Net Objectives
ESSENTIAL WHITE PAPERS

The Business Case for Agility

by Al Shalloway

The Business Case for Agility

by Alan Shalloway

A Net Objectives Essential White Paper

Net Objectives Press, a division of Net Objectives
1037 NE 65th Street Suite #362
Seattle, WA 98115-6655
404-593-8375

Find us on the Web at: *www.netobjectives.com*

To report errors, please send a note to *info@netobjectives.com*

Published by Net Objectives, Inc.

Notice of Rights

Notice of Liabilities

The information in this book is distributed on an "As Is" basis without warranty. While every precaution has been taken in the preparation of this book, neither the authors nor Net Objectives shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the instructions contained in this book or by the computer or hardware products described in it.

10 9 8 7 6 5 4 3 2

Printed in the United States of America

INTRODUCTION

Agile is often described as iteratively building software in increments. This is a focus on the team and the mechanics of how they work. It is more effective to focus on the reason for being Agile – achieving business agility. Business agility is the ability to deliver highest business value quickly, predictably, sustainably and with high quality. Building software is not our goal, realizing business value is. Software is often a component of this, of course.

Faster "time-to-market" is a well-known mantra. Achieving this by attempting to go faster is a common trap. Achieving value faster is better accomplished by focusing on what will truly provide the greatest value. Instead of going faster, we want to build those chunks of business value which are most important. That is, we will go smaller, focusing on those value adds that contain the most value for their efforts.

We call this the "Minimum Business Increment" (MBI). The MBI is a key to delivering value quickly by focusing on the most important business value to realize. It becomes a cornerstone concept in planning and alignment.

FREQUENCY OF DELIVERY VS. TIME TO DELIVER

We are all too familiar with organizations that cannot deliver value quickly. While deliveries may take place on a quarterly basis, the work being delivered has often been under development for a year. The issue is not how frequently one delivers, but rather what is the time frame from start to finish.

MVP vs MBI. This side bar is for those already familiar with the Lean-Startup concept of the minimum viable product, or MVP. While having a similar heritage, MBIs and MVPs are not the same.

An MVP is the version of a new product which allows a team to collect the maximum amount of validated learning about customers with the least effort. Its focus is on learning what is of value.

An MBI is the smallest chunk of business value that we can realize. Its focus is on delivering value faster. MBIs are also not limited to products but can be used in IT as well.

DELIVERING IN STAGES

To set the stage for incremental delivery of business value, it is important to understand why this is so beneficial. Mark Denne and Cleland-Huang put forth some compelling arguments for quick, frequent delivery in their book, *Software By Numbers*.

Figure 1 illustrates the economics of delivery they propose. This diagram illustrates how, at the beginning of a software project, we lose money during the investment period. After the product is released, we enter a payback period and hopefully go on to make enough money to incur profit.

For example, let's say I am the CEO of a company that does product development projects for other companies. Let's further say that I've received a request for proposal (RFP) that requires 100 features to be done in 10 months with a certain amount of quality and that the price is fixed.

Alan Shalloway is the founder and CEO of Net Objectives. With over 40 years of experience, Alan is an industry thought leader in Lean, Kanban, product portfolio management, Scrum and agile design. He helps companies transition to Lean and Agile methods enterprise-wide as well teaches courses in these areas. Alan has developed training and coaching methods for Lean-Agile that have helped Net Objectives' clients achieve long-term, sustainable productivity gains. He is a popular speaker at prestigious conferences worldwide. He is the primary author of Design Patterns Explained: A New Perspective on Object-Oriented Design, Lean-Agile Pocket Guide for Scrum Teams, Lean-Agile Software Development: Achieving Enterprise Agility and Essential Skills for the Agile Developer. Alan has worked in literally dozens of industries over his career. He is a co-founder and board member for the Lean Software and Systems Consortium. He has a Masters in Computer Science from M.I.T. and a Masters in Mathematics from Emory University. You can follow Alan on twitter @ alshalloway

Let's further suppose that the RFP stipulates that I cannot change any of these factors. That is, I need to deliver all 100 features by the date, not earlier nor later. That they must be of the quality specified, not better or worse, and that I must take exactly the money offered, not more nor less.

