

Diving for Hidden Treasures

Uncovering the
Cost of Delay in Your
Project Portfolio



Johanna Rothman
Jutta Eckstein

Diving For Hidden Treasures

Uncovering the Cost of Delay in Your Project Portfolio

Johanna Rothman and Jutta Eckstein

This book is for sale at

<http://leanpub.com/divingforhiddentreasures>

This version was published on 2016-03-28



Practical ink

No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage and retrieval system, without written permission from the author.

Every precaution was taken in the preparation of this book. However, the author and publisher assumes no responsibility for errors or omissions, or for damages that may result from the use of information contained in this book.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and Practical Ink was aware of a trademark claim, the designations have been printed in initial capital letters or in all capitals.

© 2016 Johanna Rothman and Jutta Eckstein

Contents

Why do we care about the value of the project portfolio?	1
Planning and Controlling Complex Projects	3
Prediction is not possible	3
Beyond Budgeting to the Rescue	5
Applying Beyond Budgeting	7
Conclusion	11
References	11
A Focus on Continuous Value	13

Why do we care about the value of the project portfolio?

We play organizational games with estimates, budgets and goals. And, we use them to make serious decisions in organizations. Value is how we reconcile these problems. If we make decisions by value, we change how we rank the backlog for the iteration, we change how we organize the project portfolio, and we change how we fund the projects for the organization.

How do we find the value? What feeds into the value? Cost of delay is part of it. Creation of market share is part of it. Waste is part of it. Investment is part of it. You will need to determine value for your organization by looking at all of these things and discussing them.

However, if we try to create a goal with an estimate, either the goal is wrong or the estimate is wrong. People estimate so they know it will fit into the iterations, instead of looking at what they need or they estimate a project so it fits the overall portfolio. People are forced to estimate to fit the date. We see this at the iteration, project, and project portfolio levels.

How do we value different projects and programs in the project portfolio? One common way is to use estimation. Especially for large programs, estimation is costly and wrong. The estimate becomes the target, the plan. The estimate is a trap for the eventual project or program. The estimate prevents feedback or continuous learning. The complexity of today's projects and project portfolios puts an extra burden on decision making.

Estimation is insufficient for evaluating the project portfolio. We need to use value. Value is more difficult, and adheres to the agile

principles. Learn how to evaluate the project portfolio by adhering to the core agile values.

Planning and Controlling Complex Projects

Jutta Eckstein

I have been working on large and complex agile projects for more than 20 years. Typically planning and budgeting are based on trying to predict how development will turn out. Very often the stories are estimated by the development team, but the budget for the whole project is independent from those estimates. Especially for complex projects this leads most often to (unwanted) surprises. However, learning about Daniel Kahneman's and the Beyond Budgeting folks' work helped me a great deal in better understanding how planning, estimating and budgeting relate and why the traditional approaches don't work. Of course, we all know how this works in the small, how to plan and steer a project by iterations. Yet, how will you decide in favor or against a project, how do you define the budget for starting a large project, and how do you know how your tiny iterations (across many feature teams) fit into the long-term project goal? Please note, I'm not talking about small or regular projects, I'm talking about projects with 50-300 developers, taking for example 3-5 years to finish or comparable large product (line) development.

Prediction is not possible

Daniel Kahneman, winner of the nobel prize in economics although he is a psychologist, came to the conclusion that most of the things that are happening, are based on coincidence: One of the examples he gives refers to the past history — chances stood 50:50 that the embryo who became Adolf Hitler would have been female. Who

knows how this would have changed the past hundred years in the world? This means predicting complex events is not possible.

However, Kahneman collected as well work from colleagues who researched in the same field in order to verify his point. For example Philip Tetlock, also a psychologist from the University of Pennsylvania collected more than 80.000 political and economic predictions. Note that the people who came up with those predictions actually make a living from predicting the future, so we are talking about real experts! Problem was that those predictions were worse than applying just the normal curve of distribution to the prediction. Yet, when proved wrong, hardly anyone acknowledged that his predictions were wrong, almost all of the experts came up with excuses or reasons why they believe they were actually right yet the timing was wrong without providing any reasoning for what kind of timing would have been right.

