



BUSINESS VALUE ENGINEERING & AGILE SPECIFICATIONS

ScrumU, December 4, 2009

Definitions

- **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.**
- **BVE: A learning and incremental improvement approach to giving customers more of what they really want, looking at the whole process, end-to-end.**

Definition

- **Agile Specification: “Just enough” documentation developed for the implementors just in time. Not too much, not too little; just enough, in their opinion (and as results prove).**
 - Typically tied to one or a few User Stories.

Attributions

- Some people who directly or indirectly contributed: Ken Schwaber, Jeff Sutherland, Kent Beck, Peter Drucker, Takeuchi & Nonaka, Jim York, Chris Mats, Kent McDonald, Womack & Jones, Mary & Tom Poppendieck, Taiichi Ohno, 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

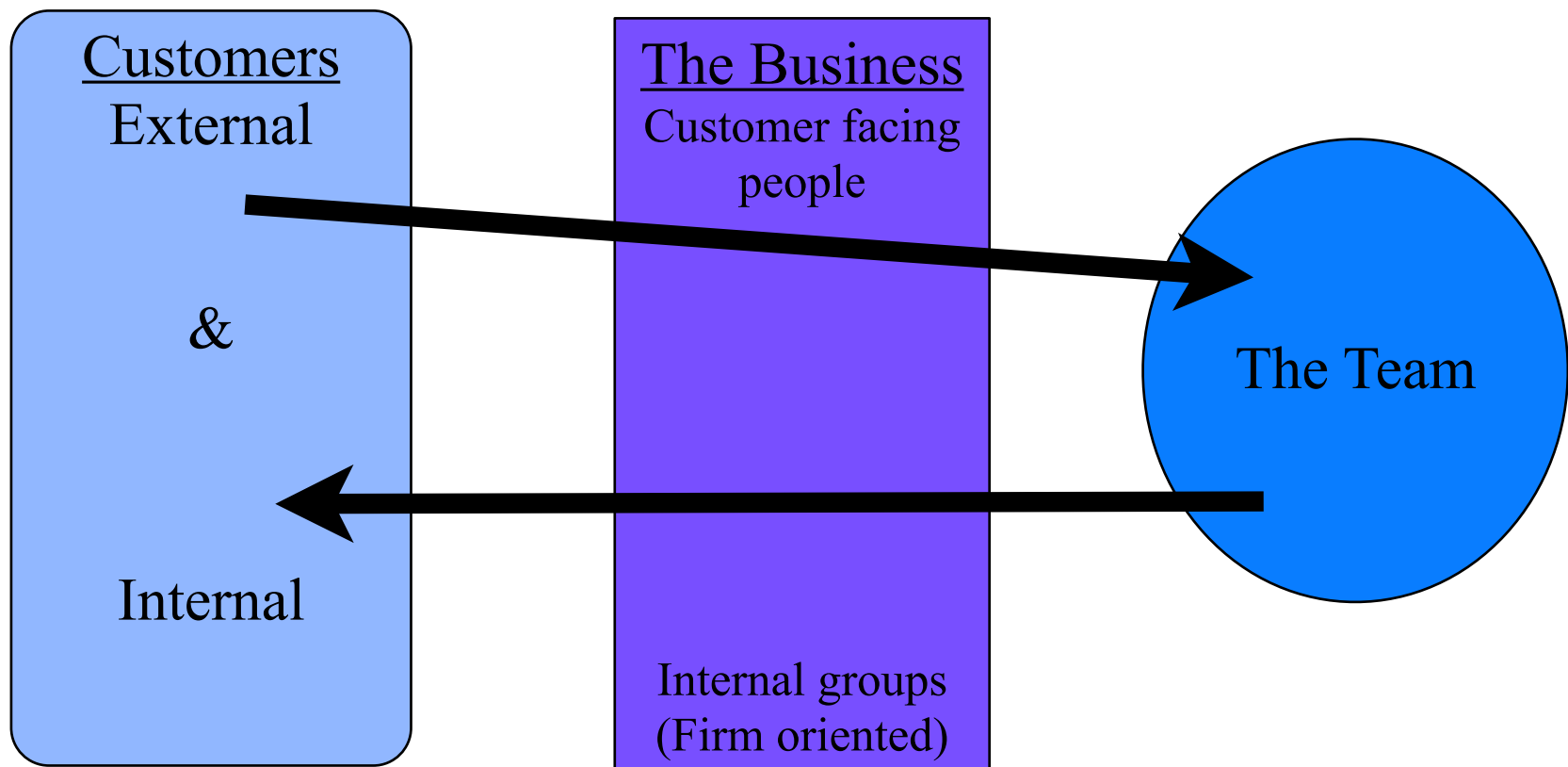
In other words: In my opinion, BV Engineering is the most important thing to work on....

