

Vol. 22, No. 1  
January 2009

**“I believe that we must begin by evolving our organizations toward *adaptive governance* and away from plan-driven governance, just as agile projects have evolved toward adaptive management and away from plan-driven management.”**

**— Sanjiv Augustine,  
Guest Editor**

## Using Lean Portfolio Management to Scale Agile Methods

### You Can't Make a Silk Purse Out of a Sow's Ear

Agile methods don't scale and will be passé in a year or three. Hitching them to lean won't change anything.

### The Wind Beneath Agile's Wings

Despite industry-wide adoption of agile methods at the project level, sustained large-scale agile initiatives are fewer and further between. But adaptive, lean governance of programs and portfolios will lift agile to new heights and help transform IT governance overall.

Opening Statement by Sanjiv Augustine .....	3
Portfolio Management and Agile Software Development by Scott W. Ambler .....	6
Selecting a Ranking Method for Your Project Portfolio by Johanna Rothman .....	13
Agile Planning Over the Long Term: The Portfolio Planning Game at TWeb by Jens Coldewey .....	19
Lean Portfolio Management at SciQuest by Jamie Duke and Sam Bayer .....	26
Transformative IT: Creating Lean IT Portfolio Management ... or Not by Bob Benson and Tom Bugnitz .....	31

## About Cutter IT Journal

Part of Cutter Consortium's mission is to foster the debate of, and dialogue on, the business technology issues challenging enterprises today, to help organizations leverage IT for competitive advantage and business success. Cutter's philosophy is that most of the issues that managers face are complex enough to merit examination that goes beyond simple pronouncements. Founded in 1987 as *American Programmer* by Cutter Fellow Ed Yourdon, *Cutter IT Journal* is one of Cutter's key venues for debate.

The monthly *Cutter IT Journal* and its weekly companion *Cutter IT E-Mail Advisor* offer a variety of perspectives on the issues you're dealing with today. Armed with opinion, data, and advice, you'll be able to make the best decisions, employ the best practices, and choose the right strategies for your organization.

Unlike academic journals, *Cutter IT Journal* doesn't water down or delay its coverage of timely issues with lengthy peer reviews. Each month, our expert Guest Editor delivers articles by internationally known IT practitioners that include case studies, research findings, and experience-based opinion on the IT topics enterprises face today — not issues you were dealing with six months ago, or those that are so esoteric you might not ever need to learn from others' experiences. No other journal brings together so many cutting-edge thinkers or lets them speak so bluntly.

*Cutter IT Journal* subscribers consider the *Journal* a "consultancy in print" and liken each month's issue to the impassioned debates they participate in at the end of a day at a conference.

Every facet of IT — application integration, security, portfolio management, and testing, to name a few — plays a role in the success or failure of your organization's IT efforts. Only *Cutter IT Journal* and the *Cutter IT E-Mail Advisor* deliver a comprehensive treatment of these critical issues and help you make informed decisions about the strategies that can improve IT's performance.

*Cutter IT Journal* is unique in that it is written by IT professionals — people like you who face the same challenges and are under the same pressures to get the job done. The *Journal* brings you frank, honest accounts of what works, what doesn't, and why.

Put your IT concerns in a business context. Discover the best ways to pitch new ideas to executive management. Ensure the success of your IT organization in an economy that encourages outsourcing and intense international competition. Avoid the common pitfalls and work smarter while under tighter constraints. You'll learn how to do all this and more when you subscribe to *Cutter IT Journal*.

### Cutter IT Journal®

*Cutter Business Technology Council:*  
Rob Austin, Ron Blitstein, Christine Davis, Tom DeMarco, Lynne Ellyn, Jim Highsmith, Tim Lister, Lou Mazzucchelli, Ken Orr, Mark Seiden, Ed Yourdon

*Editor Emeritus:* Ed Yourdon  
*Publisher:* Karen Fine Coburn  
*Group Publisher:* Chris Generali  
*Managing Editor:* Karen Pasley  
*Production Editor:* Linda M. Dias  
*Client Services:* [service@cutter.com](mailto:service@cutter.com)

*Cutter IT Journal*® (ISSN 1522-7383) is published 12 times a year by Cutter Information LLC, 37 Broadway, Suite 1, Arlington, MA 02474-5552, USA (Tel: +1 781 648 8700; Fax: +1 781 648 8707; E-mail: [citjeditorial@cutter.com](mailto:citjeditorial@cutter.com); Web site: [www.cutter.com](http://www.cutter.com)).

©2009 by Cutter Information LLC. All rights reserved. *Cutter IT Journal*® is a trademark of Cutter Information LLC. No material in this publication may be reproduced, eaten, or distributed without written permission from the publisher. Unauthorized reproduction in any form, including photocopying, faxing, image scanning, and downloading electronic copies, is against the law. Reprints make an excellent training tool. For information about reprints and/or back issues of Cutter Consortium publications, call +1 781 648 8700 or e-mail [service@cutter.com](mailto:service@cutter.com).

Subscription rates are US \$485 a year in North America, US \$585 elsewhere, payable to Cutter Information LLC. Reprints, bulk purchases, past issues, and multiple subscription and site license rates are available on request.

Start my print subscription to *Cutter IT Journal* (\$485/year; US \$585 outside North America)

Name _____	Title _____
Company _____	Address _____
City _____	State/Province _____ ZIP/Postal Code _____

E-Mail (Be sure to include for weekly *Cutter IT E-Mail Advisor*) \_\_\_\_\_

Fax to +1 781 648 8707, call +1 781 648 8700, or send e-mail to [service@cutter.com](mailto:service@cutter.com). Mail to Cutter Consortium, 37 Broadway, Suite 1, Arlington, MA 02474-5552, USA.

**SUBSCRIBE TODAY**

### Request Online License Subscription Rates

For subscription rates for online licenses, contact us at [sales@cutter.com](mailto:sales@cutter.com) or +1 781 648 8700.



by Sanjiv Augustine

# Opening Statement

Agile methods (Scrum, XP, Crystal, DSDM, Feature-Driven Development, etc.) have moved into the mainstream over the past few years. Earlier in their adoption cycle, industry leaders such as BMC Software, British Telecom, Capital One, DTE Energy, and Yahoo! discovered that, when implemented appropriately, agile methods accelerate project delivery times, increase customer and employee satisfaction, and enable flexible changes in business requirements. These companies adopted agile methods and, as a result, realized faster throughput and higher business customer satisfaction on individual projects. Based on this evidence and growing success, the move toward agile methods is now a widespread industry phenomenon.

Yet despite undeniable wins with agile at the project level in many quarters, sustained large-scale initiatives to adopt agile methods are fewer and further between. Agile adoption initiatives are known to run into hurdles when the company culture is at odds with core agile values. VersionOne's 2008 *State of Agile Development Survey*<sup>1</sup> identifies the inability to change organizational culture as the leading barrier to further agile adoption. What are some of the organizational factors that are at odds with agile values and are preventing further adoption of agile within organizations?

One factor is IT governance. Many experts in the agile space believe that there is now a significant misalignment between the way agile projects are run and the way IT projects are governed in general. IT program and portfolio management, in particular, seem to be at the root of many of these alignment issues. As a consequence, corporate project portfolios remain challenged in many organizations. Executives still see projects that are late and overbudget, deliver poor value, and have less-than-satisfied business sponsors and end users. How can we scale agile methods beyond individual projects so that the programs and portfolios within which they exist can benefit unequivocally? I believe that we must begin by evolving our organizations toward *adaptive governance* and away from plan-driven governance, just as agile projects have evolved toward

adaptive management and away from plan-driven management.

In most organizations that have adopted agile methods, the techniques used for program and portfolio management are still predictive or plan-driven. They consist of yearly budgeting cycles and capacity planning and heavily matrixed resource management. It's no surprise, then, that despite adopting agile methods for their projects, many organizations have yet to exploit their full benefits at the portfolio level. Or to put it another way, agile projects are constrained because portfolios are clogged with the debris of failing projects and with slow-moving projects that delay more critical initiatives. Consequently, organizations are unable to properly staff projects and deliver them at a rate that their business customers would appreciate.

**Agile projects are constrained because portfolios are clogged with the debris of failing projects and with slow-moving projects that delay more critical initiatives.**

How can the project portfolio be unclogged so that business value can flow and projects can complete faster? How, specifically, can program and portfolio management be improved? In this issue of *Cutter IT Journal*, our contributing authors explore a variety of different approaches to this challenge.

In our first article, Scott Ambler explores how portfolio management and agile software development can not only coexist peacefully, but actually enhance each other. Ambler presents an overview of both portfolio management and agile software development and then explores how portfolio management fits into the agile project lifecycle. Ambler takes a comprehensive view of that lifecycle, noting, "Many people like to solely focus on the predevelopment and development aspects of portfolio management, but that would be a mistake."

Indeed, he sees portfolio management beginning with “Iteration -1” (a pre-preproject phase in which the organization weighs the risks and decides to pursue a project) and ending only with system retirement. Regardless of what stage you are at, Ambler argues that the greater project visibility and increased stakeholder control offered by agile development will “enable effective monitoring and management of the development projects within your IT portfolio.” He offers an IT asset classification approach that can help guide organizations’ refactoring efforts, and wraps up by relating portfolio

management to other enterprise disciplines, including enterprise architecture, enterprise business modeling, and strategic reuse of assets across the portfolio.

Next, Cutter Senior Consultant Johanna Rothman delves into the details of selecting a ranking method for your project portfolio. Rothman begins by recommending that you ask the hard question of whether the project should exist at all, and she closes with the sage advice that you not forget to kill non-performing projects. In between, she tells how you can rank projects by assigning them a point value, weighing the risk of doing (or not doing) them, reviewing the project context, using double- or single-elimination methods, or assessing them in light of your organization’s mission and values. To get an idea of when each of these methods is called for, we follow Dave the CIO and his team on their portfolio management journey, seeing how they employ the various ranking methods as their circumstances dictate. And if committing to a definite project ranking seems too daunting, Rothman reminds us that ranking “isn’t forever” and that “you have only to wait for one iteration to finish until you can rerank your portfolio.” Knowing they have that flexibility, even commitment-phobes will find it easier to make the necessary project ranking decisions.

