

# **An Introduction to the Business Value Engineering framework.**

We believe Business Value Engineering is the key area for improvement in new product development in the firms we talk to.

## **Short Definition**

Business Value Engineering is the values, principles and practices we use to continuously improve the delivery of business value.

Or, put another way, Business Value Engineering is a framework for helping us to continuously deliver more and more business value to our customers, for the workers, and for the shareholders.

## **Introductory Ideas**

Business Value Engineering (BVE) is an approach towards business value that has a lot of similarities to Scrum and to Lean. And in fact, borrows ideas from those two sets of ideas.

So, like Scrum, it is purposefully a simple framework, and one is expected to take the framework and add more to it for specific usage in a specific situation. BVE sets up a basic framework, but the people involved must add the richer detail.

Like Lean, BVE recommends using simple tools quickly to gain some view on the problem, and then to incrementally make improvements in the process. So, similar to Value Stream mapping, we recommend mapping out one's specific BVE process in the specific situation in some detail, but relatively quickly. Our assumption is that in one or two hour's worth of mapping, serious weaknesses can be readily identified by the team. And then incremental improvement can start.

## **Initial Practices**

We recommend that a given firm form a small team to do BVE. Much like a Scrum team, although the BVE team may work together only a few hours, days, or weeks (rather than the months and years we like to see a Scrum team stay together).

We recommend that the team first define what Business Value means in the specific context they are working in. Typically Business Value is defined from the point of view of the customers, the shareholders and the workers. Often these can be somewhat contradictory or even very contradictory viewpoints.

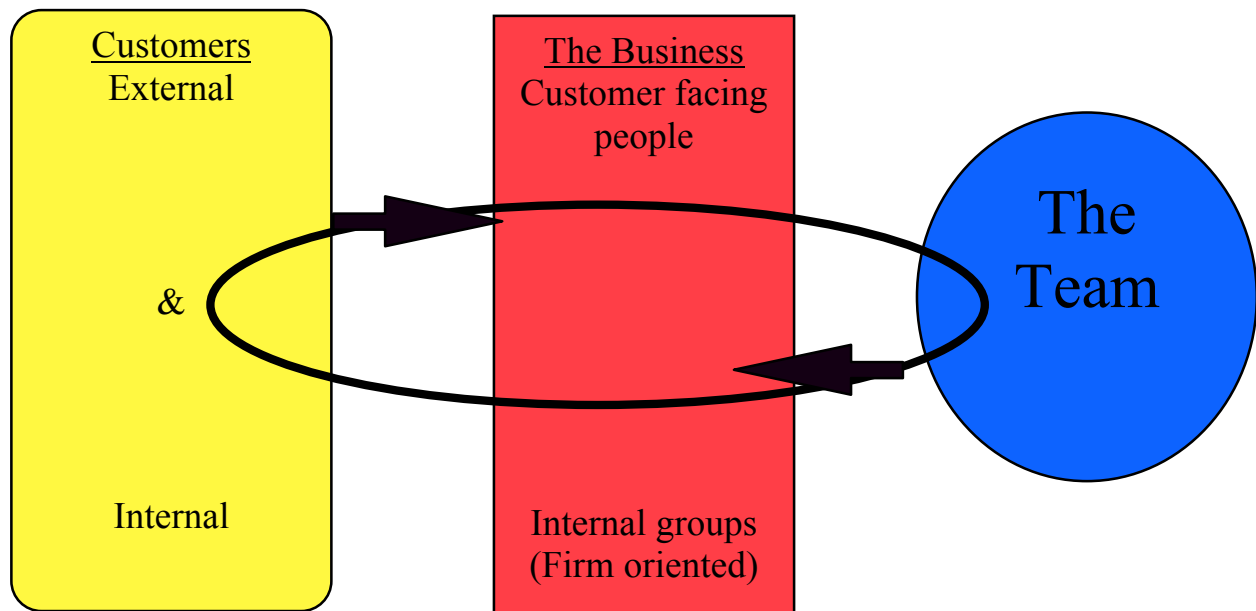
The discussion of the definition of Business Value might include:

- \* Return on Investment
- \* Net Present Value
- \* Risk
- \* Customer satisfaction
- \* Increased revenues
- \* Decreased costs
- \* Various proxies for some of the above things, such as Six Sigma, Lean or other metrics.

In a typical project, we find there are two or more somewhat competing definitions of business value. For example, some of the work may increase customer satisfaction (but there may be no direct monetary effect), some work may reduce risk, and some work may lead directly to increased revenues or reduced costs.

