




STARTING TO USE BUSINESS VALUE ENGINEERING

RTP, Oct 28, 2010

BV Engineering is...

- the values, principles and practices that enable us to deliver more and more Business Value [from a given team] as we improve.
- a learning and incremental improvement approach to giving customers more of what they really want, looking at the whole process, end-to-end.
- a framework for getting better.

Attributions

-  **Some people who directly or indirectly contributed: Peter Drucker, Takeuchi & Nonaka, Jim York, Chris Matts, Kent McDonald, Womack & Jones, Jeff Sutherland, Kent Beck, Mary & Tom Poppendieck, Taiichi Ohno, Ken Schwaber, some friends at “a large financial institution in Virginia”, and many others.**

Joe Little, CST & MBA

- 🕒 Agile Coach & Trainer
 - 🕒 20+ years in senior level consulting to well-known firms in New York, London and Charlotte
 - 🕒 Focus on delivery of Business Value
 - 🕒 CST, CSP, CSM
 - 🕒 Was Senior Manager in Big 6 consulting
 - 🕒 Head of Kitty Hawk Consulting, Inc. since 1991
 - 🕒 Head of LeanAgileTraining.com
 - 🕒 Started trying to do [Agile] before reading The Mythical Man-Month
-
- <http://agileconsortium.blogspot.com>
 - jhlittle@kittyhawkconsulting.com



A Start

"You've got to be very careful if you don't know where you're going, because you might not get there." Yogi Berra

"Some people, if they don't already know it, you can't explain it to them." Yogi Berra

My main stance

“I’m mad as hell and I’m not going to take it anymore.” (movie quote)

“The biggest thing to fix is how we do BV Engineering.”

What is the framework composed of?

- Identify what BV is
- A mapping of the BV Engineering process, including all the tools and techniques we use to increase BV delivered.
- The assumptions or theories that underlie the usage of that process and those tools and techniques.
- A BV model (this is a mathematical modeling of the BV definition).
- An approach to improving BV Engineering.
- By checking the BV Model against reality, we are proving whether the process or theories or the Model is correct.

Identify what BV means

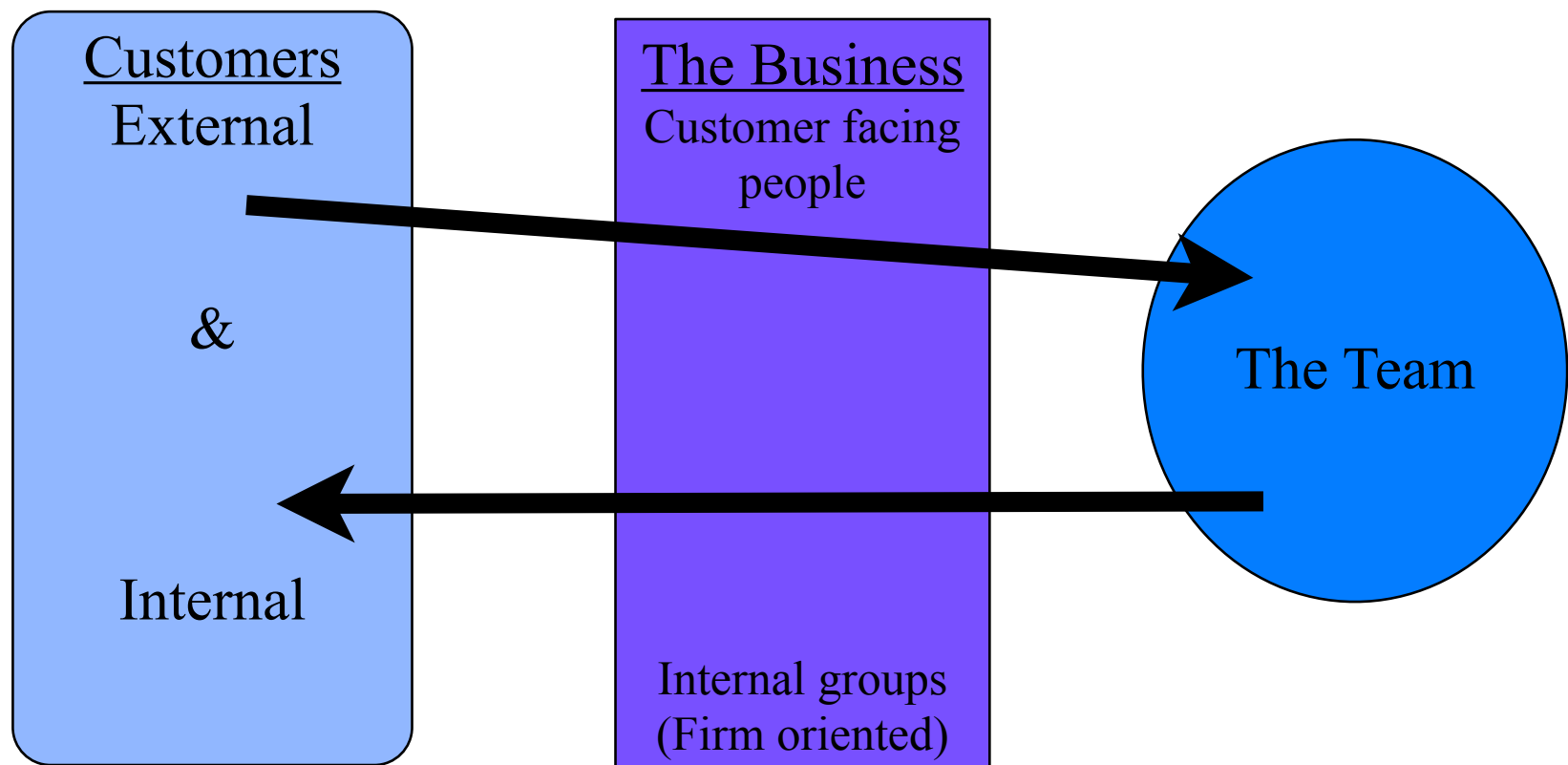
- **Describe what BV is for a given project / situation**
- **Typically BV is 2 or 3 different things for a given project, eg, NPV, risk and customer satisfaction**
- **Start to describe BV deltas to be obtained from the next product release**