In our third article, Cutter Senior Consultant Jens Coldewey walks us through his agile portfolio planning experience at TWeb, a product company in the travel industry. TWeb originally hired Coldewey to introduce XP, with the aim of improving the productivity of their development team. However, it didn’t take him long to discover that TWeb had much bigger problems, including a planning process in such disarray that “measured on a CMM® scale, this company was definitely on Level 1 — chaos — for the single reason that CMM does not define a Level 0.” Some of TWeb’s customers had been waiting on promised features for up to three years — and were starting to call in their lawyers and/or publicly threatening to decamp. Coldewey describes how TWeb took their first steps toward introducing basic agile planning and how success in stabilizing their short-term plans eventually led to a portfolio planning game designed to forecast into the next two years. The devastating picture that emerged led to a tense confrontation, a momentous decision — and the resignation of one angry executive. Coldewey concludes by bringing us up to date on what has happened at TWeb in the 15 months since then.

Next up, we have another case study — but all resemblance between the last article and this one ends there. In contrast to TWeb, with its development chaos and

## IN NEXT MONTH’S ISSUE

### Managing Enterprise Risk in a Failing Economy: Is It Time for Risk Management 3.0?

Guest Editor: Robert N. Charette

Many economists believe that the risks present in the current economic downturn have the potential to repeat the Depression years of the 1930s. Governments across the globe are struggling to stabilize their individual economies from the financial contagion that started with subprime mortgages and spread all the way to the corporate and consumer credit markets. As governments commit trillions of dollars in coordinated risk mitigation efforts to try to prevent a total global economic meltdown, ironically, fingers are also pointing at risk (mis)management as a major cause of the current economic crisis. Poor risk management models have been blamed, at least in part, for the problems at Lehman Brothers, Bear Stearns, and UBS — which in the first two cases led to their demise. Investor Warren Buffet recently summed up the skepticism about complicated quantitative risk management models, noting: “All I can say is, beware of geeks ... bearing formulas.”

In next month’s issue of *Cutter IT Journal*, we’ll debate the role of risk management in the current economic crisis. You’ll hear from one author who pins the economic collapse not on flawed risk management models, but on the lack of moral fiber in executive suites and boardrooms. You’ll learn how the “predictable irrationality” of human beings is to blame for the waxing and waning of enterprise risk management efforts — and what you can do to stabilize risk management practice in your organization. And you’ll discover the three enterprise risk management gaps you must close to help your organization withstand risks and ultimately improve the creation — and protection — of shareholder value.

Have the reports of risk management’s death been greatly exaggerated? Tune in next month and find out.

customer defections, SciQuest is a fast-growing software as a service (SaaS) company that has embraced a Scrum-based, agile product development process and enjoys near 100% customer renewal rates. But even award-winning companies can find their numbers going in the wrong direction, as SciQuest's customer support department did last year. SciQuest COO Jamie Duke and Cutter Senior Consultant Sam Bayer narrate the story:

In the case of our customer support department, our mandate was clear. We needed to reduce the number of customer incidents reported per module per month by 20% in Q4 2008 as compared to Q4 2007. ... At the end of Q1 2008, we were startled to learn that not only had we not improved over the previous quarter, but in fact our performance had deteriorated significantly.

Customer support responded by assembling a cross-functional team in a two-day workshop to identify projects that could get the department back on track. Taking a lean portfolio management approach, the team came up with 17 potential incident-reducing projects and graded them on an A+ to C scale. Obviously the A+ projects — those with “the highest potential to remove the greatest number of incidents in the shortest period of time with the least amount of cost and risk” — were tackled first, and by the end of Q2 2008, the number of reported incidents per month had improved 22%. But where SciQuest's lean portfolio management approach really showed its mettle was not in these successes, but in one notable failure. When customers rejected SciQuest's efforts to productize internal tools so the customers could debug their own workflow snags, the company responded to the feedback by redeploying resources to higher-value projects in the portfolio.

Last but not least, Cutter Senior Consultants Bob Benson and Tom Bugnitz explore how lean portfolio management *might* be transformative for IT. I say “might,” because the authors clearly feel that the jury is out on this question. They see two barriers to the widespread adoption of lean portfolio management:

The first is that lean portfolio management touches on the fundamental issue in IT decision making: justifying and reconciling conflicting demands for resources. This may require more structure and deliberation than lean portfolio management is prepared to provide. The second is that effective business management participation in this decision making may be hard to come by.

Benson and Bugnitz speak to the very real challenges of getting business partners on board who may not “want to spend the time and energy or risk the political

exposure.” They also analyze whether lean portfolio management is compatible with corporate IT decision-making processes. The authors then propose two approaches that may help us begin to define a lean portfolio management process — one that relies on Warren McFarlan's “IT importance” quadrants and another based on the cultural assessment method they've devised in their 25 years of consulting. They conclude with the hard truth that lean portfolio management must enable organizations to kill off failing (or simply low-value) projects, for “without the possibility that projects can be stopped, management will have no stake in the process.”

All our authors bring us valuable portfolio management perspectives and techniques as the industry attempts to figure out how to scale agile methods, sustain agile adoption over the long term and across the enterprise, and provide adaptive governance of our IT programs and portfolios. I'm confident that you will be able to glean insights from their work that will aid you in your own agile journey.

Happy Reading!

## ACKNOWLEDGMENTS

I extend special thanks to Chris Generali and Karen Pasley of Cutter Consortium for their advice and invaluable review and editing assistance.

## ENDNOTE

<sup>1</sup>3rd Annual State of Agile Development Survey. VersionOne, August 2008 ([www.versionone.com/AgileSurvey2008/index.asp](http://www.versionone.com/AgileSurvey2008/index.asp)).

*Sanjiv Augustine is a Senior Consultant with Cutter's Agile Product & Project Management Practice. He is the President of LiteSpeed, an agile consulting, development, and training firm, a cofounder and advisory board member of the Agile Project Leadership Network (APLN), the author of several publications and the book Managing Agile Projects, and the founder and moderator of the Yahoo! Agile Project Management discussion group. He presents regularly at several agile and project management conferences worldwide, including ProjectWorld and Project Summit. Mr. Augustine has assisted clients in deploying agile and lean methodologies at many institutions large and small. As a management coach and consultant, he has advised executives in the enterprise rollout and adoption of agile at several Fortune 250 companies. As an in-the-trenches practitioner, he has personally managed agile projects varying in size from five to more than 100 people and coached numerous project teams. He can be reached at [saugustine@cutter.com](mailto:saugustine@cutter.com).*



# Portfolio Management and Agile Software Development

by Scott W. Ambler

The goal of portfolio management within an IT environment is to help improve the overall efficiency and effectiveness of IT efforts within an organization. You do this by ensuring that all projects and existing systems are visible, planned for, and aligned to the goals of your organization. Critical activities within your portfolio management discipline include project identification and selection, project monitoring and governance, and IT inventory management. Enterprise disciplines, such as portfolio management, are important because successful IT departments look beyond the needs of a single system. Studies have shown that organizations that manage their IT investments most successfully generate as much as a 40% higher return than their competitors.<sup>1</sup> Portfolio management is arguably one of the most misunderstood aspects of IT activities, perhaps because of its cross-system scope and because there is so little written about it at the practitioner level. My goal with this

article is to shed some light on this mysterious discipline, explain how it can be an effective aspect of your agile endeavors, and show how you can be more agile in your approach to portfolio management.

Defining agile software development isn't as straightforward as one would think, which is one of several reasons that there still isn't an industry-standard definition. Many people point to the values of the Agile Manifesto<sup>2,3</sup> as a definition, yet for me those are interesting philosophies (to which I wholeheartedly subscribe), but they're not a very good definition. Some people simply state that you know agile when you see it — which is a great definition for the agile consultants among us, but not so great for people who are still trying to understand what agile is all about. The definition for agile software development presented in the sidebar is the one that we developed within IBM Software Group and promote with our

## TERMINOLOGY DEFINED

I'd like to define several important terms used throughout this article:

- **Agile software development** is a disciplined, evolutionary (iterative and incremental) approach to development that regularly produces high-quality software in a cost-effective and timely manner. It is performed in a highly collaborative and self-organizing fashion, with active stakeholder participation to ensure that the team understands and addresses the changing needs of its stakeholders. Agile development teams provide repeatable results by adopting just the right amount of ceremony for the situation they face.
- **IT portfolio** is a collection of IT projects, both proposed and in progress, as well as the deployed systems within your organization. Your IT portfolio should be a diversified mix of high-risk/high-reward and low-risk/low-reward elements, all leading to the creation of systems to support a strategic business goal.
- **Portfolio management** is the management of all the various programs within your IT portfolio.
- **Agile portfolio management** is a lightweight and highly collaborative approach to portfolio management.
- **Program management** is the management of a subset of projects within your portfolio. Large organizations will often choose to organize related groups of projects or systems into programs; for example, a financial institution may have banking, insurance, online brokerage, and private banking programs within its portfolio. Doing so can help create synergy between similar efforts and offer a more efficient mechanism for managing and reporting projects.
- **Project management** is the management of a single project. A project may exist outside of any particular program, but it still should be identified by the portfolio manager.

customers worldwide. Although this definition describes what we believe to be the essential aspects of agile, we still find that many of our customers struggle to identify which of their teams, if any, are taking an agile approach. We believe that to claim a project team is taking a disciplined approach to agile development, it must conform to all of the following criteria:

- **Working software.** The team delivers a potentially shippable working system for each iteration.
- **Regression testing.** Minimally, developers do regular regression testing as part of their continual integration efforts. Better yet, they're doing test-driven development (TDD) and independent parallel testing<sup>4</sup> to address scaling issues.
- **Teams are self-organizing within a reasonable governance framework.** Agile team members should be actively involved with detailed planning and estimating; they should not just have plans and estimates provided to them by others. Agile teams will often do two-level planning, including both high-level release planning and detailed iteration planning; take a risk/value driven approach to development; collect simple and relevant metrics; and continually monitor those metrics.
- **Active participation by stakeholders.** Stakeholders should actively participate in the day-to-day activities of the project team. Ideally, stakeholders should control the budget, scope, and prioritization of requirements and be able to adjust these things throughout the project.
- **Regular improvement.** Members of the team regularly reflect on how they work together and then act to improve on their findings.