Some prerequisites

- **You agree that...**
 - Our business is JIT knowledge creation
 - Our business is JIT knowledge delivery
 - Optimizing the Pareto Rule is key to success
 - Customers don't really know what they want
 - "I know it when I see it."
 - What they really to act on is a complex set of trade-offs, including benefit, cost and time
 - Tacit knowledge is more important than Explicit knowledge

What the Product Owner does

BV Engineering



First problem

- **Customers want infinite features since effective cost to Dept is zero.**
 - This is not really how things “are”, but how most university departments are treated.
 - (Universities really do not have infinite resources.)
- **So, what to do?**

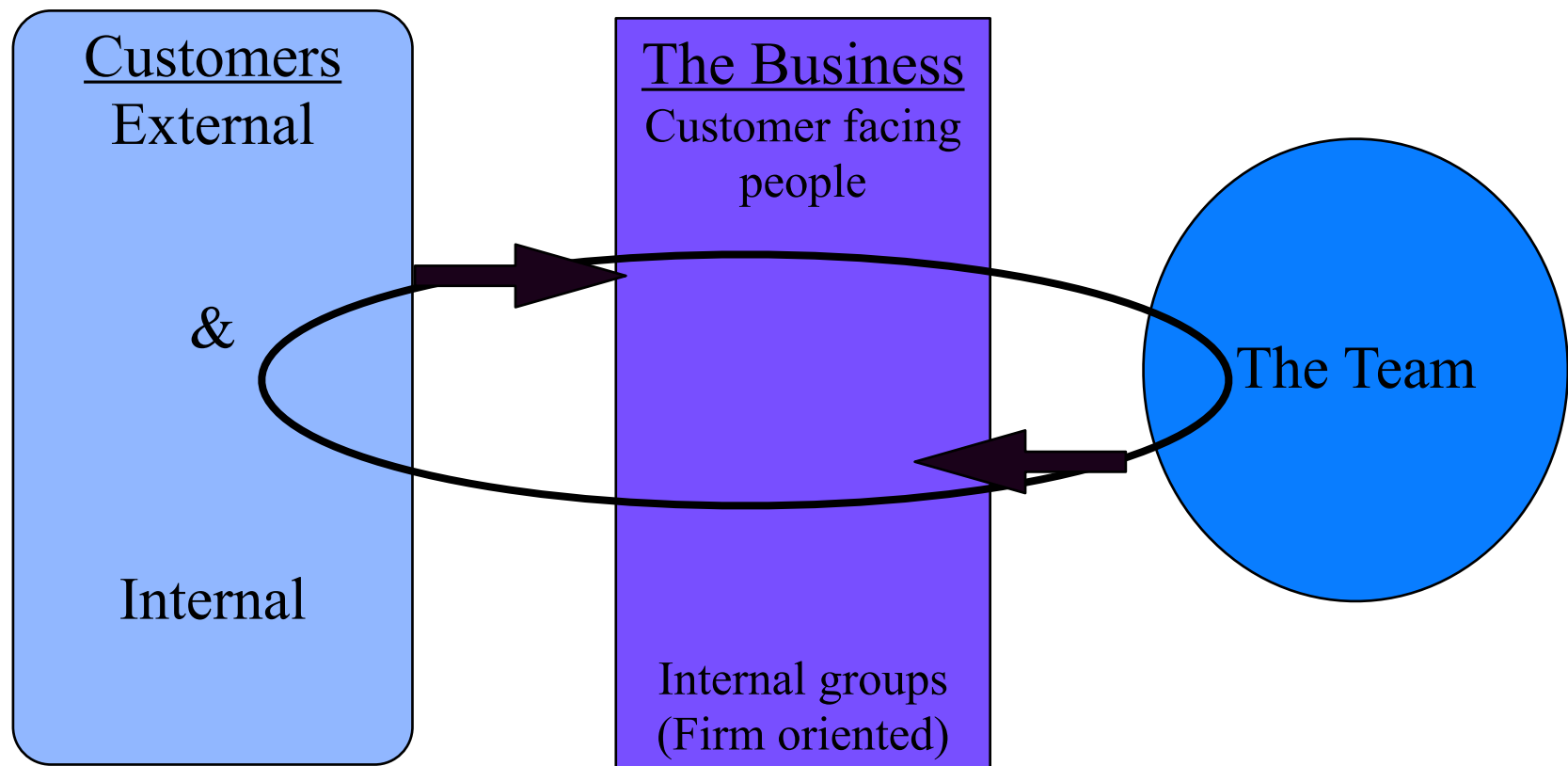
“Ideal” solution

- **We do cost benefit analysis in \$, and any effort that gives less than 3X return does not get done. And efforts are ranked by their ratio.**
- **And any story that gets less than about 3X return does not get done.**
- **And, the senior manager gets real \$ benefits and real \$ costs, so she has an incentive to deploy early.**
- **BUT....**

Some solutions

- **Make each Mgr estimate the benefits and have “the delphic 5” review the estimate vs other projects.**
- **Fix the time period. (“You can get any stories you want in 5 Sprints.”)**
- **The delphic 5 might do two rounds of guessing at the \$ benefits of a project. And then avg.**
- **“Assume” that a project will deliver \$ benefits that are 3X the costs. Give these projects to the delphic 5 for a “sniff test”. Probably for a few, they will say: “You must be kidding!”**

Is it better this way?



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)**

Agile Specifications

- **So, one of the key things about BV Engineering is:**
 - How do we get the “requirements” into the Team, so they can do them the best?
- **Agile Specifications are a partial answer**

More assumptions

- **We're still doing lots of other stuff**
- **We're still using User Stories and Acceptance Criteria**
- **The PO or another business person is having daily conversations with the Implementors (pigs).**
- **We developed "all" the stories as quickly as possible.**
- **Arch & Design were initially done using the User Stories (+ conversations, etc); and is continually being improved.**