BVE Mapping

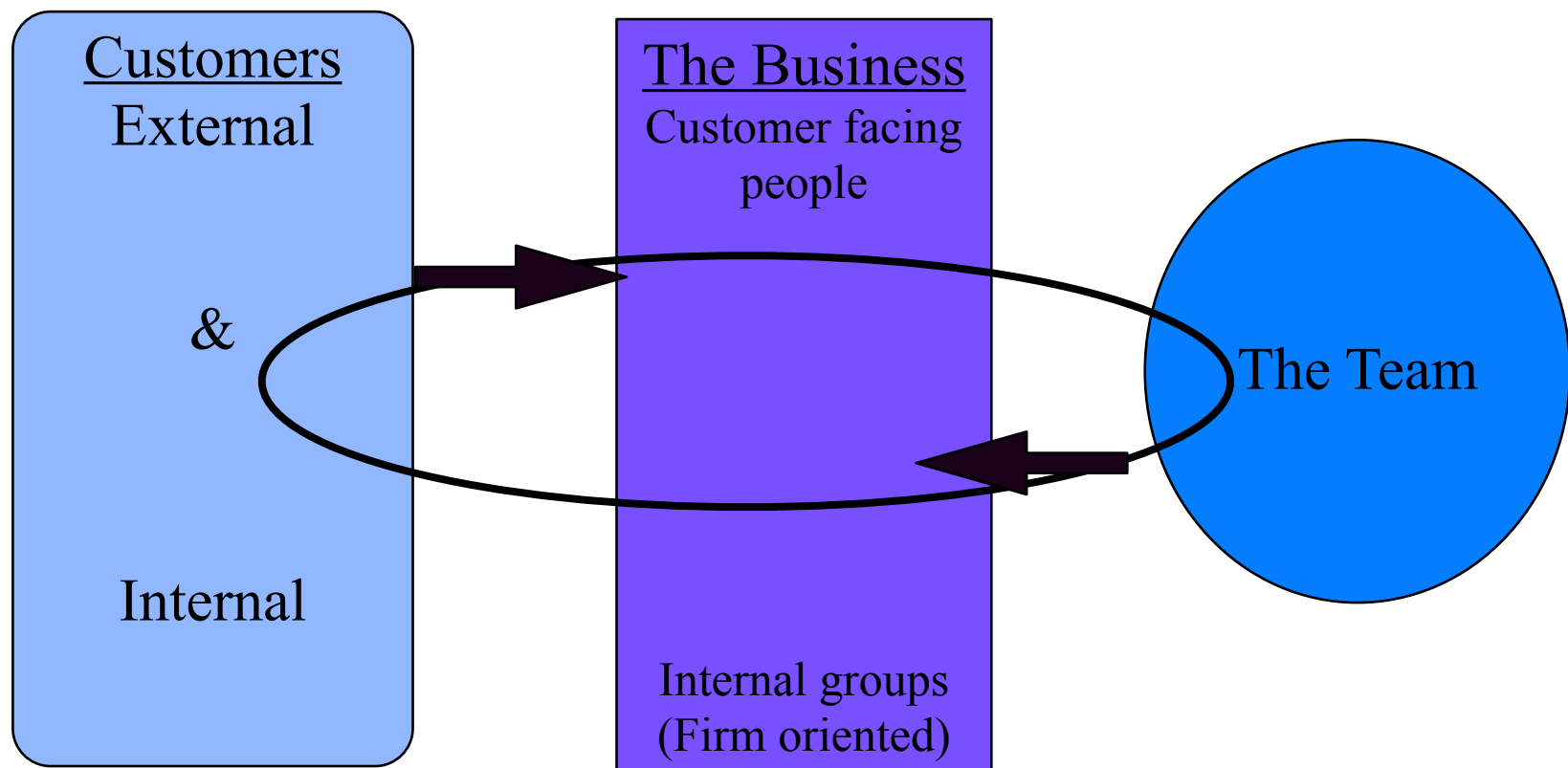
- Like Value Stream mapping in Lean
- Map about 20-30 steps currently use to deliver BV
- Include the main steps and tools and techniques
- Do it in a small team, quickly (1-2 hours)
- Identify the biggest problems