I'd like to make several important observations. First, there is nothing inherent to agile software development that means agile projects cannot fit into a portfolio management framework. Second, nor is there anything that implies that all software development projects within a portfolio need to follow the same approach. Third, the choice of development paradigm for a potential system is an important decision point in project identification and selection because of the different risk and benefit profiles of each paradigm. Fourth, because different projects will follow different development paradigms, your project monitoring efforts must be flexible enough to handle the range of projects that your organization is doing.

### PORTFOLIO MANAGEMENT THROUGHOUT THE PROJECT LIFECYCLE

Although portfolio management is a cross-system enterprise discipline, it is valuable to consider it from the point of view of a single system. Figure 1 depicts this viewpoint, showing that portfolio management occurs throughout the entire lifecycle of a system — from before its inception through development, production, and even retirement. Perhaps one of the reasons most IT professionals misunderstand portfolio management is that so much of it occurs beyond the scope of software development. Project identification and selection occur before the development effort, and much of the monitoring effort occurs afterward (production and system retirement).

Organizations have limited resources and many opportunities in which to invest those resources. As a result, you must choose wisely when starting up a potential IT project if you want to truly maximize stakeholder ROI. A common concept within the agile community is that

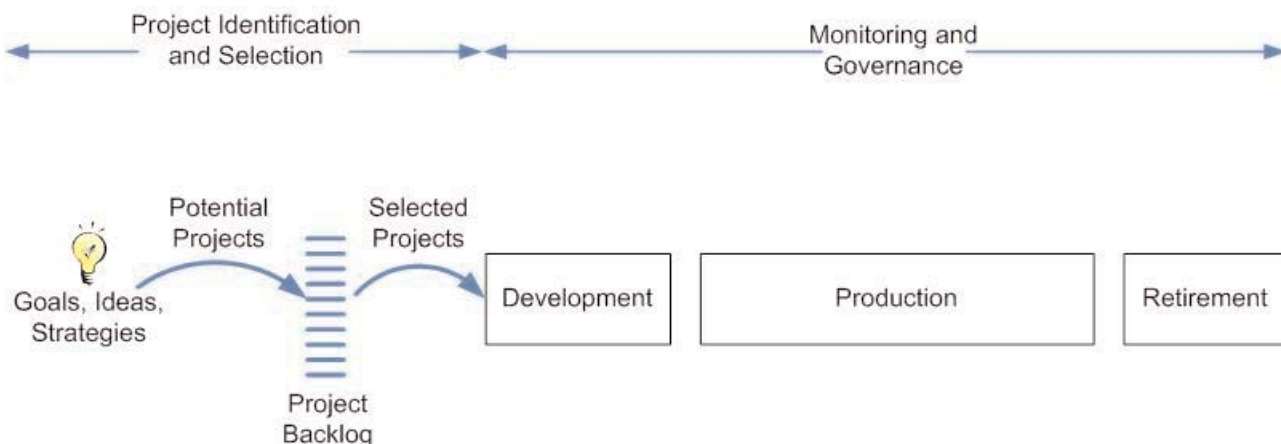


Figure 1 — Portfolio management activities throughout the system lifecycle.

of “Iteration 0,” which is the initial part of the development portion of the system lifecycle.<sup>5</sup> It is during this iteration that you do the fundamental work of getting your project team organized, including initial requirements and architecture envisioning, bringing people into the team, setting up workstations, and initial high-level release planning. Iteration 0 presumes that someone has already made the decision to go ahead with the project, so this “first” iteration really *isn’t* the first, because work must occur before it can begin. To keep the numbering scheme consistent, let’s say that this work occurs during “Iteration -1.”<sup>6</sup>

**Just as agile developers implement requirements in priority order, agile portfolio managers should initiate projects in priority order.**

### Project Identification and Selection

Although Figure 1 shows the project identification and selection activity occurring before a project begins, which is true from the point of view of the project, this effort is actually an ongoing activity that takes place over the years. Ideas for potential projects will come from a variety of sources — senior managers who are reacting to opportunities and challenges within the marketplace, enterprise business modelers who recognize gaps in your organization’s ability to address your chosen market niche, enterprise architects who are responsible for your organization’s technical direction, existing project teams working on related systems, and even line staff.

The project identification and selection effort can and should be as agile as you can possibly make it. You should collaborate with stakeholders who are knowledgeable enough and motivated enough to consider the proposed project and invest in just enough effort to understand the potential opportunity of each project suggestion. When a project is first proposed, an initial investment will be made in:

- **Defining it at a high level.** This includes exploring how the new functionality will improve your organization’s presence in the market, how it will affect profitability, and how it will affect the people within your organization. Remember that not all projects will make the initial cut, so you only want to invest enough effort at this point to get a good “gut feel” for the business potential. Although traditional methodologies will recommend developing a vision document, I prefer Outside-In Development’s<sup>7</sup> focus on

identifying the potential stakeholders and their goals, which is key information to help identify the scope of the effort. At this point in time, strategic business ideas start to turn into tactical decisions, something that will continue into Iteration 0 and beyond.

- **Identifying a viable strategy for the project.** There are several issues to consider when identifying a potential strategy for the project. For example, should you build a new system or buy an existing package and modify it? If you decide to build, do you do so onshore or offshore? Will the work be solely done by your own development team, by a team from a system integrator (SI), or in partnership with the SI? What development paradigm — traditional/waterfall, iterative, or agile — will you follow? Will the team be colocated, near-located within the same geographic region, or far-located around the world? As you can see, there are many strategy combinations available to you. At this stage you may only be able to narrow the range of the possibilities, leaving the final decision to the project team if and when the project is funded.
- **Assessing its feasibility.** Many organizations choose to do just a little bit of feasibility analysis during Iteration -1, and then if they decide to fund the project, they will invest more effort during Iteration 0. In my experience, you need to consider four aspects when exploring feasibility: economic feasibility (Does it make sense from a business point of view?), technical feasibility (Are you capable of either building or buying this system?), operational feasibility (Can you run the system effectively?), and political feasibility (Can your organization tolerate the system and the changes that it will engender?).

Based on this initial identification work, an important part of which is modeling (which can and should be done in an agile manner)<sup>8</sup>, you will either discard the idea or prioritize it and put it in the appropriate place on your project backlog. The project backlog is a prioritized list of potential projects that could be taken on by your IT department, and your prioritization strategy will be determined by your organization’s willingness to take on certain types of risks and the potential opportunities provided by the various candidate projects. The potential projects available to you will almost always be far greater than your capacity to address them, hence the need for a prioritized list. You want to invest in the projects that will give you the best return, so just as agile developers implement requirements in priority order, agile portfolio managers should initiate projects in priority order.



## Monitoring and Governance

Initiating the “right” projects is just one aspect of portfolio management; you must also appropriately monitor and guide ongoing development projects as well as systems that are in production or being retired. During development, it is critical to monitor your organization’s standard metrics (more on this in a minute), as well as those issues that are specific to the individual project. These risks will have been identified during Iteration -1, but they will evolve over time as a project progresses.

Two aspects of agile software development — aspects not found in traditional/waterfall approaches — enable effective monitoring and management of the development projects within your IT portfolio:

- 1. Greater visibility.** The agile approach of producing working software on a regular basis gives stakeholders greater visibility into what a project team is actually doing. This informal visibility into development projects is far more valuable than the traditional documentation-based “earned-value management”<sup>9, 10</sup> because it’s concrete. Stakeholders don’t have to rely on the promises made in traditional status reports or in comprehensive requirements documentation; instead, they work with real software and determine if the team is actually delivering value.
- 2. Greater control.** Extensive involvement of the stakeholders — they control the budget, scope, and schedule on agile projects — enables them to direct the project teams effectively.

Disciplined agile teams will also have a metrics program in place to support good governance. Lean development governance (LDG)<sup>11</sup> includes practices — simple and relevant metrics, continuous project monitoring, and integrated lifecycle environment — that offer guidance for establishing a streamlined metrics program. Simple and relevant metrics are crucial to your success because you want consistent, reasonably accurate, and useful measures. Continuous project monitoring is crucial for properly steering your projects — or for cancelling them if they get into serious trouble. An integrated lifecycle environment is crucial because you want to automate your metrics as much as possible. The more you rely on manual metrics collection, the greater the cost and the less accurate the metrics will be. Worse yet, they may actually be outright lies, because people are trying to cover up.

In my experience, monitoring a software development project effectively requires three basic measures:

- 1. Financial.** This is obvious for any project regardless of development paradigm. Simply track the actual hours expended and then multiply by your fully burdened hourly rate.
- 2. Quality.** For agile projects, an obvious quality metric is defect trend tracking. If defects are rising over time, then the team isn’t addressing quality adequately.
- 3. Productivity.** Productivity metrics often stump many organizations, because they have a tendency to fall back on the traditional, bureaucratic ways of thinking and will attempt to adopt “consistent” measures, such as function points.<sup>12</sup> There are two significant problems with these sorts of measures. First, there are always fudge factors that are unique to the individual estimator — my function points are different from your function points — which implies that the metrics really aren’t consistent after all. Second, and most important, when you consider the total cost of ownership (TCO) for the estimate, including both the direct cost to do the count and the indirect costs associated with gathering the required information,<sup>13</sup> it becomes very difficult to justify these measures.