In another study, Terry Odeon, a finance professor at the University of Berkeley analyzed approximately 10.000 broker results consisting of 163.000 trades. The interesting aspect with brokerage is that for every trade there must be someone who believes that it will be better to sell e.g. the stock and someone else believes buying it would be better. Yet, both traders are typically regarded as experts in the field. In this analysis Odeon found that on average the stocks sold performed 3.2% better than the ones bought.

The last example, I want to mention is one based on Kahneman's own experience. He used to work for the Israel army. There he created a test which he used with his group to evaluate candidates for the officer career. You can imagine probably very well that kind of test - next to interviews, people had to solve some tough problems where they had to build something, get over another thing and this way it became apparent who is taking the lead, who is opposing, who is a good team player and so on. Based on observing the applicants during this test, Kahneman and his team developed always a high confidence for the further qualification of the people.

However, the feedback they received from the commander every few months was that their evaluation was only a tiny bit better than blind guess. Interesting fact is, that neither Kahneman nor his colleagues changed their approach or rather the conclusions from their observations based on this feedback. When seeing an applicant taking the lead it was still obvious to them that this will be the perfect candidate for the officer career. They were just too convinced about their impression that changing the conclusions seem to be impossible for them.

So what can we learn from those studies? There are two really important lessons to be learned:

- First, predictions will always be error-prone, just because the world “or rather any complex event” is not predictable.
- Second, high subjective confidence (as experts often show no matter if their confidence refers to recommending soldiers for an officer career or to estimating a project) is no sign of accuracy, it is only a sign that the expert has a coherent story. High confidence (sometimes as well called intuition) can only be trusted if you are acting in a stable environment.
- And there is actually a third lesson in Kahneman’s research I didn’t bother verifying this with a study because this is the foundation for iterative development: In contradiction to long-term trends, short-term ones can be predicted with fair accuracy as long as they are based on previous behaviors, patterns and achievements.

Beyond Budgeting to the Rescue

Beyond Budgeting (see <http://bbbt.org> and <http://www.bbtt.org/>) is a development driven by CFOs from different companies. They were unhappy with the effect budgeting had on the success of the

respective company. The typical experiences and you might be able to relate to those are two-fold:

- Imagine somebody is asking for a specific budget for e.g. a project and then the project turns out to be cheaper. Being afraid that next time when asking for money only getting half of it, the remaining money will be spend anyway (although not for the original purpose). Thus, this isn't really contributing to the company's success.

And now imagine that someone asks for a specific budget, but then the market changes, which might mean he would need twice the budget. Well, he didn't ask for it ahead of time and therefore it's not possible to act according to the market needs. And again, the company's success suffers from the original budgeting.

Based on these experiences, the Beyond Budgeting people developed different kinds of principles and recommendations. Neither of those necessarily mean to get rid of budgets but to use them more flexible. For example, negotiating annually on the budget is regarded as inflexible. Therefore, the recommendation is to use for example a rolling budget, which means to verify every month where the money is spent best. An alternative to the rolling budget is the event-based budgeting, thus whenever some changes are happening, the budget will be reconsidered.

One of the most important insights of beyond budgeting is in my opinion to differentiate between target and forecast. The rationale is that every goal should be ambitious whereas the forecast (or estimation) is a way to close the gap to the goal. Now if both -target and forecast- are forced into one number the result is that either the target isn't ambitious anymore or the estimation is a deception.

Applying Beyond Budgeting

In order to decide in favor or against a project or product, somebody has an idea and then asks maybe a senior developer or even a team for a number in terms of effort and therefore cost. The problem is that at this point in time not much is known about the project and therefore the estimations are not really solid. Sometimes it's sales who comes up with an estimation without consulting development and then the decision is made. (And of course no matter who comes up with this first estimation it's always assured toward development that this number will not be taken serious and it will be adjusted once we know more!)

Given the findings of both Kahneman and Beyond Budgeting this approach isn't helpful. First of all it ignores Beyond Budgeting's differentiation between target and forecast. Second, in contrary to Kahneman's insights the approach assumes that complex projects can be predicted and third, it suggests against Beyond Budgeting's advice to approve the budget upfront.

