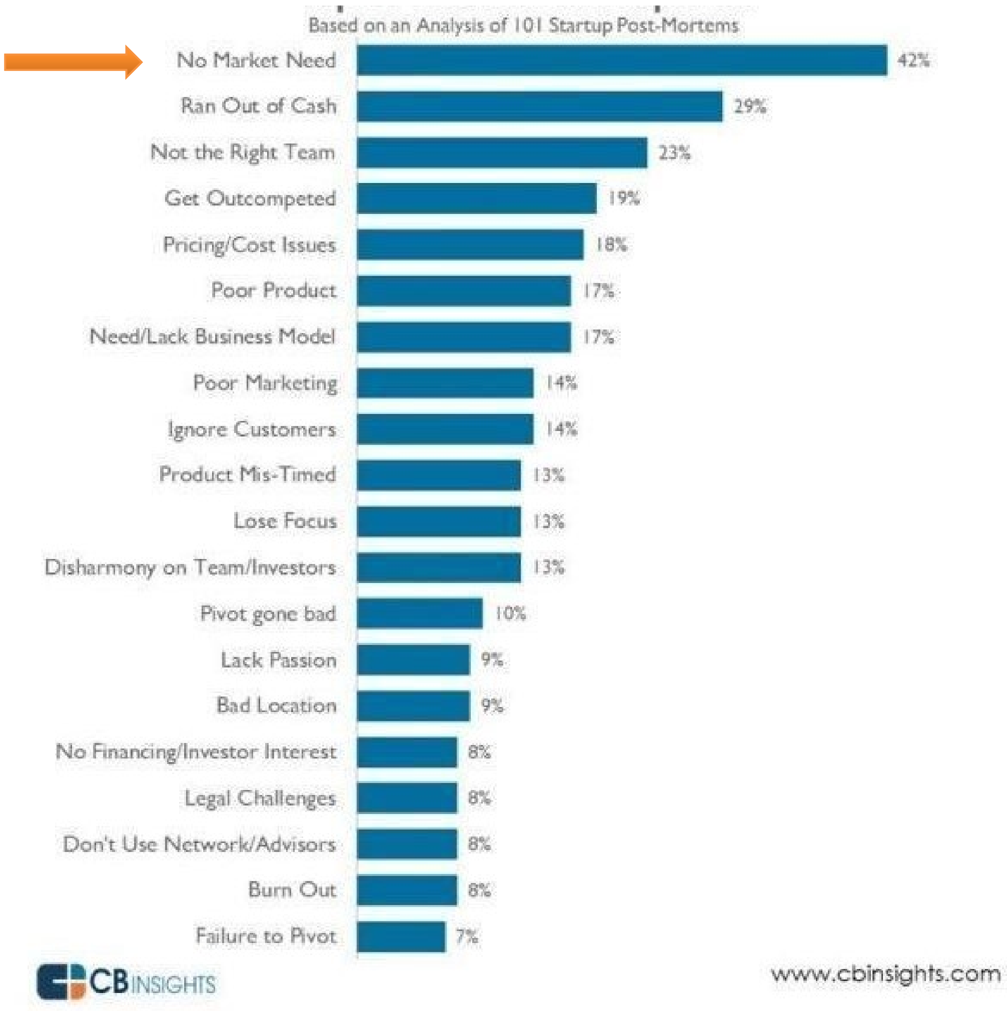
Lean StartUp basics

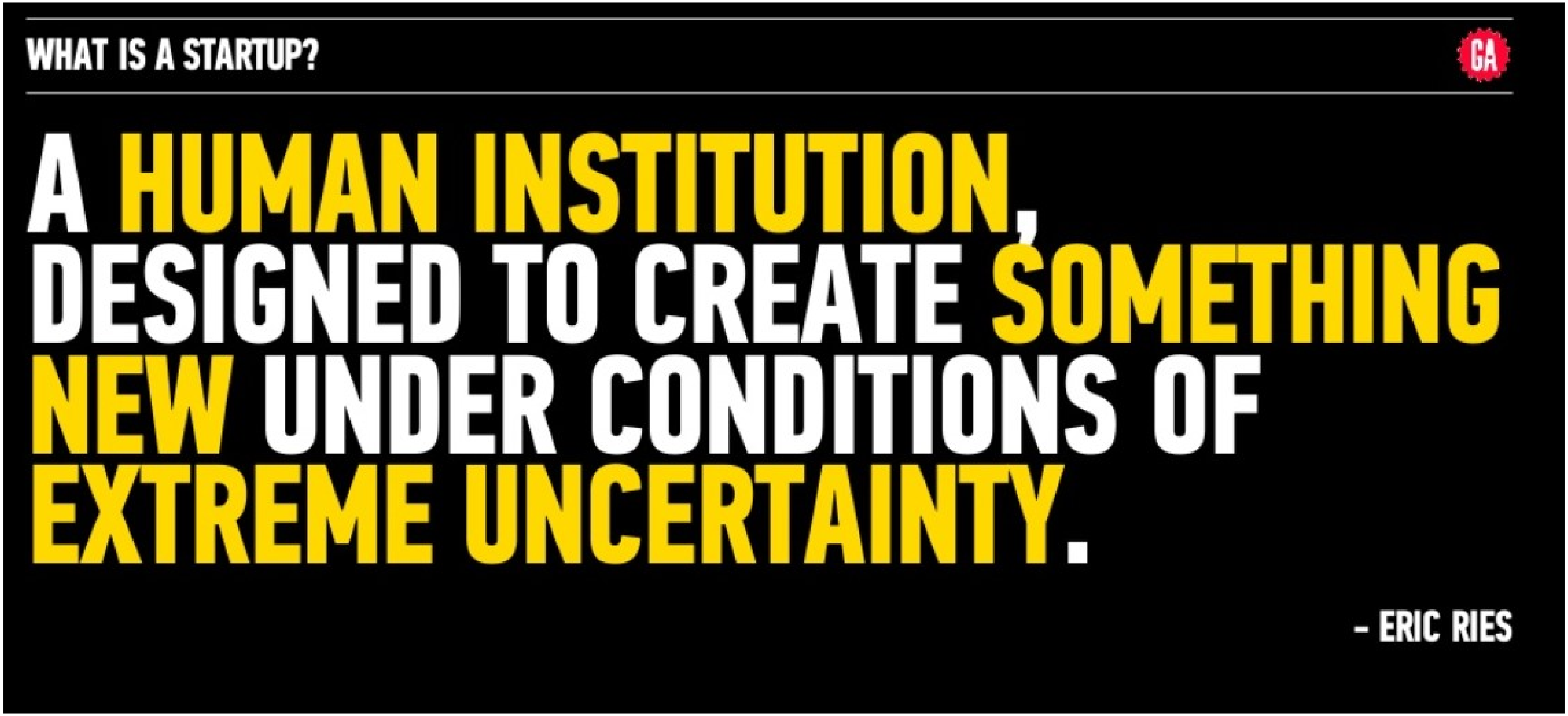