Luckily, at least for agile teams, there’s a better way. The critical observation is that the goal really isn’t to measure the level of productivity; it’s to determine whether a team is becoming more productive over time so that you can share with other teams what members have learned. A common metric that agile teams capture in order to identify the amount of work that they’ll commit to for an iteration is called velocity. A team’s velocity is the number of points of work that it can perform each iteration, calculated by observing the team’s track record from previous iterations. This is an existing and relatively painless metric that most agile teams collect, and when you look at the change in velocity over time — the team’s acceleration — you get an inexpensive measure of the change in productivity.<sup>14</sup>

Monitoring your projects, and then governing accordingly, is absolutely vital to your IT effectiveness. In 2007, *Dr. Dobb’s Journal* ran a survey that explored how people define success on IT projects and what level of success they believed they were experiencing.<sup>15</sup> The survey found that agile teams reported a success rate of 72% (compared with 63% for traditional projects), which means that almost 30% of agile projects are considered failures. The implication is that it behooves you to question periodically whether a project is still feasible in order to identify struggling projects and either turn them around or cancel them as quickly as possible.

Disciplined agile teams adopt the Unified Process's concept of go/no-go decisions at key milestone points during a development project,<sup>16, 17</sup> an important practice that enables good portfolio management. These milestone reviews don't need to be onerous, multiday bureaucratic efforts. They could be something as simple as a quick meeting where key stakeholders discuss how well the project is actually going.

**Many people like to focus solely on the predevelopment and development aspects of portfolio management, but that would be a mistake.**

Monitoring and governance of systems once they are in production (and of systems that are truly at the end of their lifecycle and are therefore being retired) is also an important part of portfolio management. Your operations and support staff are responsible for monitoring production systems, but metrics such as uptime statistics, newly identified defects, and support calls against a system are essential quality metrics that your portfolio managers will want to have.

Many people like to focus solely on the predevelopment and development aspects of portfolio management, but that would be a mistake. Because your IT portfolio includes existing systems, portfolio managers must have a good understanding of the systems within your organization's overall IT inventory — as well as those about to leave it. Retiring an existing system (e.g., removing your homegrown accounting system from production once a newly purchased package is safely deployed) can be a risky endeavor, and it also needs to be carefully managed.

### IT Inventory Management

Your enterprise architects should have a good understanding of your organization's existing production systems. That's the good news. The bad news is that few, if any, of these systems are perfect. One of the greatest challenges for agile teams — and for traditional teams, for that matter — is to determine viable strategies for leveraging these legacy assets. The technical strategies for working with and fixing legacy source code<sup>18</sup> and for refactoring production data sources<sup>19</sup> have been in place for several years. Although clearly a lot of hard work, these techniques are fairly straightforward

technically. What is often missing in most organizations is an effective approach to managing your existing inventory of systems.

A portfolio management strategy that I've seen implemented in several organizations is to classify IT assets, either systems or data sources, into the following categories:

- **Class A** assets are critical to your organization's long-term operation.
- **Class B** assets are important, but it isn't obvious that you're going to keep them around for a long period of time.
- **Class C** assets are not critical to your long-term strategy and are often slated for replacement if not outright elimination.

These categorizations are vitally important for agile teams because they help determine whether you should invest time in refactoring. For example, you would automatically invest in the quality of class A assets, so you will put regression tests in place and refactor those assets gradually over time as you work with them. At the other end of the spectrum are class C assets, which are not worth investing in except to address critical defects. Class B assets prove to be problematic in practice because you have to make case-by-case decisions as to whether you want to improve their quality whenever you enhance them. As a result, my recommendation is to minimize the number of assets that get categorized as Class B.

### PORTFOLIO MANAGEMENT AND OTHER ENTERPRISE DISCIPLINES

Portfolio management is only one of several enterprise disciplines your organization needs to address. The Enterprise Unified Process (EUP)<sup>20</sup> shows how various enterprise disciplines fit together and can be implemented in a relatively agile manner. For the sake of our discussion here, Figure 2 depicts how portfolio management activities are supported by enterprise business modeling, enterprise architecture modeling, and strategic reuse within an overall governance framework.

The following enterprise disciplines are critical to the success of your portfolio management efforts:

- **Enterprise architecture.** The enterprise architecture helps to identify whether potential assets fit into your organization's long-term vision, and it thereby affects your project prioritization efforts.

- **Enterprise business modeling.** Enterprise business modeling focuses on the business side of the enterprise. (Sometimes this is considered an aspect of enterprise architecture, but EUP separates the two disciplines.) When the portfolio manager is prioritizing projects, the enterprise business model provides vital business context.
- **Strategic reuse.** The reuse of assets across the portfolio is an important goal within most organizations; therefore, your portfolio manager will want to monitor relevant reuse metrics. The real issue is that there are potential cost savings due to reuse within a program, and you may want to schedule some projects before others in order to lay a foundation for future projects.
- **Governance.** Although many agilists don't like to focus on governance (and the same can be said for many traditionalists), the reality is that if you have one or more IT systems or projects, you have IT

governance. As Per Kroll and I have shown in our work on LDG,<sup>21</sup> it is not only possible to govern in a lean/agile manner; it is highly desirable to do so. As Figure 2 indicates, there is clearly overlap between governance and portfolio management, but they are still two separate efforts. Governance is an aspect of all IT disciplines, not just portfolio management.

Unfortunately, many agilists are leery about enterprise modeling due to the poor practices often exhibited by traditional enterprise modelers.<sup>22</sup> Many traditional enterprise modeling efforts run aground when the enterprise modelers focus just on modeling and not on executing the concepts captured in the models. If you choose, though, enterprise-level modeling can be performed in an agile manner.<sup>23</sup> The main strategies are to keep the documentation light, to work iteratively and collaboratively, and to work actively with the development teams to implement the systems called out in the enterprise models.

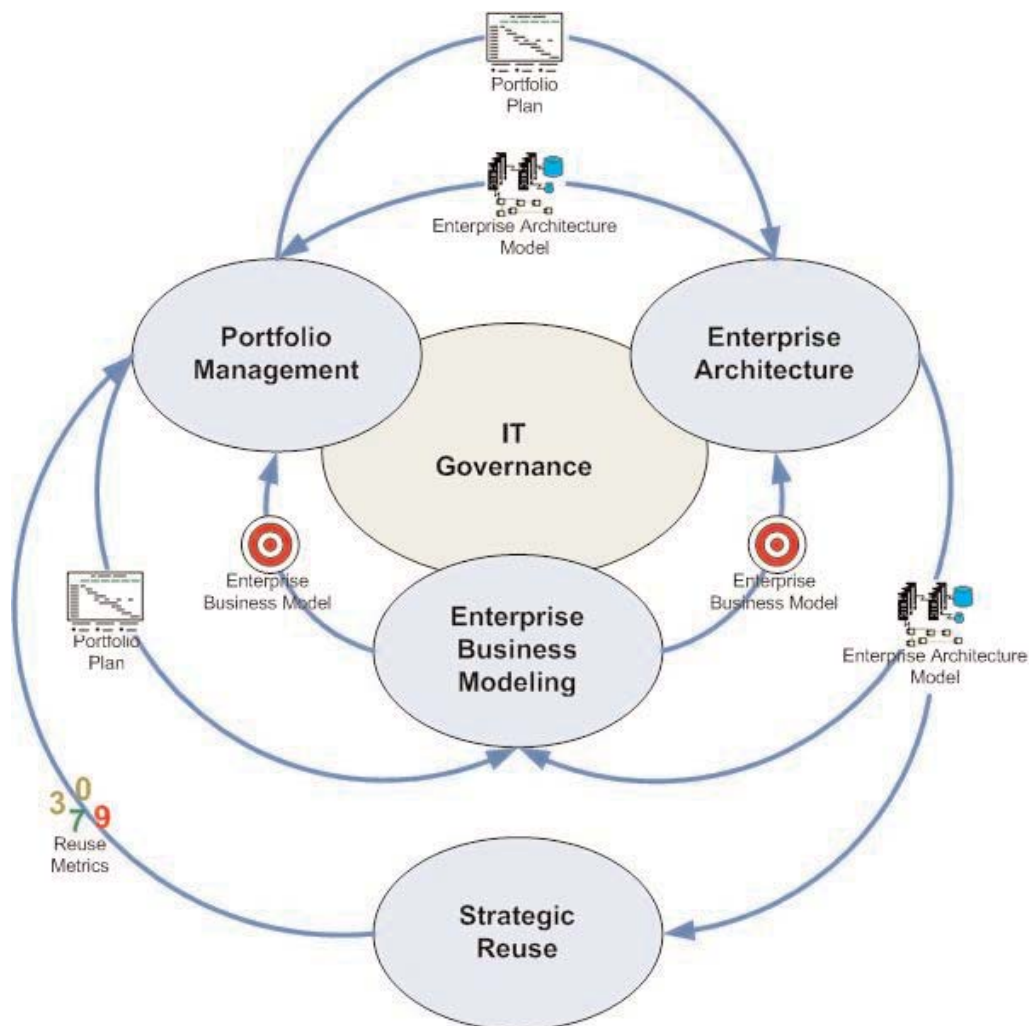


Figure 2 — Portfolio management and other enterprise disciplines.

## PARTING THOUGHTS

Portfolio management is an important IT discipline, one that can be adapted to address the needs of agile development teams. More important, your portfolio management efforts can be improved via the inclusion of more agile strategies. Effective portfolio management strategies reflect the needs of systems throughout their entire lifecycle and the realities of other enterprise disciplines such as enterprise architecture and IT governance.