The purpose of an MBI is not to deliver only part of the system but rather to deliver value sooner – by focusing on the most valuable functionality.

It sounds like all of the factors in the iron triangle have been specified: scope, time and cost (with quality being thrown in as well). How could I win such a contract? Perhaps if my company had a more envious track record than my competitors I might win on the basis of credibility, but let's say that is not the case. Although it seems that all factors have been specified, there is another, that of rate of delivery.

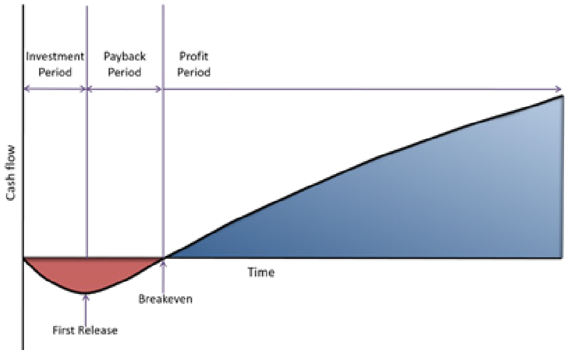


Figure 1. The Economics of Responsiveness

For example, I committed to delivering 50 features after five months and the other 50 features at the 10 months. I am complying with the requirement to deliver 100 features after 10 months; I am just delivering some earlier. What would this look like economically? If I assume delivering the first half of the features cost me half the work and delivered half the value, the investment and return graphs would look something like that shown in Figure 2. Even if these assumptions are not completely correct, let's explore this simple model and then discuss the issues that must be dealt with to make it more realistic.

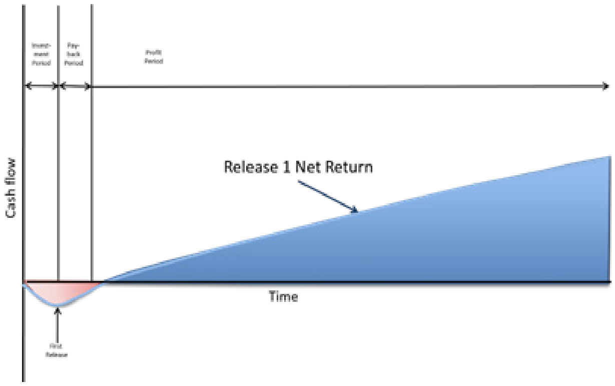


Figure 2. The Economics of Two Releases

While we cannot always do this staged release scheme, most software can be delivered in increments. For example, ignore the factors of extra development time for a staged release (which would make the cost curves deeper) as well as extra delivery and consumption costs for a staged release (also making the cost curves deeper). Figure 3 shows the combined cost curves in Figure 2 compared with the cost curve of the single release, the net cost of the stage release is less because the return from the first, partial release, offsets the investment required by the second partial release.

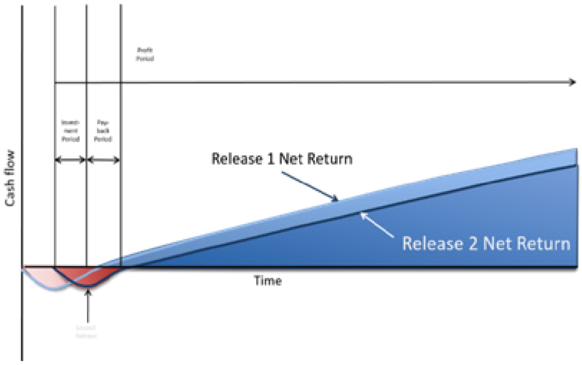


Figure 3. Comparing Phased and Single Release Cost/ROI Curves

TARGETED MARKETS AND INCREASED ALIGNMENT

When considering what 'minimum business increment' we should select, we have found it very useful to look at the different scenarios in which the new capability can be used. Very often each of these scenarios will have different market segments using them. MBIs can be selected on the basis of which scenarios, and which markets should be targeted for maximum value delivery.