LIFE IS TOO SHORT TO BUILD SOMETHING NOBODY WANTS

- MICHAEL CHIK

Top 20 reasons Startups fail

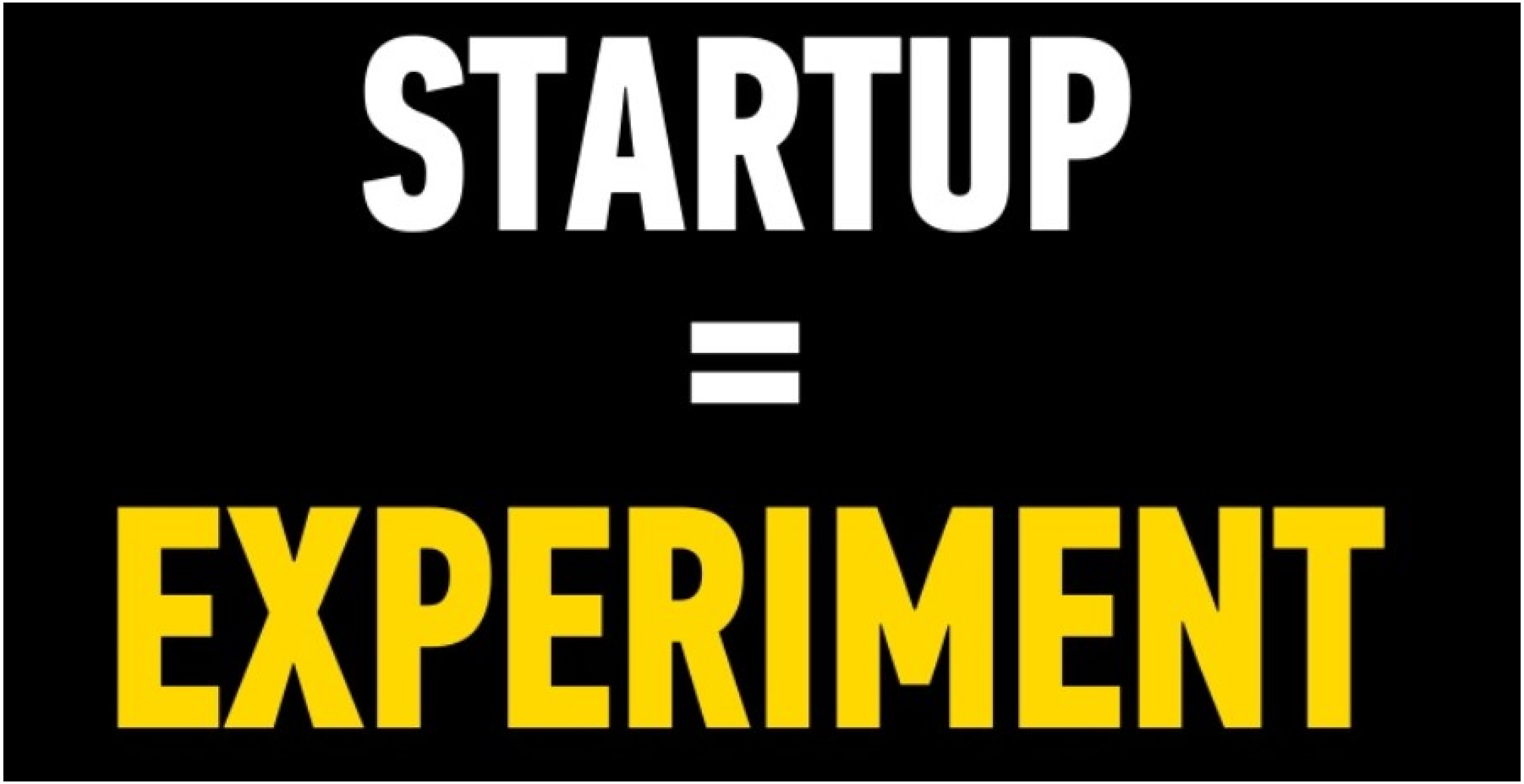




WHAT IS A STARTUP?

A HUMAN INSTITUTION, DESIGNED TO CREATE SOMETHING NEW UNDER CONDITIONS OF EXTREME UNCERTAINTY.