What the Product Owner does




BV Engineering



Is it better this way?



List assumptions/theories

-  **Make a list of the related assumptions and/or theories**
-  **Identify which seem to be the weakest theories**
-  **Try to prove that the 'weaker' theories are correct or incorrect.**

BV Model

- **Identify the main drivers (NPV, Risk, customer SAT, whatever....)**
- **Identify the specific factors for each driver in the model**
- **Identify the formulas (calculations) to be made**
- **Identify reasonable assumptions for the factors.**
- **Calculate the expected BV total.**
- **Review how/when to revise the model.**

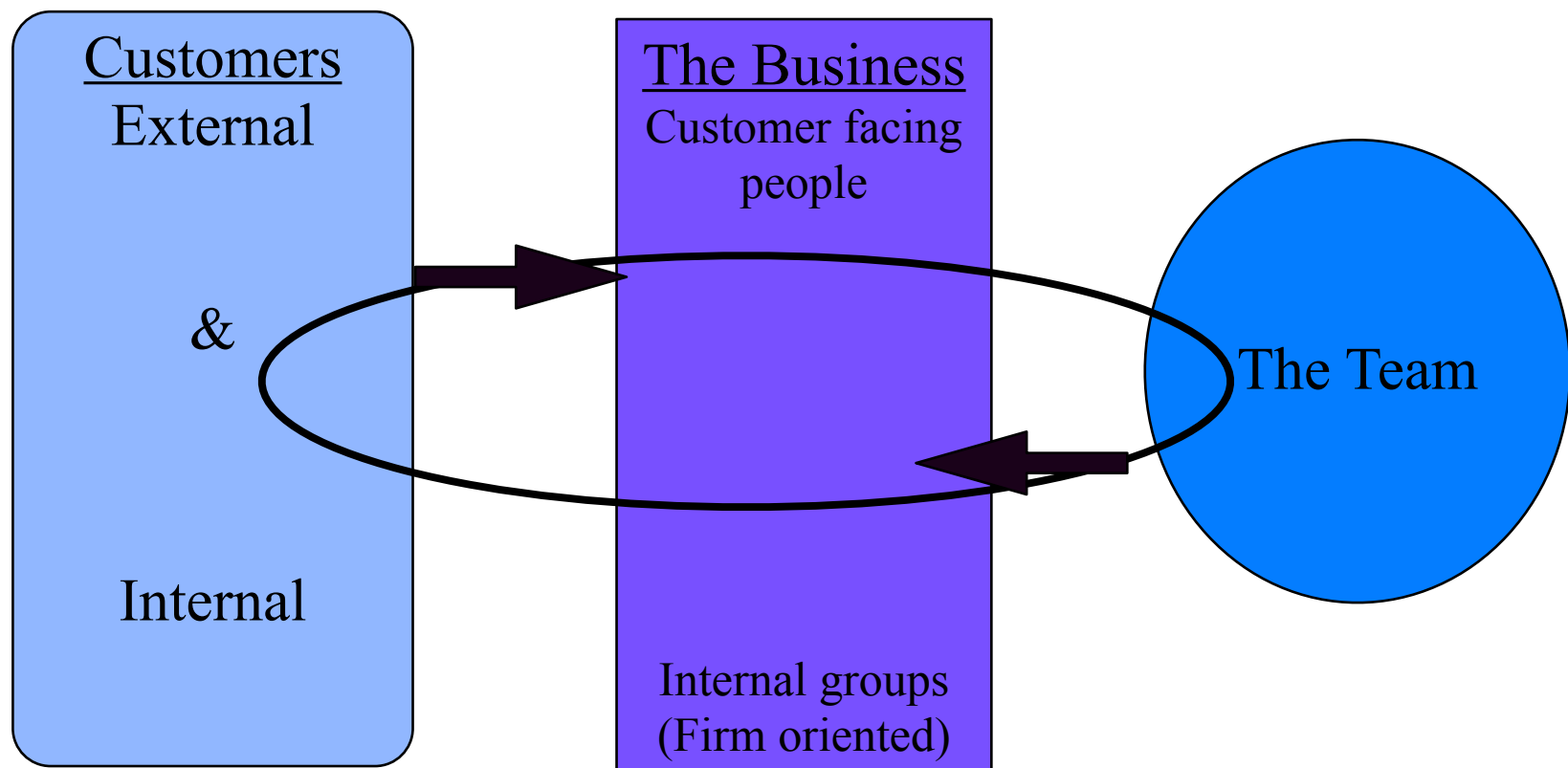
Improvement approach

- **Identify the improvement approach to be used.**
- **We recommend a review of BV during each sprint for a Scrum team.**

Testing

- **Identify how BV will be tested (for a given project).**
- **Then conduct such tests, in a P-D-C-A approach.**

Making more sense?



Hallmarks of real BV Engineering!

- 1. The process is visible and articulated & improved**
- 2. Failures in BV communication are identified and corrected frequently, quickly**
- 3. There is a theory, and a concerted attempt to prove out the theory**
- 4. There is appropriate dynamism and change**
- 5. Business & Technology are partners**
- 6. Success is forecast and also measured after the fact**
- 7. Human judgment is involved (it's not just the numbers)**

Scope of BVE

- **Everything that happens after a customer need is identified....**
- **...and until it is satisfied.**
- **Alternately: Anything that may be said to delay delivery of satisfaction to the customer.**
- **Normally we ignore what goes on in the Scrum team (the purview of Scrum).**

Scope of BVE - sample specifics 1

- 🌐 **Searching for unmet customer needs**
- 🌐 **Identifying customer feature requests**
- 🌐 **Determining if a product will be profitable**
- 🌐 **Ranking alternate projects (agile portfolio management)**
- 🌐 **Re-evaluating product viability monthly, as things change**