What we need instead is first to clarify the overall goal. In order to not rely just on one group of experts (remember Kahneman's findings), I recommend performing this clarification in a diverse group. Depending on the project this group should be assembled by e.g. the customer, marketing, sales, product management, and at best supported with scientific methods in order to find out the real needs of the market as Lean Startup suggests. The outcome of this clarification is not an estimation but the definition of the business value that the company (or the customer) imagines to achieve with the project. Looking at the business value shifts the mindset from what it will cost to what we will gain. The business value can be defined by discussing it, using a Delphi session or as I have done in the past as well by using a variation of planning poker. For the latter instead of using estimation numbers for driving the discussion, the group members use numbers referring to the business value.

Then instead of asking how long it will take, the same diverse group has to find out how much they are willing to spend (based on the business value they previously came up with). Naturally this group will struggle (at first) in a similar way coming up with a number for the investment as the developers do when coming up with an estimation. Yet, the business has to decide where it wants to spend the money. From the development point of view, everything can be made in a really high-sophisticated and expensive way (which typically is very much in line with the working ethics, speaking e.g. of clean code) but at the same time things can be implemented at low cost (which could lead to unmaintainable code). Yet, the business has to decide how much the business value is worth (which can result in asking for a cheap solution). This does not mean that there is no responsibility left for the developers in terms of planning - their task is to ensure business understands what impact a cheap (or expensive) solution might have. Consequently, estimation is not in the focus at that point. It is the decision on the business value and the investment the company wants to make that has to drive the development. Again, these drivers are created by the business and not by the development. Some teams just do the opposite, by asking development to not estimate on the story level only, but as well on the epic level (by estimating e.g. in T-Shirt sizes) or on the project level. Yet, trying to answer how much the project, epic, or story costs (or how long it takes) focuses on the wrong question. Again: the appropriate question to ask is how much is this worth and how much are we willing to invest in it.

This way, both the business value and the investment define a framework for steering the development. At best, both should be broken down to the story level yet at least to the epic level. For large projects, we have made good experiences planning on at least three levels. The highest level is the overall project plan (often called roadmap). The next level is the release plan, with a release enduring three months at maximum. Thus, before starting a release, the epics for this release will be defined in terms of business value

and investment. At the level of the iterations (the lowest level), the business value and investment for the stories is defined before the classic iteration planning happens (i.e. sprint planning one and two). Depending on the lengths of the release (and as well on the complexity of the project), we sometimes have the need for what Mike Cohn once called a “rolling lookahead plan.” Thus, we look into the business value and investment of the stories not only for the upcoming iteration but as well for the next two or even three iterations. Planning on these different levels provides the possibility for the business to define the business value and investment first at a coarse grained level and then iteratively making it more fine-grained. Defining it on a fine-grained level right away is on the one hand for large projects impossible and on the other hand not useful because it is very likely that the business value (and the reasons for the investment) will change over time.

Applying this approach at the different levels ensures that development is always driven by business value and investment and not by estimation. I still recommend to conducting a planning poker session for estimating the stories as a development team. Yet, this is done for creating the knowledge about the stories within the team, but not for driving development. Thus, the resulting estimation numbers are only important for driving the conversation, especially when team members at first disagree on the estimation numbers the discussion on their different assumptions will create a shared understanding of the story. This way, the estimation is still helpful for the team but it will not drive the planning. The prioritization and the definition what is in and what is out of scope of development is based on what the story is worth in terms of business value and investment (and not in terms of how long it will take or of how much story points it costs). Thus, business value together with investment provides a guideline for the prioritization and for the development. For example, if a specific story has assigned a low investment and doesn't provide a high business value the priority of the story has to be questioned (independent of the story points

estimated for it). Furthermore, it should be obvious that for such a story the development team should look for the cheapest solution and if there is none then the story has to be exchanged for one that provides a higher business value (and / or the company decides to make a higher investment).