There is a powerful side effect to doing this. By creating MBIs that represent focused business value, they can be sequenced in order of importance to the organization. That is, those MBIs that deliver the greatest value soonest can be sequenced as a higher priority than those that don't. This alignment of what is of greatest business value can also be used to align disparate teams in building things in this same order – thereby working together more effectively.

Note how MBIs are fundamentally different from Epics. First, business folk typically don't know or care what an epic is. This is with good reason, I can have strong feelings about a car but never car how fuel injection works. MBIs, however, are oriented toward business stakeholders. Epics, on the other hand, are an Agile construct that represents a big story – much of which has little value (which is why we build things in stages – the most important part of the epic first). An MBI, on the other hand, is an atomic unit of value and this enables it to provide deeper agreement on if and when it should be built.

THE ISSUES OF STAGED DELIVERY

Of course, it is not as simple as this. Building the software in stages may take more effort than building it in one pass. Furthermore, two deliveries may cost more as well as be more disruptive for customers. These factors would all take the return curves down by adding more cost. However, there are some other factors that might vastly increase the return curves. However, I'll deal with these increased cost considerations first, before discussing other potentially even greater advantages to phased deliveries.

Extra cost of building phased deliveries

While this is of concern to many people, it does not have to cost more to building stages. If you are extending a legacy system that has large amounts of technical debt and is difficult to test, then multiple deliveries may, in fact, cost more. But if your code quality is good and you have automated tests, it often costs less to build and deliver in stages. There are several reasons that contribute to this. One is phased deliveries can result in not building certain function that at first appears necessary, but with the feedback from the first phase, one learns it is not as desirable as first thought. Also, the use of emergent design can actually speed up phased delivery. Emergent design is beyond the focus of this book. The interested reader is referred to Scott Bain's excellent book, *Emergent Design: The Evolutionary Nature of Professional Software Development.*

Extra cost of deployment

This again can be an extra cost, but it does not necessarily have to be. Making an investment in being able to release and deploy quickly is important so as to take advantage of the extra return it affords.

Extra cost of consumption

While there are some applications which users do not want updated on a regular basis, studies have shown that most users prefer small, frequent upgrades to large, infrequent ones. The salient characteristic, of course, is the level of disruption to the user. If the consumption by the user is small enough, they won't mind it. Most objections to upgrades are due to the effort required by the user to do the upgrade.

THE POTENTIAL RETURN OF STAGED DELIVERY

Of course, in the real world, when we can make staged deliveries, we won't arbitrarily pick half the features and release them. Instead, we would pick those features that made the most sense to deliver. This would include looking at the factors of – where's the greatest value and what will it cost to build, deploy and consume the release. Clearly the smallest release that is useful to the customer and worth the incremental cost of building and deploying is what we should do. We prefer to call this the "Minimum Business Increment" or MBI. Others have used the term Minimal Marketable Feature (MMF), Minimal Viable Feature (MVF), and Minimal Viable Product (MVP).

These are not just name changes. They also indicate intent of the phase. Eric Ries talks about MVPs as a way to test the market – both to see if your product is viable and to gain quick entry. While we talked about how the return curves for a phased release may not be as good as illustrated, there are some other considerations that may make them considerably better. In a highly competitive situation, for example, whoever gets into the marketplace first may capture it. In the earlier RFP example, if our client had a competitor that is going to release in eight months, the corresponding values of phased vs. full releases may look like Figure 4.

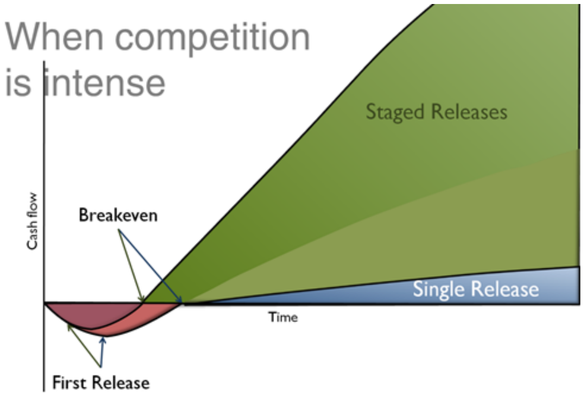


Figure 4. Illustrating Differences Competition Can Make

In this extreme case, gaining entry into the market 3 months before our competition may enable our client to own it. Coming in 2 months after their competition may be an effective barrier of entry. Another advantage is that customers will be able to provide greater feedback to improve the effectiveness of the subsequent phases of delivery.