- 🌐 **Marketing the product (pricing, advertising, etc)**

Scope of BVE - sample specifics 2

- **Confirming that customer needs are met**
- **Assuring quality for customer**
- **Making it easy for customer to accept product (eg, UAT is not onerous, integration at customer site is less complex)**
- **Delivering customer needs while still 'hot'**
- **Being responsive to bug fixes and small priority enhancement requests**

Notes

- **Yes, doing some of these things happens (partially) within the Scrum team process**
- **AND...some of this happens (partially) outside the Scrum team**

ACTION // DISCUSSION

- **In teams of 5.....for 4 minutes....**
- **Describe how your team/project thinks of BV**
 - One or multiple
- **How much is the team 'into' this BV Def?**
 - Fully, so-so, or very cynical
- **How much time should we spend to explain BV to the team? (For what size project?)
What are the benefits of explaining it to them?**

Something useful (finally!)

- **When you hear a new idea about BVE, use the framework to ask yourself: Is this the area that we need to improve on the most?**
- **If yes, then go to X's presentation (or whatever).**
- **If no, go where you need the new knowledge the most.**

Start Exercise #1

Real Situation

-  **It must be a real situation....at least to the “Product Owner” of the group.**
-  **All others help PO describe his/her situation.**

Exercise 1:

- As a team, you will have 5 mins.
- Pick a PO and a specific situation (project).
- Define BV in words for all those who must use it. In the context of a specific situation.
- Define the basics of your BV Model. “If x and y and z , then over 3 years we will make \$3 million from this software.” x , y , z will be variables (assumptions) in an equation.

Conditions of Satisfaction

- **Your BV definition must be compelling to key participants**
- **The linkage from the product (to be built) and your BV (to be realized) must be reasonably obvious or explained.**
- **Hint: Most teams deliver against multiple definitions of BV (eg, reduced risk and higher NPV).**

What is BV?