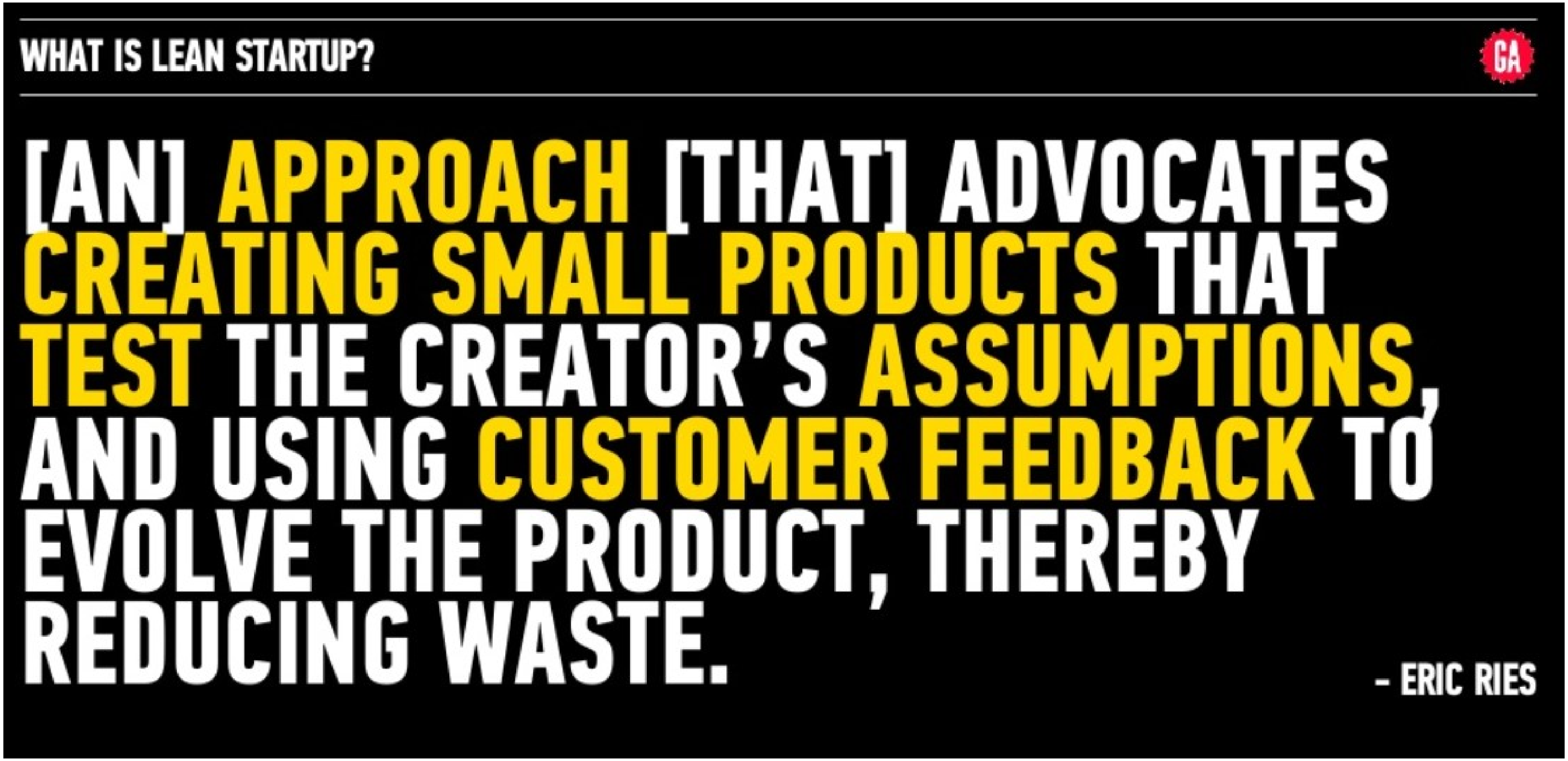
- ERIC RIES



STARTUP

=

EXPERIMENT



WHAT IS LEAN STARTUP?

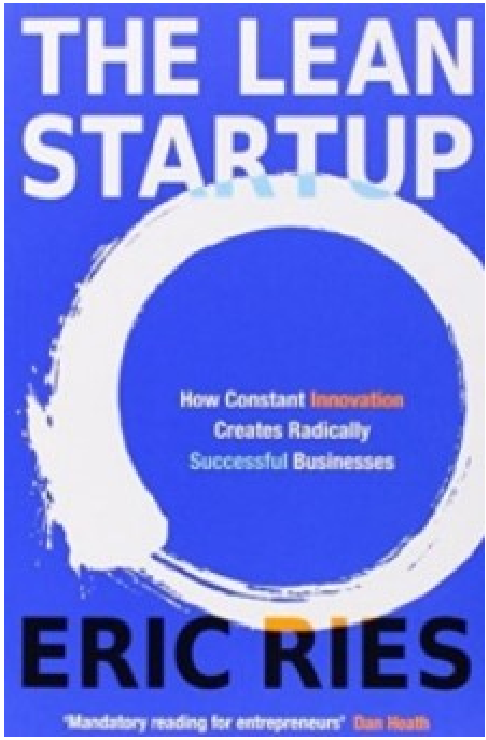
GA

[AN] APPROACH [THAT] ADVOCATES CREATING SMALL PRODUCTS THAT TEST THE CREATOR'S ASSUMPTIONS, AND USING CUSTOMER FEEDBACK TO EVOLVE THE PRODUCT, THEREBY REDUCING WASTE.