## ENDNOTES

<sup>1</sup>Ross, Jeanne W., and Peter Weill. "Six IT Decisions Your IT People Shouldn't Make." *Harvard Business Review*, November 2002.

<sup>2</sup>"Manifesto for Agile Software Development," 2001 (<http://agilemanifesto.org>).

<sup>3</sup>Ambler, Scott W. "Examining the Agile Manifesto," 2003 ([www.ambysoft.com/essays/agileManifesto.html](http://www.ambysoft.com/essays/agileManifesto.html)).

<sup>4</sup>Ambler, Scott W. "Scaling Test-Driven Development." *Dr. Dobb's Journal*, 3 January 2008 ([www.ddj.com/architect/205207998](http://www.ddj.com/architect/205207998)).

<sup>5</sup>Ambler, Scott W. "Initiating an Agile Project." *Dr. Dobb's Journal*, 1 June 2006 ([www.ddj.com/architect/188700850](http://www.ddj.com/architect/188700850)).

<sup>6</sup>Ambler, Scott W. "Iteration Negative One." *Dr. Dobb's Journal*, 4 August 2008 ([www.ddj.com/architect/209902719](http://www.ddj.com/architect/209902719)).

<sup>7</sup>Kessler, Carl, and John Sweitzer. *Outside-In Software Development: A Practical Approach to Building Stakeholder-Based Products*. IBM Press, 2007.

<sup>8</sup>Ambler, Scott W. *Agile Modeling: Effective Practices for Extreme Programming and the Unified Process*. John Wiley & Sons, 2002.

<sup>9</sup>"Earned value management," Wikipedia ([en.wikipedia.org/wiki/Earned\\_value\\_management](http://en.wikipedia.org/wiki/Earned_value_management)).

<sup>10</sup>Ambler, Scott W. "Questioning Earned Value Management (EVM) on IT Projects." *Dr. Dobb's Agile Newsletter*, 22 May 2008 ([www.ddj.com/architect/207801786](http://www.ddj.com/architect/207801786)).

<sup>11</sup>Ambler, Scott W., and Per Kroll. "Lean Development Governance." IBM Rational White Paper, 2007.

<sup>12</sup>Garmus, David, and David Herron. *Function Point Analysis: Measurement Practices for Successful Software Projects*. Addison-Wesley, 2001.

<sup>13</sup>Ambler, Scott W. "Examining the 'Big Requirements Up Front' (BRUF) Approach," 2005 ([www.agilemodeling.com/essays/examiningBRUF.htm](http://www.agilemodeling.com/essays/examiningBRUF.htm)).

<sup>14</sup>Ambler, Scott W. "Acceleration: An Agile Productivity Measure." *IBM developerWorks*, 14 October 2008 ([www.ibm.com/developerworks/blogs/page/ambler?entry=metric\\_acceleration](http://www.ibm.com/developerworks/blogs/page/ambler?entry=metric_acceleration)).

<sup>15</sup>Ambler, Scott W. "Defining Success." *Dr. Dobb's Journal*, 31 October 2007 ([www.ddj.com/architect/202800777](http://www.ddj.com/architect/202800777)).

<sup>16</sup>Kruchten, Philippe. *The Rational Unified Process: An Introduction*. 3rd edition. Addison-Wesley, 2003.

<sup>17</sup>Kroll, Per, and Bruce MacIsaac. *Agility and Discipline Made Easy: Practices from OpenUP and RUP*. Addison-Wesley, 2006.

<sup>18</sup>Feathers, Michael. *Working Effectively with Legacy Code*. Addison-Wesley, 2004.

<sup>19</sup>Ambler, Scott W., and Pramod J. Sadalage. *Refactoring Databases: Evolutionary Database Design*. Addison-Wesley, 2006.

<sup>20</sup>Ambler, Scott W., John Nalbone, and Michael J. Vizdos. *The Enterprise Unified Process: Enhancing the Rational Unified Process*. Prentice Hall PTR, 2005.

<sup>21</sup>Ambler and Kroll. See 11.

<sup>22</sup>Ambler, Scott W. "Enterprise Modeling Anti-Patterns," 2006 ([www.agilemodeling.com/essays/enterpriseModelingAntiPatterns.htm](http://www.agilemodeling.com/essays/enterpriseModelingAntiPatterns.htm)).

<sup>23</sup>Ambler, Scott W. "Agile Enterprise Architecture," 2002 ([www.agiledata.org/essays/enterpriseArchitecture.html](http://www.agiledata.org/essays/enterpriseArchitecture.html)).

Scott W. Ambler is a regular contributor to Cutter Consortium's Agile Product & Project Management and Enterprise Architecture practices. Mr. Ambler is also the Practice Leader Agile Development at IBM Software Group. He travels the world helping clients understand and adopt the software processes that are right for them. Mr. Ambler is an award-winning author of several books, including those focused on the Unified Process, agile software development, database development, the Unified Modeling Language, and CMM-based development. He is a regular speaker at international IT conferences and is a senior contributing editor with Dr. Dobb's Journal. Prior to joining IBM, he led the development of several software processes, including the Agile Modeling (AM), Agile Data (AD), Enterprise Unified Process (EUP), and Agile Unified Process (AUP) methodologies. Mr. Ambler can be reached at [sambler@cutter.com](mailto:sambler@cutter.com).



# Selecting a Ranking Method for Your Project Portfolio

by Johanna Rothman

One of the most difficult parts of project portfolio management is deciding how to rank the projects — that is, determining which should be done now, later, and, most important, never. There are several ways to rank a project portfolio. Each is useful in specific situations and not so useful in others. But all share the same goal; namely, arriving at a single ranked list of projects.

Dave, a CIO, gathered his directors together. “OK, everyone, listen up. We have too much to do. Everyone’s multitasking, which means we’re not getting anything done. Our projects are still late. We need to decide what projects we’re going to do first, second, and third, and then stick with that, at least for a while.” His directors looked at him as if he had three heads.

“I’m serious. We’ll rank our projects and see which ones are most valuable, do those, finish them, and then go on to the next project.”

“How do you propose we do that?” asked Tanya, a director. “We don’t know what’s most important to you or to the rest of the senior management team. In fact, we keep getting different messages. First, my calendar integration project is first, then it’s the data integration project. Once last week, it was the performance project — but only for a day. I’m quite happy to rank the projects, but how long do we think that ranking will last, and how can we do it?”

Dave frowned. “Well, we’ve got a chance to make some decisions, show how we made them, and make them stick just for a few weeks. That will give us a chance to finish some work and rerank, if we have to.”

You probably share Dave’s problem. You have too many projects to do at the same time, not enough people, and not enough projects finishing. If that’s the case, it’s time to consider ranking the projects your organization has. In this article, we will explore a number of approaches to ranking projects:

- Assigning projects a point value
- Assessing the risk of a project
- Reviewing the project context
- Using double or single elimination
- Ranking a project according to your organization’s mission and values

So let’s get started. First, list all your active projects — that’s your project backlog. Now it’s time to start asking the difficult questions.

## ASK THIS QUESTION FIRST

Once you start ranking the projects, make sure you ask this question first:

Should this project be done at all?<sup>1</sup>

If there’s no sponsor, no customer, and especially no identifiable value for the project, stop it.

Don’t be afraid to not do projects at all, and especially *for now*. Project portfolio management is about making choices that guide the technical staff to finishing work on just the most important projects in the organization. If you can’t make a compelling argument for finishing this project, put the project back on the project backlog for now. Or add it to a parking lot,<sup>2</sup> where you list the projects that are not under active consideration but that you don’t want to forget about.

More likely, the answer to the “Should we do this project?” question will be “yes,” and you will need to proceed to ranking your various projects. Ranking requires discussion among the sponsor and the customer or customer surrogate, as well as some description of project value.

## RANKING WITH POINTS

Project value is the value of the project to the organization. If you could rank the projects with an ordinal ranking — as in, 1, 2, 3, and so on — your work would be done. But you often need more conversation than mere ordinal ranking affords. Points help you articulate the business value and provide a visible means of showing the relative business value of each project.

When you rank with points, you assign a number of points to a project, allocating more points to more valuable projects. Points represent business value. If you wanted, you could use dollars (or euros, or what have you), as in “How many dollars do we want to invest in

this project?" Sometimes, that's difficult to do if you aren't sure how long the project will take or don't know enough about the features. When you're evaluating the portfolio, you may not have thought through all those issues yet. Using points as a way to rank projects separates the project funding question from the business value decision.

Separating project sizing from duration helps project team members make better estimates. In the same way, separating business value from funding allows managers to see what output they desire from the organization. If all the projects have a small number of points relative to each other, then no one cares much about the set of projects.

When you rank with points, you take a large number of points, much larger than the total number of projects. For example, if you have eight projects, take 10,000 points to start. Now assign a unique number of points to each project. If you have two very important projects, you cannot assign each of them 5,000 points. You can assign 5,001 points to one project, and 4,999 points to the other project. Yes, both projects are very important, more important than anything else you have in your portfolio. However, allotting 5,001 points to the first project says, "This project is the most important one." The 4,999-point project is next in line.

Table 1 — A Ranked Listing of a Small Number of Projects

Project	Points
Project 1	3,000
Project 2	2,500
Project 3	1,250
Project 4	750
Total Points	7,500

Table 2 — Ranked Listing of Projects Where the Decision Makers Changed the Number of Points Used to Rank

Project	Original Points out of 10,000	Updated Points out of 20,000
Project 1	2,000	5,000
Project 2	1,900	4,000
Project 3	1,800	3,500
Project 4	1,700	3,400
Project 5	1,600	1,000
Project 6	1,000	500
Total Points	10,000	17,400

By assigning a unique number to each project, you show the organization which project is most important *for now*. If you allocate the same number of points to more than one project, the technical staff doesn't know which project is most important, and you don't have a project ranking. It's too easy for each person to make his or her own decision, which might be different from yours.