- **Opinion: Defined by people**
- **Which means: There are many many opinions. Which means what?**
- **Involves customers, stakeholders, even workers**
- **No one right answer for all situations**

What is Business Value?

- **ROI/NPV**
- **Reduced risk**
- **Higher customer satisfaction**
- **Higher revenues**
- **Lower costs**
- **Proxies: More eyeballs, More usage, More units sold**
- **Other: Some Lean, 6Sigma or other metric**
- **Special: Movement toward some org goal**
- **What are yours???**



Observations

- **Most projects involve multiple sets of people to whom we deliver business value**
 - Shareholders, who get profits.
 - Customer sets, who each get a type of customer satisfaction
 - How about BV to the team members?
- **Most projects involve multiple definitions of BV**

Hints

- **Decide the PO quickly.**
- **Get a real situation. Minimal abstract discussion; mostly concrete specific discussion.**
- **Argue some (that's where you learn).**
- **When you come to a fork in the road, take it. (ie, the PO has to decide when to "decide & move on")**
- **The SM is responsible for getting everyone involved, and some less involved**
- **The SM is responsible for "team mojo"**

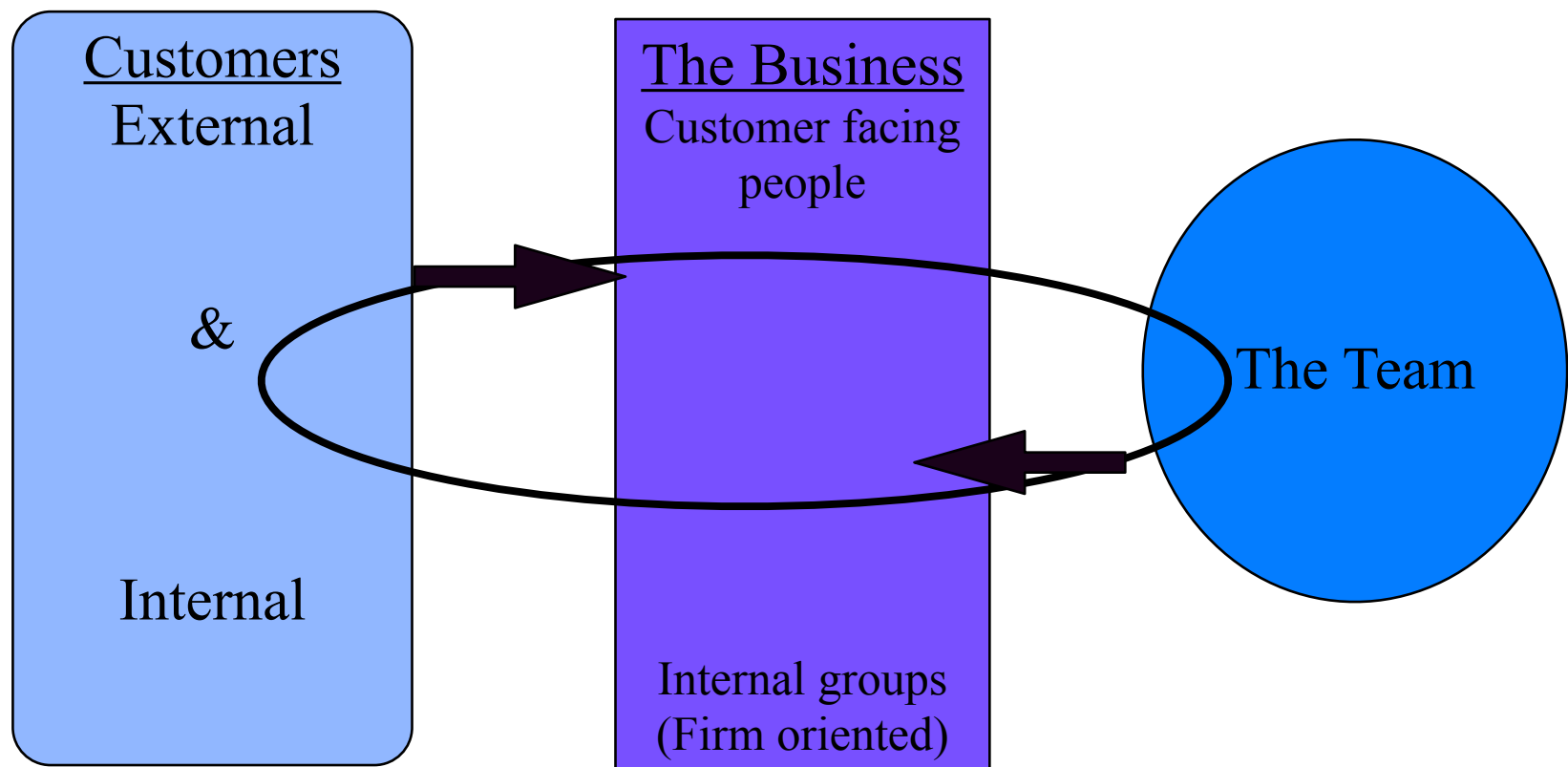
Debrief

-  **In a sentence, not repeating what someone else said, what was the (next) biggest thing you learned?**
-  **Biggest = most useful ??**

Multiple Steps are important

- **Some firms focus too much on one or two steps (eg, initial focus group, user story creation, the PO review of completed stories, the product launch)**
- **It is not one play; it is the culmination of plays that wins the game**
- **Examples: Understand the customer better and spend more time to assure that the Team understands the customer's problem better and better**

It is not one play...



Elements of BV Engineering 1

1. PO Team
2. Product Backlog
3. PB prioritized by BV
4. Priority Poker
5. Story Points (proxy for cost, for cost-benefit analysis)
6. Minimum Marketable Feature Set
7. Reprioritize before each Sprint
8. Increase velocity (remove impediments)

Elements of BV Engineering 2

1. Making the stories smaller
2. Value Stream mapping
3. Kano Analysis
4. Voice of the Customer
5. Having the team live with the customers
6. Pareto chart (eg, of causes of customer problems)
7. Process charts or high level use cases