As well the business value and the investment have to be verified regularly, similarly to the rolling or event-based budget. For example, at least after every other iteration look at what has been learned from the stakeholders (e.g. what kind of changes will provide a competitive advantage), from the market (e.g. where do we need to invest in order to create a higher business value) and as well from development (e.g. what advantage does a specific technology provide). Then analyze how this influences the business value and / or the investment on the different levels. It is helpful to first take a look at what this means for the release plan - often it affects only this level, so there is no need for taking it to the overall project level. Yet, at some times it has such an impact that as well the roadmap will be influenced by the changes. Certainly when defining the business value and investment for the upcoming iteration (and if applicable for the rolling lookahead plan, i.e. for the next two or three iterations) the new lessons learned should be taken into account.

In general, if the customer's competitive advantage is really our focal point then the business has to be driven by the business value. Thus far in agile development we have been talking about the business value but we just considered the priorities (often additionally under consideration of the estimates). Yet only the combination with the investment ensures that we always maximize the value in the customer's interest.

Conclusion

Recent findings in research together with the insights from Beyond Budgeting prove what many of us have experienced: Accurate forecasts aren't possible because the world is not predictable (especially not nowadays). Getting a prediction from an expert with high confidence just means that the expert has a coherent story and not that the accuracy of the prediction is high therefore instead of asking an expert, ask a diverse group of people.

For not falling into the trap of mixing estimation and planning, define a business value and come up with an investment you are willing to put into this undertaking and use these values for planning. These values help you to decide on the decision for or against starting a project and help you to come up with a roadmap for the project (or product) and a release plan. So especially on the project and on the epic level the business value and the defined investment will drive development.

Short-term predictions are possible, therefore measure the velocity and take this into account when planning the next iteration. The feedback of the iteration and of the stakeholders helps to improve the handling of both, the business value and the investment. So on the one hand the business value and investment steer the iteration and on the other hand the feedback of the iteration helps improve the business value and the investment. Or in other words: the roadmap influences the release plan which influences the iteration and vice versa - the result of the iteration feeds back into the release plan and this in turn into the roadmap. Therefore, the business value and the investment are treated as a rolling budget that is revisited regularly so all lessons learned will be considered.

References

Daniel Kahneman, *Thinking Fast and Slow*, Penguin Books, 2011

Bjarte Bogsnes, *Implementing Beyond Budgeting*, John Wiley & Sons, Hoboken, NJ. 2009

Beyond Budgeting Roundtable (Homepage of Beyond Budgeting):
<http://bbbt.org>

Mike Cohn, *Agile Estimating and Planning*, Prentice Hall, Upper Saddle River, NJ. 2006

A Focus on Continuous Value

Jutta Eckstein

Focusing continuously on the business value isn't possible!?

How can you maximize the business value in an agile project if you have to consider as well non-technical requirements. The focus of an agile approach is to maximize the business value constantly. Yet, how this can be done leads often to misunderstandings. Frequently people are asking how the following can possibly contribute to the business value:

- Fixing bugs?
- Refactoring?
- Non-functional requirements in general?

Those questions are motivated by some key misunderstandings: Namely that most people think solely about the end user if they think about business value. This way they ignore that next to the end user there are other kind of groups that have an interest in the system and for whom the business value has to be maximized as well.

Therefore, you have to clarify at first who are the stakeholders for the system. All these diverse groups (among them for example operations, maintenance, or other developers who are building on your system) have to be considered. You can do so by collecting as well their user stories and by asking them for feedback regularly. The key is to prioritize the different stories of the diverse stakeholders - this is the reason why the task of the product owner is extremely

challenging (and why it is almost impossible to find somebody who can take that role amongst your customers). This means the product owner has to weigh the demands of the diverse stakeholders and to come up with the right balance through prioritization.

The foundation for the weighting lies in clarifying the following:

- For whom does a particular story provide a business value (or rather: who is the stakeholder for a specific story)?
- How can you measure that business value?

The measurability is typically defined in terms of acceptance criteria and ideally implemented as acceptance tests. If the stakeholder and / or the acceptance criteria of a story are unclear no matter if this is a new story, a bug, or a refactoring an agile team can hardly maximize the business value. Or in other words: only the definition of the stakeholder and the acceptance criteria allow a team to keep the focus of an agile approach, which is maximizing the business value continuously.