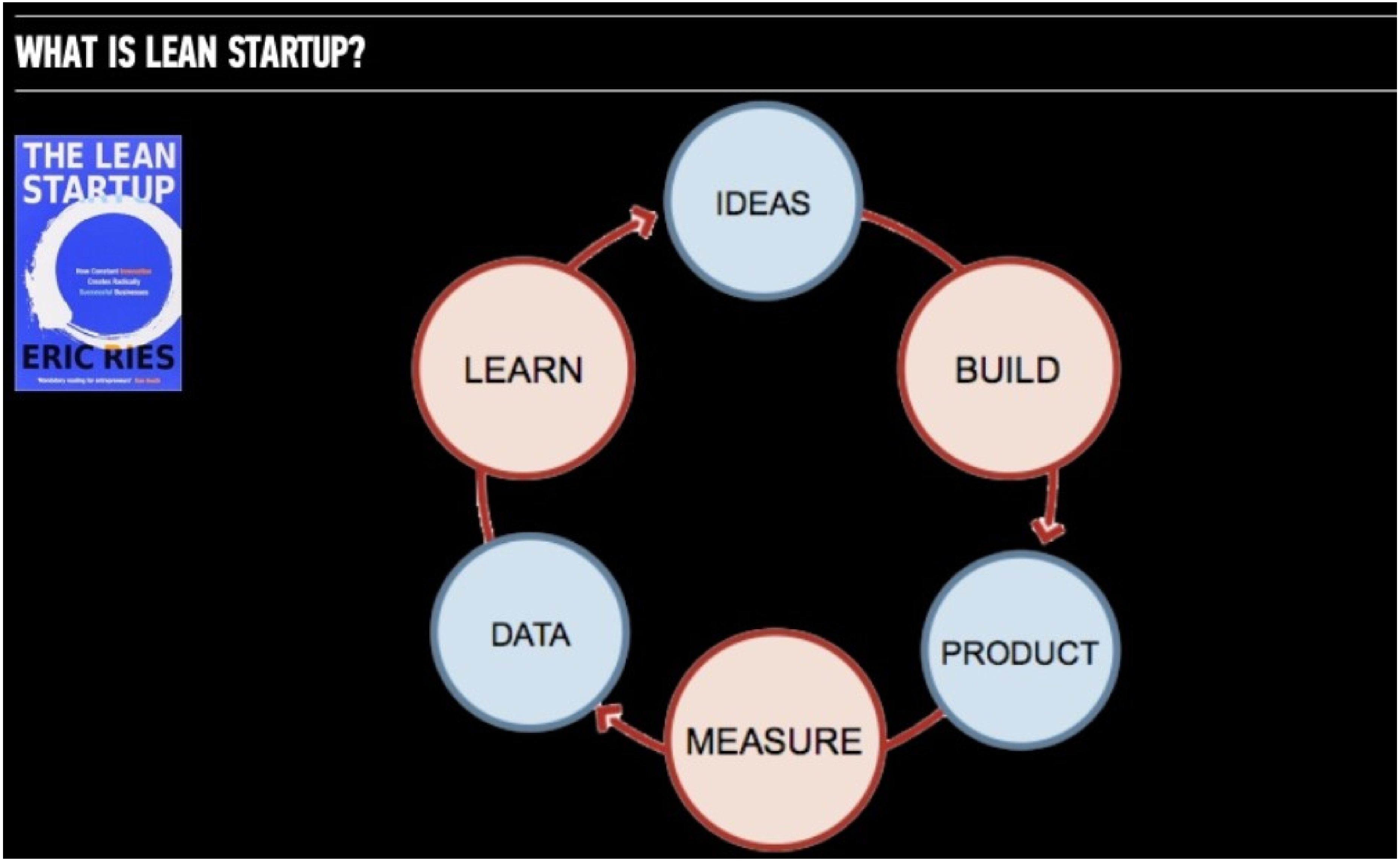
- ERIC RIES

What is lean startup?