8. Other Lean, Six Sigma, or TQM tools

Elements of BV Engineering 3

1. Understanding the importance of minimizing technical debt
2. Agile portfolio management
3. What quality means to the customer and why it is 'free'
4. Just-in-time knowledge creation
5. Modifying the BV model frequently (& the values in the model)
6. Removing impediments
7. Comparing our BV Engineering to theirs
 - 🌐 We're different; what does it mean?

Elements of BV Engineering 4

1. Identifying better sources for good user stories (eg, observation, “living with”, experts, user interaction, “prototypes”, etc)
2. Identifying good user stories
3. Fleshing out good user stories with an Agile specification
4. Improving the monetization of User Stories (or themes)
5. Improving the conversations around the user stories
6. Getting better feedback faster



Some metrics I like

- 1. NPV (net present value)**
- 2. ROI (return on investment)**
- 3. Faster end-to-end cycle time**
- 4. Increased sales**
- 5. Increased market share**
- 6. More eyeballs (on a webpage)**
- 7. Improved eyeball demographics**
- 8. Reduced costs**

More metrics I like

- 1. Reduced risk (although I prefer if this is made more concrete by being monetized...see underwriting)**
- 2. Net promoter score**
- 3. Any specific metric showing higher customer satisfaction**
- 4. Others??**

Lies, damn lies & statistics

-  It is not having numbers...
-  It is making good use of numbers (that are reasonably accurate)

Theories

-  **The next section is about the explicit, but usually implicit, theories we use to conduct BV Engineering**

Comments about the theories

- A theory is a stated or unstated way of looking at the world. Ideas that affect how we act.
- The following 5 pages is a list of theories I see people use.
- Not complete; just there to stimulate your thinking.
- Some I agree with strongly; some I disagree with strongly. (Some are stated in possibly a sarcastic way.)
- The point is to help you to discover your firm's underlying (implicit or explicit) theories.

Theories (examples) - 1

- The customer won't change her mind in (X time period).
- The customer knows what he wants.
- The customer can explain clearly what she wants.
- The customer only knows it when he sees it.
- The customer does not want software, just a solution to her problem.
- The Sales guys are the best ones to explain what the customer wants.

