Creating the Agile Portfolio

Rich Mironov, Mironov Consulting
Jonathan Propp, Red House Innovation

November 6, 2014

The Agile Portfolio

Since the publication of the Agile Manifesto in 2001, software companies have achieved significant improvements in developing and releasing new products by using Agile techniques. What started as a localized, project-by-project movement, however, has now reached the point where companies are trying to implement Agile across the enterprise. This has led to a new problem: the benefits seem to plateau as the scaling of Agile expands.

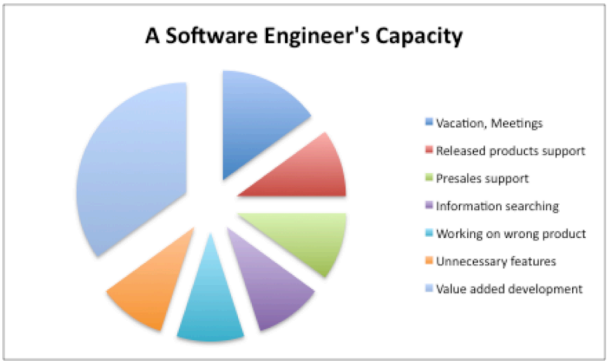
The problem is not with Agile and Scrum techniques themselves, but rather the overall environment in which companies are applying them. Individual development programs are not fully independent of each other. They exist within a *portfolio* utilizing scarce resources, namely developers, testers, product managers, UI designers, etc. And these portfolios have traditionally been managed in a non-Agile way.

We propose an Agile approach to planning the product portfolio. In other words, we apply the principles and techniques of Agile and Scrum to manage the business activity of allocating scarce resources to products. We believe the results can be transformative for software companies.

The Seven Wastes of Software Development

Mary and Tom Poppendieck, applying the ideas of the Toyota Production System, identified the "Seven Wastes of Software Development":

1. Building the wrong product
2. Unnecessary features
3. Excessive documentation
4. Partially done work (WIP)
5. Task switching
6. Waiting for information
7. Defects



While most Agile implementations focus on the last five wastes (the execution side), our experience has been that the biggest gains to be made are from the first two (the business side). Software companies have limited resources to work on product definition, development, and launch. Every hour spent working on the "wrong" product is one that can't be spent on a potentially

successful product. Furthermore, even the right product can become the wrong product if it is started too late and misses the market window.
Agile portfolio planning addresses both of these problems. It provides a way to accelerate and strengthen the management activity of choosing what to invest in and what not to invest in. It does so by applying the same rigor to portfolio management that Agile and Scrum apply to software development.

The Startup vs. The Established Company

Eric Ries' The Lean Startup presents an approach for emerging companies developing new software products, called "Minimum Viable Product." Ries' recommendation is to identify a potential product that meets a customer need, develop and test rapidly, and iterate constantly in response to customer feedback. The key question to be asked by management is "Is it valuable?" If the answer is yes, then we work on it.

The established company faces a different situation because it has multiple products with multiple releases in various states of their lifecycle. In an environment of limited resources, the question changes from "Is it valuable?" to "Is it *more* valuable?" The established company has multiple investment options and therefore needs to choose among them. So the decision making process becomes more complicated. The challenge is fundamentally an *allocation* problem.

Rush Hour in the Engineering Group

There is an additional reason why the lack of good portfolio management leads to poor business performance. A company can be investing in all the right products, but if it is overcommitting its development resources, than none of those products will be successful. This is an allocation problem of a different sort – an *overallocation* problem.

We start with a simple illustration. Ever drive down the freeway at rush hour? Suddenly the traffic slows to a crawl. A few miles later, it's back up to the speed limit, with no apparent accidents. What is going on?

Every operations manager is familiar with this scenario, known as *queueing theory*. When a process or system (in this case, a freeway with a certain number of lanes) is filled to capacity, then minor variations in service time (in this case, people entering or exiting, or switching lanes) will cause delays. As the system approaches 100% capacity, delays increase exponentially, and the whole system slows down. This is why manufacturing operations are typically run at less than 90% capacity with a focus on eliminating variation. (And this is why metering lights, which limit and time entrances to the freeway, make the biggest impact on rush hour traffic.)

In a software development organization, capacity is the number of hours of developer/tester and product manager time that you have. Many companies are,

consciously or unconsciously, running their software development systems close to or over 100% capacity. No matter how well you run individual projects, if you're overloading the system *every project slows down.* So despite companies' best efforts to implement Agile on a project-by-project basis, the benefits can be limited by inattention to the overall development portfolio.

Traditional Portfolio Planning

Most companies do not actively manage their development portfolio. They evaluate potential projects in a vacuum, looking at the risks and returns of a particular release. They generally fail to consider the *opportunity cost* of a project – the fact that putting developers and testers on a project means that they can't work on something else. Is that something else a better opportunity than the proposed project? Does it offer better returns, or does it better meet the company's strategic objectives?

Often a company's products will have some interaction with each other, further complicating the value analysis. Multiple projects, when combined, may support a strategic initiative such as entering a new market. By looking at projects individually, companies can fail to take into account the value of this synergy. Do the combined projects support an overall initiative to deliver specific value to targeted market segments, rather than trying to incrementally improve all products at the same rate for all audiences?

Engineering budgets are usually fixed during the annual operating planning (AOP) process. Once a year, Product Management and Engineering will propose a set of projects and estimate the resources required to complete them. Finance will usually send back a lower spending target and ask the product groups to make it work. Projects may be cut or scaled back. Either way, the resource pool is generally fixed for the next 12 months.

