

Business Value Driven Software Development

Chris Matts

ThoughtWorks

9th Floor Berkshire House

168-173 High Holborn

London, WC1V 7AA

cjmatts@thoughtworks.com

Andy Pols

Pols Consulting Limited

5 The Granary House,

Hope Wharf, London, UK

andy@pols.co.uk

1 ABSTRACT

In this article, we introduce the concept of *business value*, describe the things you need to do to make delivering business value a reality and present some small steps that you can apply to your organisation today.

2 INTRODUCTION

We start by asking, “*What is the objective of the IT Department?*” The answer is quite simple. The IT department exists for one purpose alone – the same purpose as any other department – to help the company make a profit [REI97]. We would argue that it has no important objective that cannot be quantified in economic terms. If we fail to generate sufficient profit, then the business will spend their money on something else.

Most IT projects traditionally focus on delivering working software, features, functionality or a service to the business. We believe that IT departments should focus purely on delivering frequent business value.

Focusing on business value provides a number of key benefits:

- The IT department’s goals align with those of the business.
- The business’ perception of the IT department changes from service provider to partner.
- Builds a trusted relationship between the IT department and the business.
- Describes how your IT department can

satisfy the business’ fundamental requirement, which is to rapidly respond to change.

3 WHAT IS BUSINESS VALUE?

Quite simply, a project creates business value when it increases or protects profit, cash flow or return on investment in alignment with the company’s strategy.

3.1 *It’s a model, not a number*

Business value should be presented as models rather than statements. This allows the business value to be challenged, and allows it to be re-evaluated as conditions change or further information is discovered.

It is almost impossible for business management to make a sensible decision when presented with a statement such as:

“This project will generate an additional \$15m in profit”.

It is much easier for management to assess a model where the developer of the model states the assumptions and inputs. For example:

“This project will generate an additional \$15m in profit”.

The model is based upon the following assumptions:

We achieve 20% of the sales of existing product XYZ (\$100m / year).

The total cost of designing and producing the product is \$5m

Our product is first to market.

We are able to release the product two months before Christmas.

In effect, the project builds a financial statement with explicit assumptions about market conditions, revenues, costs and risks.

It's important to remember that the model is not the goal but a tool to achieve business value. The model should be barely sufficient [COC00] – with just enough information to enable senior management to decide whether to continue; anything more is a waste of time and energy.

3.2 Case Study – The MODEL

Business Value is driving one of the projects we are working on. It is a Credit Risk project to manage the exposure of the company to the possibility that a counterpart goes bankrupt. The original business case was that the introduction of the system would reduce losses during a default by 20%. It then looked at the defaults for the previous year and projected these forward to come up with an annual value of the system.

This was challenged on two fronts. First, the number and size of defaults for the previous year had been exceptional. Secondly, no one knew how the system would reduce the losses by 20%.

With the business value model, we took a completely different approach. It turns out that the credit risk department did not trust their existing system. As a result the credit limits are kept lower than they would like. This is restricting the growth of the business. The new system should be more accurate and allow the credit limits to be increased. A 10% growth in business is predicted as a result. This translates to a \$X million annual increase in profit. Interestingly, the increase in limits will also give a \$Y million annual increase in losses as the company will be taking more risks. This is acceptable as the resulting business value shows a net profit of \$X-Y million.

4 COMMON PITFALLS

Defining business value is hard. It is often difficult to see what is wrong with a business value statement. It may feel right, but without proper understanding, it can destroy business value rather than create it.

Let's look at some examples:

“Project X will generate revenue of \$1M.”

The statement is expressed as an absolute rather than a model. The biggest problem with this statement is that it is not possible to re-evaluate the business value if market conditions change.

Further, there is no mention of the cost of generating this revenue, or the amount of capital investment required to generate the revenue. This investment may deliver less than the required rate of return for the organisation.

“Project X will generate savings of \$1M.”

As with the above statement, conditions can change and the savings are either not available or another solution emerges. Projects based on savings need to be carefully monitored as any increase in costs eat directly into the savings. In addition, you need to know what will generate the savings, so as to be sure that the project correctly implements those things!

“Project X will automate the current manual process”

This focuses on a specific solution without detailing the benefits or the cost of implementing the change. Automation for its own sake is not always a good idea.

The emphasis should be on business value, which may result in automation rather than automation for its own sake.

“Project X will deliver World Class operations and systems”

Once again, there is no reference to business value. World-class operations for their own sake

may destroy value. A better statement would consider unit costs, customer service or ability to adapt to exploit opportunity, all quantified to identify the business value they will generate.

“The regulator will shut us down if we do not do this!”

This statement may be true. However, what is the loss of profit and how much will the project cost? It could well be that a much more minimal approach is appropriate. In response to new regulations, some organisations create multi-million pound solutions whereas others create a manual process involving a spreadsheet.

“Project X allows us to achieve straight-through processing”

This is a solution, not a statement about business value. Once again, the business value should be identified and then the appropriate solution adopted. Implementing straight through processing may cost more than its business value.