OPPORTUNITIES AND THE NEED TO LOOK ACROSS PRODUCTS

I have always disliked the concentrated focus that XP and Scrum provide between the team and the customer. While at the team level this focus is appropriate, it has always implied a myopic view that the focus is on the customer and the team. In reality, a broader view is needed. Within that broader view, the focus of the team to the customer is good, but when the context within which this exists is ignored this focus can be problematic.

To understand this, one needs to consider different possible rate of returns one can have on projects. The ideal case is what is called the "Pareto curve." This occurs when most of the value is achieved by doing 20% of the work. That is, a small investment can return a large gain. This is illustrated in Figure 5.

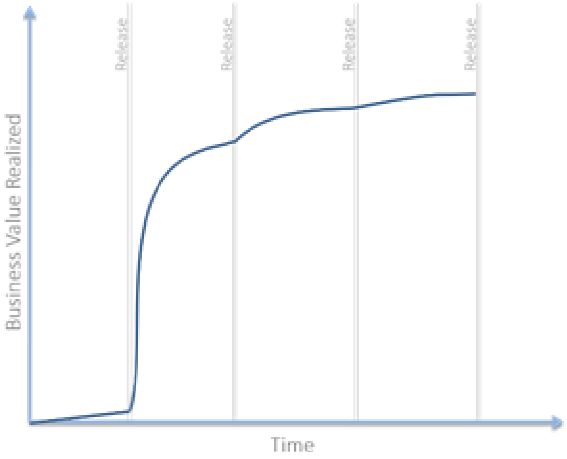


Figure 5. The Pareto Return

This example illustrates the case of when we get a big return from the first release. Each subsequent release gives us less and less in return. Although customers who are using this product may want us to continue working on this, we run the risk of missing opportunities for other products that can give a big return on its first release. In other words, instead of getting the tail end of the Pareto curve, we may want to look for other products that will give us a big return for relatively little effort.

We often see budgeting errors in this situation. Many times companies continue to provide funding for established products because they follow the logic of basing budgets along the lines the returns are occurring in. In reality, budgets should be made along the lines of what future returns would be available. Of course, one must consider what will happen to existing systems if one does not continue to enhance them at all. But even this is looking at investment on the basis of future returns, not on past laurels or on the idea that a team belongs to a customer.

Looking forward in this way opens up the opportunity for better product portfolio management. We can pick those features, regardless of which product they are for, that produces the greatest value to the organization.

Of course, not all products give such a Pareto return. Many give fairly linear results – each release providing an amount of value pretty much in line with the effort it took to achieve it. Some product enhancements return value relatively quickly; others take a few months to build. What we want to avoid is long delayed value simply because we didn't consider the potential of incremental business delivery.

THE LEAN-STARTUP MOVEMENT

It is worth taking some lessons from the Lean-Start up movement here. I've been discussing return of business value of mostly established products. The Lean-Startup movement, however, suggests delivering business value incrementally in order to determine which products actually have value. In other words, delivering part of a system, that may not be complete yet but which will provide some value to customers while indicating how much value the product can eventually provide is a useful consideration. The Lean Startup approach using Minimum Viable Products (MVPs) is a different, albeit useful, tact on incremental business value.

Both MBIs and MVPs have the same heritage of Denne and Cleland-Huang's MMF. Having both in your arsenal of tools can be quite powerful. But it is important to know the difference. MVPs a la Lean-Startup are designed for startups and focused around the discovery of new product value. MBIs are designed for all companies of any maturity and is focused on increasing value realized by the business. Both allow for pivoting as discovery of the true value of the product is delivered.