Theories (examples) - 2

- **No one in our firm could possibly learn about BV by using metrics.**
- **Numbers are too hard to collect, so it is better to ignore any potential benefit from them.**
- **It's really good to use documentation to convey "requirements", since we get to lose all the Tacit knowledge.**
- **The telephone game is useful in conveying requirements.**
- **It is too risky to ask the customers for feedback on unreleased products.**

Theories (examples) - 3

- **It's better to give them a low value feature that they asked for up-front than a high value feature that they identified later.**
- **What the customer wants and what the shareholders want are always aligned.**
- **All projects are equally valuable.**
- **We already put projects in priority order, so all Prod Bkg Items within an approved project are "required".**
- **It would be wrong to tell the Team the expected NPV of the effort; they might**

Theories (examples) - 4

- **Getting feedback on how bad the upfront NPV estimates are could never help us learn ...[x]**
- **I.T. is just a cost center, so projects should only be cancelled if the team is bad.**
- **There is only one kind of user: “the user”.**
- **What the customer wants is always somewhat different than what the buyer thinks.**
- **The functionality needed by customer set 1 is never in conflict with the needs of customer set 2.**
- **The BV model could not possibly change during an 18 month project.**




Theories (examples) - 5

- **“End-to-end” starts when we get the business requirements document, and ends when we hand-off the SW to the final test group.**
- **Coders don’t need to know “business value”.**
- **We should never let the coder talk to the end user.**
- **A bad [X] from team [Y] is not my problem. However bad, it could not have been better. I should just do what I am told. Then things will turn out for the best in this best of all possible worlds. [Apologies to Voltaire.]**
- **Knowledge creation and knowledge decay have nothing to do with business value.**

Theories (examples) - 5

- There are no cost-benefit trade-offs in our work. And anyway, IT's costs, always fully understood up-front, have nothing to do with delivering business value.
- The bad news gets better with age.
- It's cheaper to fix the bugs at the end than right after they are created.
- The customers prefer big releases once every two years.
- It's better to mine the silver and copper in this hole than to dig a new hole where I will gold, platinum and diamonds. After all, they said they wanted the silver and copper (and even some of this dirt).

A note on sarcasm

-  **I have made every mistake, so the sarcasm is always partly at myself.**
-  **Sarcasm is the acid that frees us from our box. Maybe a bit painful, but useful.**
-  **Even in being painful (to oneself), one also feels compassion. One has to be cruel to be kind, sometimes.**

Exercise 2.

- **Map out one specific BV Engineering process.**
- **At your table, led by one PO.**
- **Timebox: 40 mins. (4 “days” with each: 2 mins of Daily Scrum and 8 mins of real work.)**
- **Required output (see later).**

Situation

- **A real, specific situation for the Product Owner.**
- **Situation: mainly one product. (But could use other definitions.)**
- **Other team members act as consultants.**
- **Mapping only; not fixing. Current state, not future state.**