“Project X provides a real-time Profit and Loss statement”

This is a description of a feature. How does the real time Profit and Loss statement generate business value? Would end of day suffice? In reality, it may be a *nice-to-have* for an important person in the organisation, in effect; an Ego-App. Ego-Apps are prevalent in many organisations. If the person is a senior executive, a project may start to deliver their requirement without fully questioning their reasoning.

5 DELIVER BUSINESS VALUE NOT FEATURES

We now have a business value model that can be validated against the business strategy. We now provide some guidelines on how to deliver the business value to the customer.

5.1 Break it down!

A requirement ought to contribute to the overall project business value. Business value should therefore be broken down and attributed to individual requirements¹, or sets of requirements².

We have found that many projects slice the requirements in such a way they actually increase the operational costs rather than increase business value.

The requirements effort needs to be collaboration between the business sponsors and the IT team, rather than one group in isolation. The IT team determines how the project can be broken down and the business allocates the value to the separate pieces.

Once team members see the value of what they are doing, they are better placed to propose pragmatic alternatives.

5.2 Case Study – Break it Down

For the credit system project, we broke down the high-level requirements to show the business value for each component:

1. Calculate Exposures – 25%
2. Collateral – 25%
3. Manage Limits – 20%
4. Synergy – 30%

The collateral piece was further broken down as follows:

1. Letters of Credit – 75%
2. Four other forms of Collateral – 25%

This is despite the fact that Letters of Credit only forms 20% of the requirements for Collateral.

Therefore the business value for Letters of Credit is $\$(X-Y) \text{ million} * .75 * .25$. This value was logged against the letter of credit story card.

¹ For example use cases [COC01] or XP’s user stories [BEC99]

² For example Minimum Marketable Feature (MMF) sets [DEN04]

5.3 *Phase delivery based upon business value and cost, not functionality.*

Traditionally, projects are scheduled by functionality. Projects should be phased to deliver based on business value and cost.

The business and IT need to come together to understand how this can be achieved. Neither group can do the phasing for projects in isolation.

5.4 *Customer Pull, not IT Push.*

IT projects should be driven by business need. The business should be able to pull business value from the IT department at times convenient for the business. IT project schedules should not push business value onto the business at time convenient for itself.

A project should deliver Business Value to business at least every 3 months. If not, there is a real risk that IT will develop software that either is not needed or fails to provide business value.

Where the end users of a system resist change and refuse to accept change every 3 months, senior IT management should halt the project until senior business management addresses their change management issues.

5.5 *When to Stop!*

Business value tells you when to stop. When the cost of implementing subsequent requirements is greater than their business value, the project should stop until either more business value is found or the costs are reduced.

5.6 *Test, Test, Test!*

Any systems developed should have business acceptance tests, which are specified in advance of the development. The intension is to expose the provision of the business value so we know when it has been achieved. Where appropriate, the tests should be automated. These are used in both user acceptance testing and post delivery testing.

Post delivery testing is critical for seeing how well the system actually delivered the planned business value. By continually monitoring the actual business value delivered, the Business is better placed to make sensible strategic decisions about future releases of the system.

6 TRUST

Building trust between the business and IT takes time. Trust is built gradually each time the IT department demonstrates delivery of business value.

One of the many benefits of greater trust between the business and IT is that the business will require less compliance documentation. Documentation takes a significant amount of effort and causes delays, both of which destroy business value. Any reduction is a welcome improvement.

7 WHAT DO I DO NOW?

Follow the process we have outlined in this article:

1. The business picks one of the worst problem projects.
2. The business creates a business value model for the project (if they don't have one already)
3. Assess whether the project is achieving its goal.
4. If it is worth proceeding with the project, start grouping the requirements to see which requirements deliver the largest business value.
5. Deliver the first big-win requirements identified in the previous step (no more than 3 months of work).
6. Look for the next big win and repeat.

The big wins come from delivering business value in regular increments.

8 ACKNOWLEDGEMENTS

We must thank everyone who reviewed this article. Special thanks to Mary Poppendiek for her thoughtful review comments and kind words

of encouragement; Tracy Bialik, Russ Rufer, and Ken Scott-Hlebek of the Silicon Valley Patterns Groups for their scarily detailed feedback; Peter Brown for having to read an early version several times; XTC in London for great conversation and stimulating ideas; and finally the Writers TWIG at Thoughtworks for comments on early drafts.

9 REFERENCES

- [BEC99] *Extreme Programming Explained: Embrace Change*, Kent Beck, Addison-Wesley; 1997.
- [COC00] *Balancing Lightness with Sufficiency*, Alistair Cockburn, 2000, <http://alistair.cockburn.us/>
- [COC01] *Writing Effective Use Cases*, Alistair Cockburn; Addison-Wesley.
- [DEN04] *Software by Numbers*, Mark Denne, Jane Cleland-Huang; Prentice Hall 2004
- [GOL96] *Theory of Constraints*, Eliyahu M. Goldratt, Gower Publishing Limited; 1994
- [REI97] *Managing the Design Factory: The Product Developer's Toolkit*, Donald G. Reinertsen, Free Press; 1997.