There are a number of problems with the tyranny of the AOP. First, we all know that product opportunities do not show up on an annual basis. They can show up any time in the year. And most software companies operate in an environment of short product life cycles, often *much* less than a year.

Secondly, the AOP "freezes" the development spending and makes it difficult to move funds around despite the changing needs of the business. Development managers are forced to do triage, moving resources around to meet the most pressing needs at the expense of (hopefully) less critical needs.

Finally, proposals for new or improved products tend to sit around waiting for the next AOP. Potentially useful work on refining the business case or developing prototypes doesn't get done. This work gets pushed into the fourth quarter, when all of a sudden the start of the AOP process forces everyone to do it.

An Agile Approach to Portfolio Planning

So while companies are increasing their development productivity by upwards of 50% using Agile and Scrum techniques, the benefits they are seeing are limited by a non-Agile business process for portfolio planning. But what would an "Agile Portfolio Planning" process look like?

Let's start by returning to the principles of Agile development:

1. Frequently review market needs and customer value, reprioritizing resources toward the greatest market value and most strategic products.
2. Shorten cycles by dividing the work into smaller increments; do small, frequent releases and then iterate.
3. Do not overplan – make high level estimates, then refine estimates as more information is known and the work becomes closer at hand.
4. Empower the team to make decisions.
5. Fully complete each piece of work (develop *and* test) before moving on to the next.

Taking each one in turn, we can construct a methodology for building the Agile Portfolio. We add a first step, however, since Agile is silent about strategy. Just as Agile methodologies have evolved to include a "Sprint 0" for architectural development or project planning, we add a "Step 0" to include the necessary strategic pre-work.

0. Create funding "buckets" based on the company's strategic goals.
We assume that the company has goals and a strategy, and that these are measurable. The strategy must also define the company's target markets, its goals in each market, and financial targets that dictate the amount of overall product development resources available. We then divide the available resources into buckets for:

- Longer-range new products, initiatives, or re-architecting
- Incremental features for next release of current products
- Quality improvements, including bug fixes and test infrastructure
- One-offs for Sales or Professional Services needs (sometimes called Solutions)
- General Engineering infrastructure

These buckets should remain generally consistent, but should be revisited quarterly as needs change. Funding should remain within the limits of the overall spending pie, unless a conscious decision is made to increase the spending for a specific opportunity.

1. Frequently review market needs and customer value, reprioritizing resources toward the greatest market value and most strategic products.
Proposed products represent a *portfolio backlog* of potential work that must be estimated and prioritized on a recurring basis, just as we do for user stories in a product backlog.

This must be driven by portfolio-level strategy about where we will compete and how we generate value.

2. Shorten cycles by dividing the work into smaller increments.
This means first of all replacing the three-month long process of creating the AOP with smaller, more frequent portfolio reviews. Most software companies we know should be replanning the portfolio on a quarterly basis. But instead of the detailed analysis spent on the AOP (down to person-hours), we recommend a more simplified analysis.

3. Do not overplan – make high-level estimates, then refine estimates as more information is known and the work becomes closer at hand.
Product managers generally need to create a business case to justify the investment in a project as part of the AOP process. These business cases are invariably wrong. First, it is hard to predict revenues and costs for a new release that may be as much as a year away. Secondly, revenue estimates are usually overoptimistic, as product managers get only one chance each year to make their business cases look good. The company ends up with bad data competing against bad data to justify investments.

We prefer a methodology where business cases reflect the uncertainty over market adoption. We acknowledge the imprecision of our estimates and ask for greater detail and certainty the closer in to product release. The business case evolves to higher precision as product priorities are validated, more customers are involved and knowledge increases.

4. Empower the team to make decisions.
In Agile software development, we do high level planning and then allow the team to figure out the details – what user stories to work on first, how to organize the work, and so forth. With Agile portfolio planning, we prefer to allocate funding into "buckets" based on product areas, leaving the specifics of release content up to the business group. Develop new features or solidify the last release? We're not going to specify. The product team knows best. What we care about is how much as a company we're investing in this product area vs. other product areas, based on our strategic and financial objectives.

5. Fully complete each piece of work (develop and test) before moving on to the next.
Portfolio planning is mostly a decision making process, a way to assess investment choices and select the most promising ones within the context of limited resources. Too many companies let these decisions lag or avoid them altogether. In our experience, most development projects are late not because of poor execution, but because it took management too long to decide whether to start (or cancel) the project or not.

In an Agile portfolio planning process, then, we complete the work – making decisions – and move on. We always have the ability to revisit the decision later. But we don't let decisions wait around. We also follow the Agile principle of leaving as much flexibility as possible, only committing as much as we need to in order to get the project started. In other words, we may commit to a Sprint 0 or definitional work, reserving a decision on further development until we have a working demo and more customer feedback.

Conclusion: Linking Strategy to Products

Too many companies have a strategy and a product portfolio, but the two are not well connected to each other. An Agile portfolio planning process fundamentally links company strategy to its product development activities. Companies do not devote enough time and resources to this activity today. They need to address questions such as:

- Who participates in portfolio management?
- What skills do they need?
- How do we know that the output of the process is of good quality?

We often say when speaking to product teams that an hour spent in good product planning saves a day in execution. The same holds true of portfolio planning. Agile and Scrum have been shown to help companies develop products faster and with better quality. Without Agile portfolio planning, however, Agile and Scrum will simply help you build the wrong products faster.

This change will require some redirection of resources away from execution into planning, but we think the rewards justify the effort. Companies will also want to rethink some of their product development metrics. Metrics focused on product execution throughput do not measure the success of portfolio planning. Instead, metrics focused on strategic alignment, portfolio balancing, and throughput of *planning activities* will lead to the right outcomes.