OTHER ARTICLES OF INTEREST

Go to *www.netobjectives.com/articles* to see the following articles:

- Net Objectives Approach to SAFe®
- Why Tailored Agile Transformations Are More Effective, Less Expensive and Less Risky

Check out the Net Objectives Portal at *portal.netobjectives.com* where an extensive amount of online self-study is available.

LEARN TO DRIVE DEVELOPMENT FROM THE DELIVERY OF BUSINESS VALUE

What really matters to any organization? The delivery of value to customers. Most development organizations, both large and small, are not organized to optimize the delivery of value. By focusing the system within which your people are working and by aligning your people by giving them clear visibility into the value they are creating, any development organization can deliver far more value, lower friction, and do it with fewer acts of self-destructive heroism on the part of the teams.

THE NET OBJECTIVES TRANSFORMATION MODEL

Our approach is to start where you are and then set out a roadmap to get you to where you want to be, with concrete actionable steps to make immediate progress at a rate your people and organization can absorb. We do this by guiding executive leadership, middle management, and the teams at the working surface. The coordination of all three is required to make change that will stick.

OUR EXPERTS

Net Objectives' consultants are actually a team. Some are well known thought leaders. Most of them are authors. All of them are contributors to our approach.

Al Shalloway

Alan Chedalawada

Guy Beaver

Scott Bain

Max Guernsey

Luniel de Beer

SELECTED COURSES

Executive Leadership and Management

- Lean-Agile Executive Briefing
- Preparing Leadership for a Lean-Agile/SAFe Transformation

Product Manager & Product Owner

- Lean-Agile Product Roadmaps
- PM/PO Essentials

Lean-Agile at the Team

- Acceptance Test-Driven Development
- Implementing Team Agility
- Team Agility Coaching Certification
- Lean-Agile Story Writing with Tests

Technical Agility

- Advanced Software Design
- Design Patterns Lab
- Effective Object-Oriented Analysis and Design
- Emergent Design
- Sustainable Test-Driven Development

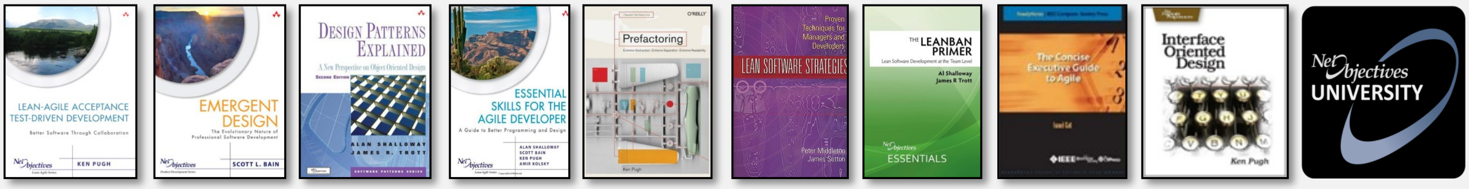
DevOps

- DevOps for Leaders and Managers
- DevOps Roadmap Overview

SAFe®-Related

- Implementing SAFe with SPC4 Certification
- Leading SAFe® 4.0
- Using ATDD/BDD in the Agile Release Train (workshop)
- Architecting in a SAFe Environment
- Implement the Built-in Quality of SAFe
- Taking Agile at Scale to the Next Level

OUR BOOKS AND RESOURCES


LEAN-AGILE ACCEPTANCE TEST-DRIVEN DEVELOPMENT
KEN PUGH
EMERGENT DESIGN
SCOTT L. BAIN
DESIGN PATTERNS EXPLAINED
ALAN SHALLOWAY
JAMES R. TROTT
ESSENTIAL SKILLS FOR THE AGILE DEVELOPER
Prefactoring
LEAN SOFTWARE STRATEGIES
THE LEANBAN PRIMER
ESSENTIALS
The Concise Executive Guide to Agile
Interface Oriented Design
Ken Pugh
Net Objectives UNIVERSITY

CONTACT US
info@netobjectives.com
1.888.LEAN-244 (1.888.532.6244)

LEARN MORE
www.NetObjectives.com
portal.NetObjectives.com