In this same discussion, we recommend that the team start talking about how business value, once delivered, would be measured. At least the basic ideas about that. Later, we ask the team to define a more specific and actionable business value model that can be measured.

The second major practice is business value mapping. This might be called several practices. It starts with mapping out the 15 to 30 steps in the business value engineering process.



Before doing this, we typically have to discuss some basic ideas about the process. Most people start with the notion of ‘gathering the requirements’ and giving those requirements to the Team (the implementers). And then the product (say, a software product) comes out of the Team and is delivered to the customers.

In BVE, we assume that the communication of requirements is always problematic, and that the communication can always be improved. Further, we assume that BVE includes more than just the communication of requirements and more than just a simple 'delivery' of the product. And that these high-level processes are based on multiple theories. And that these theories and their specific implementations can be continuously improved.

Given this situation (or at least these assumptions), BVE recommends that the process be looked upon as a Plan-Do-Check-Act cycle (aka a Deming cycle). We should be trying to prove (or disprove) some of the weaker theories we have used to create the current process. So, we Plan an experiment, we Do the experiment (eg, produce the software), we Check the results (ie, measure whether the business value of the delivered product approximated the expected business value, by some appropriate measures), and Act upon the results (eg, re-do the experiment if it was flawed, or try an alternate theory/practice that might lead to somewhat more success).

Now, as soon as one looks at the process of BVE as a P-D-C-A, where learning and continuous improvement are very important, then obviously one wants to do a series of experiments rapidly. Which means, we start to prioritize getting to more rapid releases of the product (eg, the software).

As might be guessed, we ask the BVE team to map out the process. And we also ask them to identify the theories around BVE that underlie the steps in the process currently being used. And we ask them to more specifically identify the business value model (BVM). The BVM (and there can be multiple BV models) represents the 'theory' of how to measure the BV being delivered. It is usually organized as  $f[v(1), v(2), v(3), \dots, v(n)]$ . Which is to say that the BV would be (or should be) measured by a function of variables 1 to n. (The function might colloquially be said to include multiple functions, such as addition, subtraction, multiplication, division, etc, etc.)

The BVM may be financial or it may be based on some other type or types of metrics. As suggested earlier, a given project might have multiple BVMs (eg, since the apples of dollars are not comparable to the pears of some Customer Satisfaction metrics).

Clearly, the BVM is necessary to conducting good PDCA experiments.

So, let's repeat: The second practice includes mapping, along with the underlying theories and the BVM.

The third practice we summarize for now in two words: making improvements.

## Hallmarks of good Business Value Engineering

The following are, in our opinion, hallmarks of good business value engineering, and thus are key assumptions in BVE.

1. The process is visible and articulated & always improving
2. Failures in BV communication are assumed. The job is to identify the biggest one quickly, then identify which one to fix and, fix it quickly. And repeat this cycle frequently.
3. There are many theories, and we make a concerted attempt to prove out the weakest theory
4. There is appropriate dynamism and change
5. Business & Technology are partners
6. Success is forecast (modeled) and measured after the fact
7. Human judgment is involved (it's not just the numbers)

Each of these is explained briefly below.

*Visible Process:* As we said earlier, we want the team to make the 15-30 steps visible to all. All being mainly the team itself, but possibly to others who could give useful feedback.

We choose 15-30 steps to be the right level of analysis; not too high and not too detailed.

And the purpose of the mapping is to continuously improve the process.

We purposely do not define the frequency of improvements. Scrum defines the Sprint length as 1-4 weeks (with Ken Schwaber strongly advocating 4 weeks). This frequency is fine at one level, although perhaps is too frequent at some levels, and too infrequent at other levels or in other situations.

Clearly, any PDCA cycle implies a high bias toward a greater frequency than 'now'.

*Failures in BV Communication:* First, BV communication includes many many things, not just communication of the so-called requirements. But if we only take the requirements, there are many laws of software development that teach us that communication about the requirements, even between two people, is always expected to fail to some degree.

So, our key assumption is that BV communication will always be failing to some degree. Not completely, but still importantly incorrect in at least a partial way. So, for example,

unless the 'customer' is more delighted than he or she could possibly have imagined, then our understanding of the requirements is partially failing.

BVE also assumes a theory that I will express this way: Bad news does not get better with age. All of the software industry's data about how the cost of fixing a defect rises very quickly as the fix time moves away from the time the code was initially written. And in BVE we assume this idea more broadly, about everything in BVE communication. Now, this assumption may not be wholly correct in every case, but we assume it is correct so often that the times it is incorrect are relatively inconsequential.