You might not use all your points. For example, in Table 1, one organization with four projects only used 7,500 of their possible 10,000 points. That said, if you find that *most* of the time you're not using all your points, you may need to reconsider the projects you're trying to rank. If you don't have several projects that provide substantial business value, you may not be considering enough or the right kinds of projects.

As you proceed through the project ranking, you might realize that you want a larger point spread to show the rest of the organization how the projects rank against each other. In that case, you may need more points to clarify the ranking. See Table 2 for an example of a team that started with 10,000 points and wanted to show the rest of the organization how important Projects 1, 2, 3, and 4 were. Projects 5 and 6 were much less important.

Points help you describe the relative business value of each project. If you need more points to describe the relative importance, use them. When you use more points, and have a larger difference between the most important and least important projects, project staff understand when to stay with one project.

When I introduce this approach to management teams, they ask, "What do the points mean?" I've had good results saying, "You have \$10,000 to assign to these projects. How do you want to assign the money?" There are times, though, when linking project value with dollars isn't sufficient. In such cases, the managers will try to assess the risk of not doing the project and assign this risk a dollar value. One way or another, you need to link points to project value.

Dave worked with his directors to use points to rank their projects. Their initial decision about their projects is shown in Table 3.

Table 3 — Dave's Ranked Listing of Projects

Calendar integration	5,000 points
Data integration	4,000 points
Performance project	1,000 points

But when Dave brought the ranking to his senior management team, they were concerned — wasn't the calendar project too risky to keep as the top priority? Senior management was concerned that if the entire IT group was stuck on the calendar project, nothing would get done. Dave and his team needed to address risk

## RANK WITH RISK

I bet you've heard things like, "Don't do the risky projects — do the sure things." Well, that's great, but how do you know what a sure thing is? How can you tell what the return on a project will be?

You can't tell for sure, but you can learn a little about the risk and value of a given project if you do just one iteration's worth of work on it. When you work on projects, feature by feature, inside a timebox, finishing features and showing demos as you proceed, you can learn more about the risk and the potential return. Once you've completed one iteration's worth of work and received feedback on the demo from a customer or customer surrogate, you might have enough information to evaluate the project's risk. Now you can ask the customer about the value of this project. Is the project high value? What would make it high value? The value discussion is how you can identify potential return. It's still a guess, but it's a more educated guess.

After due consideration of the risks of the calendar integration project, Dave and the rest of the IT directors decided to do just one two-week iteration on the project and see where they were after those two weeks.

Ranking with risk is not without problems. Sometimes you need more than one timebox to know about the relative risk of the project. Sometimes the customer insists the project is high value, without considering the other projects in the portfolio. In that case, you need to consult other people, such as all the project sponsors or all the customers, and explain that you can only do one of these projects. Under *these* circumstances, which project has the most value?

One aspect of assessing the project risk and return is to determine *for how long* a particular project is risky or has a potential return. A number of years ago, an organization wanted to put their client information in a secure

area of their Web site. Security was a new field, and they weren't sure they knew everything they needed to do to keep the data confidential. Once a quarter, they tried to implement a feature in two weeks and test its security. As long as the other developers and testers could break the feature, the CIO decided the project was too risky. But about 18 months later, no one could break the features. Now the CIO moved the project up in the ranking because it was not too risky and had a much higher return.

Figure 1 shows how that organization reviewed the risks and the value.

When they reviewed the risks, they knew that if they didn't keep the information secure, they would be in trouble with their clients. They had few risks if they didn't do the project for a while. If the organization could implement the proper security, they would have a jump on their competition, but it would still be a selling job to their clients. Yet once the clients realized what they could do with their online information, the organization was sure they could change the way they worked with their clients.

## RANK WITH CONTEXT

At times, it can be quite difficult to rank with points. It's especially difficult if you have different types of projects, which is quite common for IT departments. You might have "internal" projects to keep departments and systems up to date, as well as "external" projects that affect how you deliver products and services to your customers. If that's the case, you will need to consider the context of each project when doing your ranking.

One IT group first organized their projects and then ranked them as internal or external (see Table 4). In Table 4, you can see that the HR résumé system offers the highest value of the internal projects, but two of the other projects are not far behind. And in Table 5, all of the projects are ranked higher than the internal projects.

This IT department had enough teams to fully staff two of the external projects and one of the internal projects. But separating the projects by type and ranking them helped the CIO explain to the different constituencies in

Project	What kind of risk do we have if we do it?	What kind of risk do we have if we don't do it?	What kind of value do we receive from this project?	What kind of value do our customers see from this project?
---------	---	---	---	--

Figure 1 — Rating a project according to risk and value.

Table 4 — Internal Projects, Ranked with Points

Internal Projects	Points
HR résumé system	3,000
Supply chain updates	2,800
Year-end updated reports	2,700
New reports for finance	2,000

Table 5 — External Projects, Ranked with Points

External Projects	Points
Shopping cart additions	5,000
Security for specific scenario	4,500
Loyalty program, phase 1	3,500

the business why the IT group could not get to their project now. Once the senior management team saw the ranking, they could discuss the relative ranking and explain why different projects were ranked the way they were.

### PAIRWISE COMPARISON AND DOUBLE ELIMINATION

There may be times when you need to look at every project in comparison to every other project; in other words, you need to do a pairwise comparison. An easy way to do this is to write the name of every project on a sticky note, and place them all horizontally on a whiteboard or wall. Make sure you have all the necessary people in the room when you make the decisions. Once you've asked the "Should we do this project at all?" question, hold up two sticky notes and ask, "Which project is first, this one or that one?"

As you decide which project is of higher rank, put the stickies on the wall in vertical order, with the highest-ranking sticky on top. Now, take another sticky from your horizontal collection. Take the top-ranked sticky and the new one and ask, "Which one is first, this or that?" Put the top-most sticky note on the wall, pick up the next one, and ask the same question. Repeat this process until all the horizontal stickies are in vertical order. You now have your project ranking.

Double elimination is a form of pairwise comparison. Many tennis tournaments are pairwise comparisons, because the winners all play losers. Figure 2 is a picture of double elimination.

Because the winners of each round eventually are compared to the losers of each round, you have the opportunity to compare each project against every other project.

Note that while Project 1 initially beat Project 2 in the first round, Project 2 was not eliminated. As the team discussed their reasons for each project's ranking, Project 2 won against all the others and eventually prevailed.

### SINGLE ELIMINATION

Single elimination is a slightly easier approach to project ranking, as it does not compare each project against each other project (see Figure 3).

### USE YOUR MISSION AND VALUES

You might find every ranking approach difficult to implement in your organization. Sometimes, that's because you have not articulated your mission or have not defined the values of the organization. If either of those things is true, discussions about the ranking of each project might lead you in circles. In those circumstances, you will need to define the mission and values explicitly.

Once you know the organization's mission, it's possible to use it to say, "Yes, this project is part of our mission" or "No, this project is not part of our mission." If you find that you have sacred cow projects, use your organization-defined values to describe why the projects are sacred cows and should be eliminated.

As Dave was working through the projects, he discovered that several people in Tanya's group were working on a data consolidation project instead of the calendar integration project. He asked Tanya what that project was. "It's a way to reduce the number of databases and disk drives in the organization," she replied.

Dave thought for a minute, and asked, "When did this project start?"

"As part of our green initiative," Tanya replied.

Dave considered this and said, "OK; we have a problem. One of our corporate values is to reduce waste. But this project is a lower priority, because our green value is a lower priority than enabling the business to make money."

Tanya protested, "But when we save the company money, we make it easier for the money we make to go further."

Dave explained, "Yes, you're right. However, *right now* it's more important to make money than to save money. That's just for right now. I'm not saying we shouldn't do this project at all. I am saying that for right now, we should not staff this project. That decision won't last forever, but for these next few iterations put the data consolidation project on the portfolio backlog, and we'll reevaluate it the next time we review the whole portfolio."



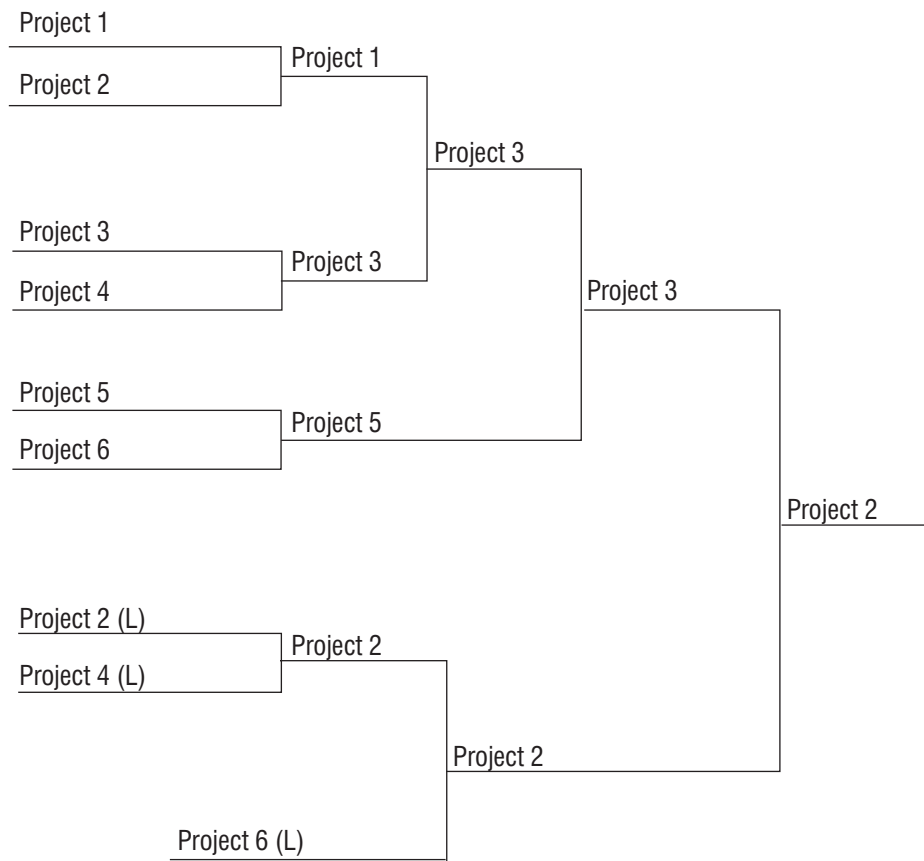


Figure 2 — Double elimination is a type of pairwise comparison.