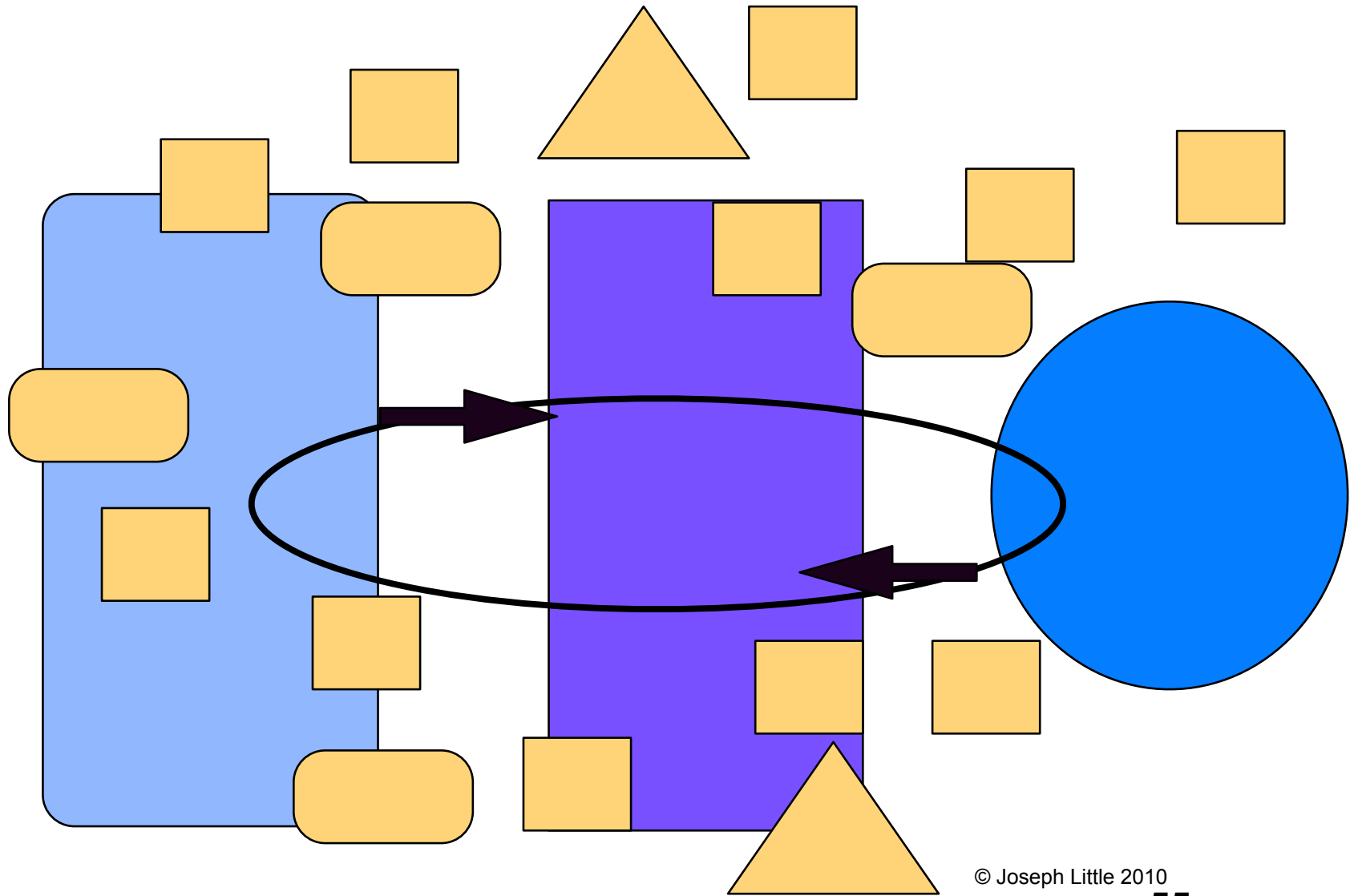
Required Output

- **A “map” (picture) of some sort**
 - Ideally end-to-end (whatever that means)
 - Show the process (at a high/medium level)
- **Describe the BV Model (better)**
- **Describe the underlying theories**
- **Describe the timeboxes and feedback loops (either in the picture or in writing)**

In describing the BV Process

- **Do it for a specific situation: a group or one team.**
- **about 10-20 steps (not more, not fewer)**
- **Can include “project portfolio” management**

Example



Should (indirectly) address...

- **Do we have a PDCA cycle?**
- **Who is involved? Where?**
- **How long does it take? (Or, how many cycles to “get the whole thing done”? Or is that a meaningful concept?)**
- **It can be similar to a Value Stream Map, but is it then a PDCA cycle?**

The BV Model

- **Improve the BV model and show (more) its underlying theories and assumptions**
- **$f(x,y,z) = \$\text{Total}$ (typical example)**
- **What is/are the function(s)? (addition, multiplication, etc, etc)**
- **How many variables? 3, 5, 8?**
- **What are the best assumptions about “best” values for the variables?**
- **What is the TOTAL BV that your model expects?**

Timeboxes/Feedback loops

- **Describe the timeboxes used in your BV Engineering**
- **Describe the feedback loops, and where new learning is used to get better.**

Hints 1

- **Think outside the Scrum “box” (or the box you have put Scrum in)**
- **Just describe, don’t fix.**
- **PO rules; just enough info for the PO to understand.**
- **Both PO and SM have a role in keeping the team from getting stuck.**
- **Do “the best we can” in this timebox.**
- **If you don’t know, guess for now. (And check later.)**

Hints 2

- **“Could we do a VS map?” Yes, and where is the PDCA cycle?**
- **It can include only one (Scrum) team or multiple teams.**
- **For here, KISS is probably a good idea. But you can use this basic framework as a start for complex situations. “Things should be as simple as possible, but not simpler.”**
- **For here: one product is enough.**
- **You won’t be able to keep yourself from fixing, just not too much.**

Hints 3

- **Do something in all 4 areas:**

- Map




- BV Model

- Theories

- Timeboxes/feedback loops

- **Don't get stuck too much in one area.**

Debrief




-  **In a sentence, not repeating what someone else said, what was the (next) biggest thing you learned?**
-  **You can “show” results.**
-  **Biggest = most useful ??**

The End

For now....

Retrospective

Take 3 minutes with your team and answer these:

-  **What do you remember?**
-  **What will you act on 'tomorrow'?**
-  **What thing(s) will you do to improve your BV Engineering?**

Contact Info

Joseph Little

Kitty Hawk Consulting, Inc.

LeanAgileTraining.com

jhlittle@kittyhawkconsulting.com

704-376-8881