As a too-quick aside to a question that might arise in a reader's mind: We do not always assume that fixing the defect should precede working on any new user feature. But we are biased, other things being equal, to act that way.

As a practical consequence, if a team is using Scrum, one wants the Product Owner (PO) to work closely and at length with the Scrum Team daily to find and correct these failures in BVE communication. The amount of time might be limited to some degree, eg, if the PO has yet more valuable things to do.

*Theories:* The underlying theories are made explicit. And the weakest theories are examined. And the weakest theory is selected for testing in the PDCA cycle or otherwise.

Our bias is to believe there are many many unexamined theories underlying what any firm and indeed any project team is doing in the area of BVE. Because these theories have lain unexamined typically for years, it is difficult for anyone to try to make these explicit. It is much like asking a fish to explain water; the fish just 'assumes' water, which, since it is ever-present, needs no explanation. So, we must have some sympathy for the effort to make these theories, often very lame in my opinion, more explicit. Some sympathy, but not too much.

*Dynamism:* In this idea, we simply mean that at certain time or in certain situations, we need more change or dynamism in the way we work on BVE. And sometimes perhaps less. For example, if the online advertising market has been continuously growing over several years and then the economy hits a serious recession, some firms that have depended on online advertising might wish to examine or improve their BVE more aggressively for a few quarters, until they feel assured that their approach is now reasonably appropriate for the new economic situation.

*Business and Technology Partnership:* Lean, Agile and Scrum and other business thinkers have said this, so this is not a new idea at all.

We do think the partnership is yet more essential in the context of BV Engineering. For example, we assume that some of the key problems that must be solved will be based

on the inter-linkages between business and technology issues. In a simple model, business represents the benefit and technology the cost, so to make good cost-benefit trade-offs, the team must be in a partnership.

*Measurements:* Some of our colleagues seem to think that measurement is everything. And some, that it is useless. We propose that measurement is key, but by no means the only thing in BVE.

To do a good PDCA, one must have not only an hypothesis, but also a way to measure whether the hypothesis is being proven or disproven in the experiment. So, following a decent scientific-like approach, BVE says that the expected results must be modeled and identified up-front. And then the actual results must be measured.

The idea of using this basic technique of science has been around for hundreds of years. And many other disciplines in business (not just the Deming cycle) call for it.

Still, we feel it must be given emphasis, since we find so few people doing it, and even fewer doing it well, in the context of new product development.

One of the complaints is the assertion that measuring BV delivered is hard. And, in the typical cases that we see, this assertion seems to be correct. It is hard. But in BVE we assume that the value of continuous improvement is so great, that it offsets the difficulty (cost) of measuring.

*Human judgment:* When I talk to others who are interested in Business Value (eg, in the context of new product development), I often hear: “Why measure business value? It is not fair to the Scrum Team who has no control over BV delivered. And anyway, the results are always about the past. We should be forward-looking.”

To the first issue: BVE is not just about the Scrum team. But just for the Scrum team: they must care to know whether they delivered any Business Value. Or how much. This is why we work. This is largely what work is about.

To the second complaint: Yes, measured metrics are always of the past. But this does not mean that the past cannot be a good predictor of the future. Anyway, it is something we have to guide us, when we have few other things available. By no means a perfect guide, but something. If we apply good human judgment, we can reach reasonable conclusions about how to use the information at each instance.

## **Uses of BV Engineering**

The BVE framework is clearly meant to be used to continuously improve or increase the delivery of BV to customers, shareholder and workers.

This is the biggest, most important usage. And we have already discussed it above.

Another usage is to enable the people with specific situations to judge where to get help.

BVE assumes that the BVE team will map the whole end-to-end process. And that the BVE team will make a concerted attempt to optimize the whole (rather than to improve one piece, possibly leading to a reduction in overall results).

Now, typically there are a plethora of competing ideas about how to improve BV Engineering. As example, we will list a few:

- Focus on the BV Model (the variables, the functions, and the constants used for calculation of a specific final value)
- Software By Numbers (book) (eg, the 'minimum marketable feature set' idea)
- The Kano Model
- Requirements elicitation
- The Agile Specification (Jeff Sutherland's idea of just-enough, just-in-time documentation, just as needed by the Scrum implementors)
- Project portfolio management
- Motivating the team by describing the Business Value of each item
- Using a partially done system to enable the customers to see what they really want
- Using a partially done system to help identify where the cash really is in this product area
- Using a kanban board to manage to completion on a small number of active business initiatives
- Working closely with customers to help them identify the business value to themselves of a specific feature set
- Changing the delivery process so that you can deliver into production at the end of each Sprint
- Focusing on fast "process cycle time", from the time the customer identifies a new idea until it is in production
- The use of Priority Poker and BV Points based on the Fibonacci scale

These are just a few of the competing ideas.