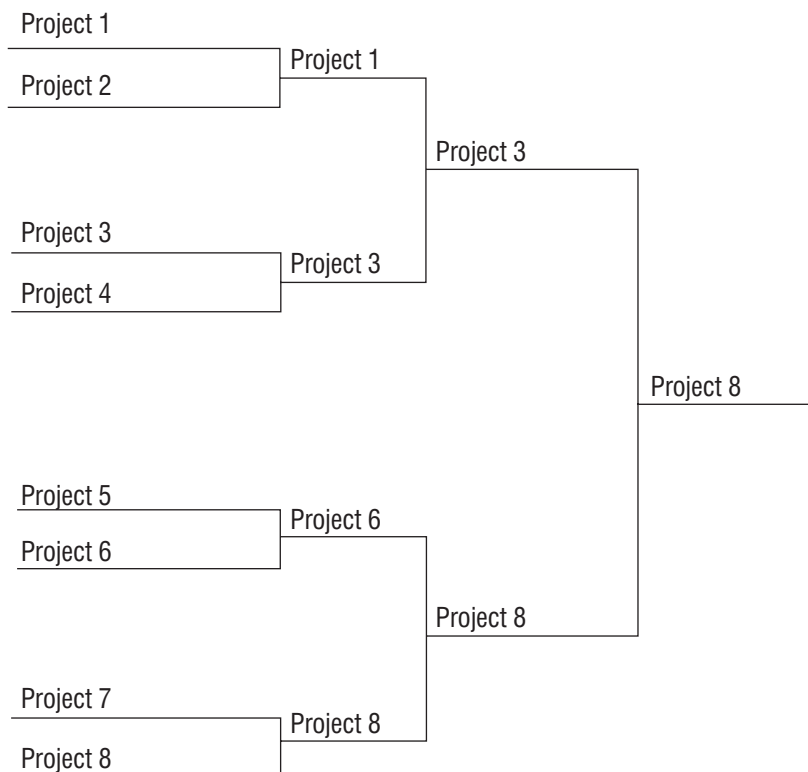


Figure 3 — The single-elimination method.

If your organization does not have a mission, or if you have not defined and articulated your organization's values, your efforts to rank the portfolio will make the lack of mission or value articulation visible to you. If you *have* defined a mission and defined your corporate values, it will be easier to make the portfolio decisions. It still may not be easy, but the decisions will be clearer.

## RANKING ISN'T FOREVER

Once you rank the portfolio, you have only to wait for one iteration to finish until you can rerank the portfolio. Having that flexibility, it's easier to make the portfolio decisions, because you know the decision doesn't have to last any longer than one iteration. Say your organization uses four-week iterations. If you need to make decisions faster, you can ask every project to move to two-week iterations. Now you have an opportunity every two weeks to reevaluate the portfolio and determine whether you want a team working on a particular project or not.

## REMEMBER TO KILL PROJECTS

Not every project deserves to stay on your backlog. Sometimes products have had several releases (projects), and you've put another release on the backlog. Eventually, you move it to the parking lot, because other projects are more valuable.

Yet while it's helpful to move projects out of the way, projects that have completed some number of iterations but are not providing value should be cancelled outright. Sure, you can take the "leave-our-options-open" route and move a project to the backlog or the parking lot. But if you have a number of higher-value projects and this one looks as if it's never going to deliver that value, kill the project. You can always transform it in some other way if you want to start it up again.

## CONCLUSION

Every leader and manager in the organization who has responsibility for staffing projects needs to rank the

projects in his or her portfolio. You might be able to staff all those projects, but more likely, you will need to make some decisions and then reevaluate the portfolio periodically. Use your organization's mission and values to help guide your decisions. The easiest way to make portfolio decisions with sufficient frequency is to work in relatively short timeboxes, implementing by feature and finishing work in that timebox. Now you have a product you can see and can reduce your portfolio decision risks.

## ACKNOWLEDGMENTS

I thank Sanjiv Augustine, Esther Derby, and Don Gray for their review of this article.

## ENDNOTES

<sup>1</sup>Rothman, Johanna. *Manage Your Project Portfolio: Increase Your Capacity and Finish Your Projects*. Pragmatic Bookshelf, forthcoming 2009.

<sup>2</sup>Rothman. See 1.

*Johanna Rothman is a Senior Consultant with Cutter Consortium's Agile Product & Project Management practice and a contributor to that advisory service. Ms. Rothman helps leaders and managers solve problems and seize opportunities, working with clients to find leverage points that will increase the organization's effectiveness. Ms. Rothman has been President of Rothman Consulting Group, Inc., since 1994. She focuses on project, risk, and people management. A frequent speaker and author on managing high-technology product development, Ms. Rothman has written articles for Cutter IT Journal, Software Development, Better Software, IEEE Software, Crosstalk, and IEEE Computer. She also publishes the Pragmatic Manager, a monthly e-mail newsletter about managing product development. She is the author of the 2008 Jolt Productivity Award-winning Manage It! Your Guide to Modern, Pragmatic Project Management; Hiring the Best Knowledge Workers, Techies & Nerds: The Secrets & Science of Hiring Technical People; and the forthcoming Manage Your Project Portfolio: Increase Your Capacity and Finish More Projects. She is the coauthor (with Esther Derby) of Behind Closed Doors: Secrets of Great Management. Ms. Rothman also leads workshops in the areas of project management, hiring and interviewing skills, management, and leadership. She can be reached at jrothman@cutter.com.*



# Agile Planning Over the Long Term: The Portfolio Planning Game at TWeb

by Jens Coldewey

There are moments in a consultant's life when you know that you are in an all-or-nothing situation. Either the client goes with your ideas, or your engagement is done. I knew that one of these moments had arrived when I looked into the faces of the executives in the middle of the first portfolio planning game I did at a company I'll call TWeb. We had just collected some numbers and had done our first pass through the backlog when I concluded that the only way to fulfill any of the promises made to TWeb's customers would be ... to stop acquiring new customers.

This caused the director of sales to explode. "I don't accept this planning style anymore," he snarled as he turned to confront me. "Before you came in, we could sell new contracts without any problem. Your job was to make development more efficient, for they never kept any deadline! And now you tell us that I should stop doing my job!"

"Well, I can't see anything wrong with these figures," the CEO answered. "What they tell us is that we're in deep trouble here, and we already have sold much more than we can deliver."

"Our business relies on new customers," the sales director insisted, "and stopping sales would mean to ruin our business!"

The CEO replied, "It seems we have to work on our business model then."

This portfolio planning game took place in month nine of my engagement at TWeb, a 100-person company that sold a highly sophisticated product to manage tourists' travels. It was a major turning point both in the strategy of TWeb and in their conversion to agile methods. We had introduced standard agile planning techniques eight months earlier and now had a basis that was stable enough to dare a forecast into the next two years. The forecast was devastating, and the fact that the CEO drew the right conclusions from it saved both his company and about 100 jobs. This is the story of the forecast and what happened during the next year. But let me start at the beginning ...

## TRAVELING IN THE FOG

When we began our engagement at TWeb, we had a clear mission: introduce XP. Management had the impression that the programmers were too slow and that XP would help them to increase productivity. It took us only one or two weeks to find out that there was much more wrong at TWeb than just the productivity of the programmers.

The company had about 30 clients using their system. For each client there was an account manager who spent most of his or her time at the client's site, helping them as second-level support and gathering requirements. Whether and when a new requirement "made it" into development was by and large determined by the connections the account manager had with the development team. The developers were under extreme pressure and spent about 75% of their time fixing bugs — bugs that were discovered by the clients in their production systems. There was no QA in place and no testing. Some clients had already called their lawyers in, while others had announced publicly that they wanted to get rid of TWeb. The situation was desperate.

A year before, TWeb had hired five new programmers, doubling their development staff. Unfortunately, that had made things even worse.

There was a backlog of more than 300 open bugs, each of them with at least one client waiting for its resolution — some clients had been waiting for more than three years by that point. No planning took place; nobody was able to tell how severe the situation really was. "The developers are too slow" was the common diagnosis we heard all through the hallways, and even the developers seemed to believe this. Measured on a CMM® scale, this company was definitely on Level 1 — chaos — for the single reason that CMM does not define a Level 0.

## FIRST STEPS: INTRODUCING BASIC AGILE PLANNING

Raising the productivity of a programming crew takes time, especially if they have to work with about one million lines of badly designed legacy code. It was quite clear that if this was the only step the company took, they would have not survived long enough to reap the fruits of a more efficient development process. Furthermore, the only tangible result would have been that the account managers would have heaped more requirements onto the development team. Both the developers and the account managers were acting as lone warriors; there was virtually no team spirit.

We had to introduce some more fundamental changes into the organization of this company, changes that would offer quick relief and move the organization into an operating mode in which they could act strategically again. We also had to make this organization act as a team rather than a group of lonesome cowboys.

Our starting situation was a clear case of development congestion enforced by bugs due to pressure and weak testing practices (see Figure 1). In a situation like this, there are three primary measures you can take to unburden development and return it to a controllable situation:

- Reduce the number of requirements that are put into development.
- Increase the quality of the requirements.
- Reduce the number of bugs made during development.