1. LIST YOUR ASSUMPTIONS
2. UNDERSTAND YOUR CUSTOMERS
3. GET A REAL PRODUCT INTO THE WORLD
4. ADJUST DIRECTION BASED ON EVIDENCE


THE LEAN STARTUP
How Constant Innovation Creates Radically Successful Businesses
ERIC RIES
'Mandatory reading for entrepreneurs' Dan Heath

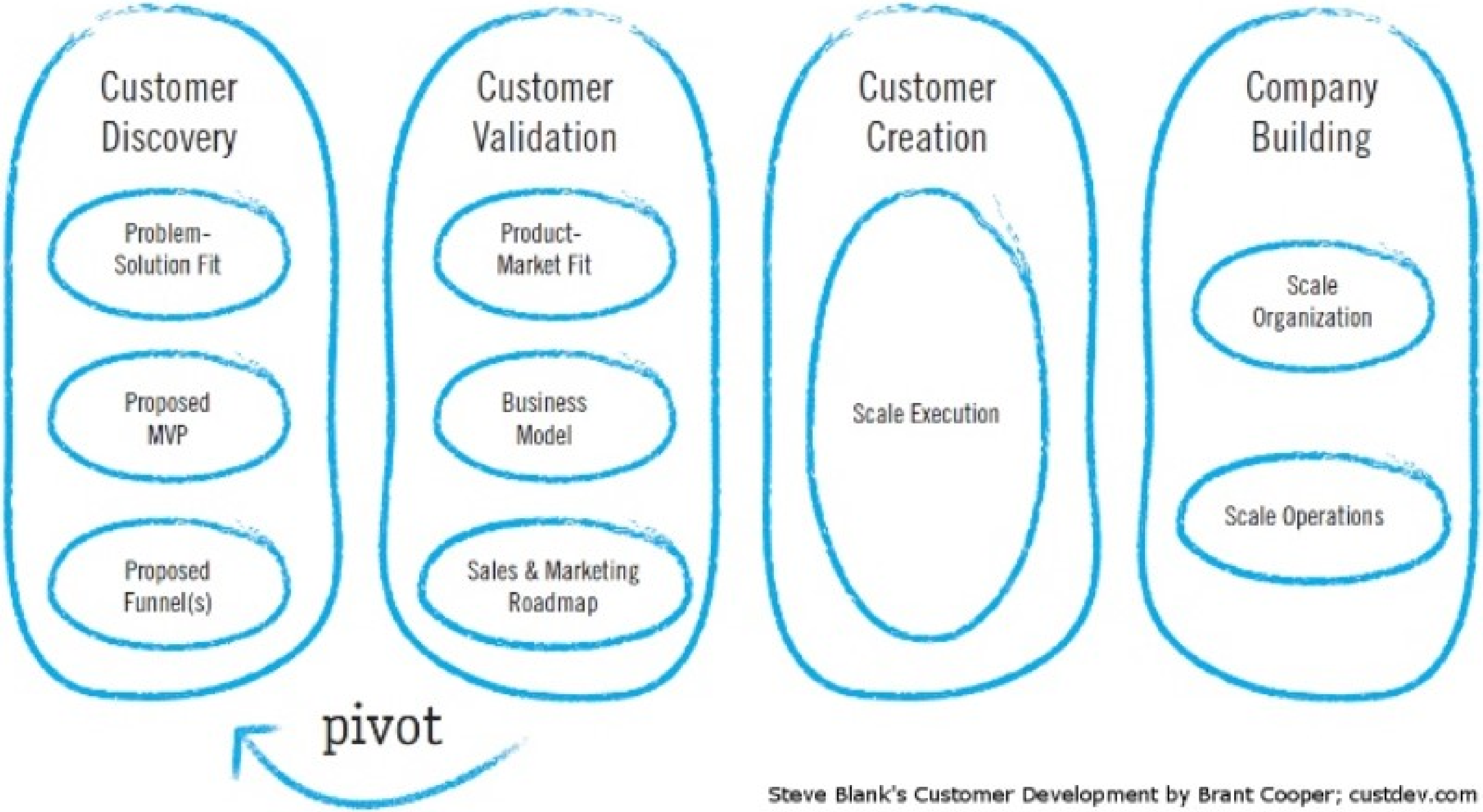
STARTUP
MVP
HYPOTHESIS
PIVOT
PROBLEM-SOLUTION FIT
PRODUCT-MARKET FIT



WHAT IS LEAN STARTUP?

THE LEAN STARTUP
How Constant Innovation Creates Radically Successful Businesses
ERIC RIES
IDEAS
BUILD
PRODUCT
MEASURE
DATA
LEARN

Where to start



Steve Blank's Customer Development by Brant Cooper; custdev.com