The bias in BVE is much like that in Theory of Constraints. Identify the single biggest thing to fix. Then fix it. Then re-identify the now-current biggest thing to fix.

So, BVE enables each team in each specific situation to identify which of these good ideas might give the most benefit now.

## **The Pareto Principle**

We all know this principle as the 80-20 rule, identified first by Vilfredo Pareto, in his studies of wealth distribution. It has since, by him and others, been found to be a key

concept. In BVE, our bias is to assume that 80% of the BV is in 20% of the work (or features, if all of roughly equal size of 'work'). Rather than a 1-1 (or 50-50) ratio, where one unit of work virtually always one unit of BV.

As Pareto identified, the Pareto Principle can be applied at every level of the population. Meaning, in Scrum and Agile Estimating and Planning, as we break down epics into small sashimi slices, usually over several cycles, we can re-apply the 80-20 rule each time to good effect (ie, more value delivered for less work).

As a practical matter, this implies a recommendation that the Scrum Product Owner spend lots of time trying to see the 80-20 rule playing out within the Product Backlog. The value of doing this better and better is so great, that it justifies the Product Owner spending a lot of time doing it. Over the whole course of the effort (not just once, up-front).

### **A few principles of BVE**

1. *Minimize the number of prescriptive elements.* This is similar to Scrum. And one reason is that we assume there are so many different kinds of situations. A second reason is that we want everyone, wherever they are, to start using BVE to get better. If we add more prescriptive elements, then many people will say 'I would like to start BVE, but I have to do a waterfall project to fix things around here, and then, after that is done in 9 months, then we can start BVE.'
2. *It's about the Team and how the Team can create knowledge.* Some people want to have one person discover the best approach to BV Engineering. It is our bias that this is, despite some appearances of simplicity, actually a very complex subject. And thus, a subject that needs all the insights and knowledge creation in your specific situation that a Team can bring.

### **Elements We Can Use**

As some suggestions for things that might be improved, and to give some readers a greater sense of the scope of BV Engineering, here are some BVE elements that could be deployed.

Again, we feel compelled to remind you of some caveats:

- a. some of these tools might be harmful in your specific situation, and
  - b. focus on the one or two most important changes first....do not try to add all these tools at one time into your specific situation.
- Product Owner Team (ie, having a Chief Product Owner, who works with team Product Owners as a PO Team)



- Product Backlog. From Scrum.
- Product Backlog prioritized by BV
- Priority Poker (to identify the business value of each Product Backlog item)
- Story Points (proxy for cost, for cost-benefit analysis)
- Minimum Marketable Feature Set
- Reprioritize before each Sprint
- Increase velocity (remove impediments)
- Making the stories smaller
- Value Stream mapping
- Kano Analysis
- Voice of the Customer (typically associated with Six Sigma)
- Having the team live with the customers
- Pareto chart (eg, of causes of customer problems)
- Process charts or high level use cases
- Other Lean, Six Sigma, or TQM tools
- Understanding the importance of minimizing technical debt
- Agile portfolio management
- What quality means to the customer and why it is 'free'
- Just-in-time knowledge creation
- Modifying the BV model frequently (& the constants used in the model for projections)
- Removing impediments
- Comparing our BV Engineering to theirs
  - We're different; what does it mean?
- Evaluating the actuals (metrics)
- Identifying better sources for good user stories (eg, observation, "living with", experts, user interaction, "prototypes", etc)
- Identifying good user stories
- Fleshing out good user stories with an Agile Specification
- Improving the monetization of User Stories (or themes)
- Improving the conversations around the user stories
- Getting better feedback faster

In addition to the 'tools' above, the following metrics could be used in Business Value Engineering. This is not to suggest that we personally think all the following metrics are of equal value.

1. NPV (net present value)
2. ROI (return on investment)
3. Faster end-to-end cycle time (aka Process Cycle Time)
4. Increased sales (units)
5. Increased market share
6. More eyeballs (on a webpage)
7. Improved eyeball demographics
8. Reduced costs
9. Reduced risk (although I prefer if this is made more concrete by being monetized...see underwriting)
10. Net promoter score
11. Increased wallet share
12. Higher quality
13. Any metric that shows higher customer satisfaction

## **Conclusion**

As could be guessed, we think BV Engineering is an important topic. We believe most of the companies we work with can improve their BV Engineering tremendously. And that BVE is the one single area (albeit big) which holds the most promise for significant improvement.

We hope this discussion helps your team get better.