- **We all actually believe in JIT knowledge creation**
- **Etc.....**

Definition

- **Agile Specification: “Just enough” documentation developed for the implementors just in time. Not too much, not too little; just enough, in their opinion (and as results prove).**
 - Typically tied to one or a few User Stories.

How do they work?

- **One or two Sprints before a story goes into a Sprint, the PO arranges for the Agile Spec to be built.**
 - Maybe by the PO, the BA, a stakeholder
- **The Team and the PO agree what will be in the Agile Spec.**
 - Maybe: Wire Frames, drawings, simple use case, data elements, key edit criteria, diagrams, pictures, more robust test examples, etc, etc.
- **The Agile Spec enables conversation, it does not replace conversation.**
- **Based on experiences, the required content of the Agile Spec is continually being revised.**

How again?

- **How does the PO, BA, Stakeholder get the content for the Agile Spec?**
 - "Agile Spec" does not define that.
 - There are many, many techniques.
 - One best practice is to watch the real users carefully; interviewing user is by comparison fairly low value.
- **Does the Pareto Rule apply to the content of the Agile Spec?**
 - YES!!!

How? (3)

- **When exactly is an A.S. built?**
 - Not defined. Probably 1 or 2 Sprints before.
- **Do the Implementors review the A.S. before the SPM? Yes!!**
- **Can the A.S. be tied to a specific User Story? Yes.**
- **What is the purpose? Probably many. One: To enable the implementors to get the PBI done as fast as possible (assuming also high quality).**
- **How do we know the content of the A.S. is right? The implementors keep giving the PO feedback. Continually improved.**

Why?

● Why don't you define the A.S. in more detail?

- Because what Team A needs is different than what Team B needs.
- Because: The following might be different: The product, the customers, the project, the environment, the PO (BA, stakeholder), the Team (memory, basic understandings, etc), etc, etc.
- We don't build documentation around well-known knowledge
- We might need to document some knowledge that is not well known.

 **Documentation is ONLY there to support communication and understanding.**

The End

For now....

Retrospective

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