We addressed the second and the third issues with such practices as up-front acceptance tests, automated testing, pair programming, and code reviews. Although these steps were important, they are beyond the scope of this article. I'm also going to forgo discussing the accompanying measures to strengthen the team spirit, which were at least as important. My focus here is on the planning process.

We had to raise the consciousness that development is a limited resource — very limited if you have such a backlog of bad practices and code — and that you get better results by using this limited resource wisely rather than complaining about its limits. To help the account managers make wiser decisions, we started introducing a standard agile planning process.<sup>1</sup>

## MULTILEVEL PLANNING AND RED CARDS

To accommodate both the short-term demands of the (buggy) software in production and the medium-term demands of the clients, we decided to use a three-level planning process:

1. In a weekly planning meeting, the tasks for the next week were estimated and scheduled. This level made sure the team could react to urgent demands within two weeks at most, without disrupting the planning process. In addition, it allowed the team to adapt to changes in the higher-level plans — crucial in this chaotic situation.

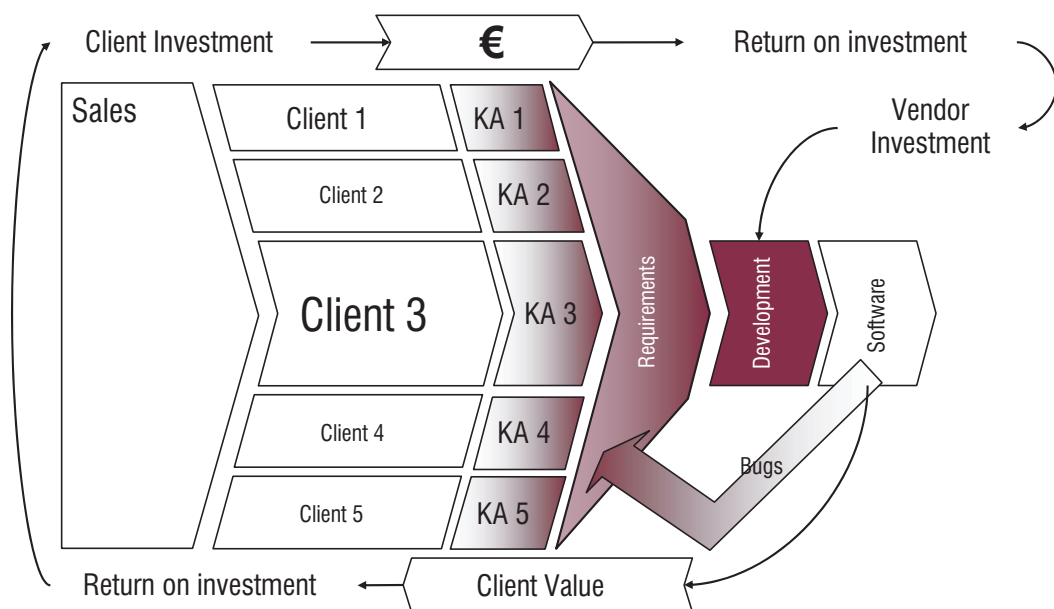


Figure 1 — How development congestion blocked the value chain of TWeb and their clients.

2. In a monthly increment planning meeting, the demands of the different clients were balanced and discussed. The priorities were readjusted, and upcoming events — such as major version changes at client sites — were scheduled.
3. Finally, a four-month release plan mirrored the release rhythm of the product.

After each release, a retrospective was scheduled. In addition, the planning process was adapted during the first monthly meetings.

Most of the time, these three planning intervals work well. In some cases, however, a two-week reaction time would be too slow. If operations at a client's site stop, for example, it has to be the number-one priority to get the site up and running again, no matter what the original plan says. To cover urgent tasks like these, we introduced "red cards," items that could be "thrown" on top of the backlog at any time to be solved, ideally, within the same day.

After the first release, most account managers were shocked to realize that only about 20% of the effort during that release went into items planned during release planning, and only about 50% of the weekly items actually came from the monthly plan. Instead, impulsively acting for the sake of acting had so much become part of the company culture that even weekly stability seemed to be painful for them. The first few weeks surprised even me. At the end of each week, nearly one-quarter of the finished cards were red, indicating that they had entered development during that week, circumventing any planning process. The key account managers had used the "emergency line" to tweak the overall planning process and channel the pressure they experienced from their clients directly onto the development team, instead of trying to alleviate it through the planning process.

Some of the account managers honored the repeated pleas to advocate the interests of the company instead of just the specific interests of their clients. This reduced the pressure on development a little bit — until other key account managers exploited this opening to pump in even more of "their" items. Inconveniences in the user interface showed up as emergency tasks, and some items raised on red cards were not deployed immediately but saved for the next release — a clear indication that these items were not urgent at all. Rather, red cards were being used to cheat the planning process and the other account managers.

It was obvious that the account managers were not able to self-organize as a team. This was partly because of the immense pressure they experienced from their clients and partly because the company's culture had promoted an "every man for himself" attitude in the past. The team had to learn that the CEO was serious about this change in culture.

To get the situation under control again, we increased the cost of red cards. Every red card had to be approved by the CEO before it could be thrown on the team. This instantly reduced the number of cards to about two per month, showing both that this instrument was necessary and that most of the red cards thrown so far were just a misuse of an emergency line. The red cards gave us enough transparency to recognize the situation and act accordingly.

**Impulsively acting for the sake of acting had so much become part of the company culture that even weekly stability seemed to be painful for them.**

In retrospect, the red cards were crucial for success during this first phase of adopting agile planning. Rather than having a completely new structure imposed on them, the key account managers were able to keep their current working style, but they now had to do it transparently and face the effects. The new planning approach gave the lie to their standard excuse that "the developers are too slow," showing instead that the true source of the problem was their habit of undermining any controlled way of managing the workload. The red cards made this obvious and delivered the healing shock they needed to embark on something new. Deprived of their major tweaking tool, they now had to come together and collaborate.

Almost immediately after the near-banning of the red cards, the weekly plans started to stabilize and provide a sensible forecast for the next week. It took TWeb another two months to stabilize the monthly planning, too, giving them a chance to provide their clients with reliable delivery information. The clients were still not happy with what the company could promise, but they quickly learned that at least these promises were not empty anymore. "We see that things are getting better and decided to put you on parole," one client told the CEO.

## THE FOUNDATION: ABSTRACT ESTIMATIONS

However, there was still a long way to go. Enervated by endless discussions as to why the 10 developers had finished only “25 person-days” in one week although they had worked 50 person-days, we decided to switch to abstract estimations. Items were estimated only in points, not in person-days anymore. We quickly figured out that the team could do about 20 points per week, 50 points per month, and 120 points per release.<sup>2</sup> Having accepted that a “task point” in the weekly task planning was something different than a “story point” in the monthly plan, discussions of the “What did you do in the rest of your time?!?” variety ceased, and the account managers started to concentrate on their real job: prioritizing the tasks in the backlog and optimizing the value generated during development.

**When the developers finished their estimations, there was enough work for the next *four and a half years* — and that’s if no new requirements came up!**

After half a year, the team managed to make quite reliable forecasts for the release to come. Thanks to these more accurate estimates and accompanying technical changes (e.g., automated acceptance testing), most clients started to regain confidence in TWeb, which led to another decrease in pressure. The company had shifted from complete chaos into a state that allowed them to plan and control their activities, at least in the short term. The fog had cleared to some extent, and though they were not pleased with what they saw, both the executives and the account managers learned to appreciate the new level of transparency.

## ADDING THE LONG-TERM PERSPECTIVE

The new transparency did not come free. To reduce some of the pressure on development, the account managers had decided to concentrate on bug fixing first. They had stopped nearly every new feature to reduce the enormous backlog of unfixed bugs. Their clients followed them, not only because they profited from the fixed problems, but also because the account managers had promised them that this was just a passing phase. However, the leeway the clients were willing to give was limited. They wanted to know when they would be able to deploy the new features they were waiting for; some of them had been waiting for more than three years.

The product’s backlog exceeded the development team’s current release velocity by far, and nobody had the full picture of that backlog anyhow. “We need some technique to get a long-term picture of the situation,” the CEO told me. “Our business model relies on new features the clients pay for, and right now we’re only doing bug fixing. I have strategic plans with this product, but I have no idea when to kick them off without running into the same old chaos again.” It was time to think about long-term planning — it was time to start the portfolio planning game.

The first step was to collect a full backlog. Until then, the account managers had kept their own backlogs, which were way too detailed to be of any use over the long term. Instead, we called for big issues only, three to 10 topics for each major client and a bag of another 10 to 20 for general support, the group that cared for the smaller clients. Finally we asked sales to contribute their requests for features from their marketing perspective.

We ended up with 76 new features, 30 of which were already contracted or at least promised. The moment of truth came when the development team started to estimate the features. Since they were all of about the same granularity as those scheduled during the release planning, we already had some experience with estimating them. Our latest statistics told us that the release velocity had risen to about 150 feature points per release by that time, with three releases per year. When the developers finished their estimations, the 76 features summed up to 2,024 points, or enough work for the next *four and a half years* — and that’s if no new requirements came up! What was even more frightening was the 1,639 feature points the developers had estimated for the 30 items that were already promised or under contract for the next year; this was enough work for the next three years.

It seemed pretty clear that there was no capacity left for any strategic ideas, not to mention new clients. However, the team did not give up that fast. “Well, you just took this number of 150 in your calculations,” the CEO objected, “but I don’t think that’s realistic. We’re going to hire additional developers, which raises our long-term velocity. And this Extreme Programming stuff will hopefully also raise our productivity. So maybe there’s still hope we can manage this!”

## SCENARIO ANALYSIS IN A NUTSHELL

The problem here was obvious. On the one hand, the CEO and the account managers needed a realistic idea of what was possible to start negotiations with the customers. On the other hand, we were running into a

debate of how the velocity might change in the future, with a severe risk of wishful thinking taking over. The problem is similar to any other long-term forecast, such as predictions of global warming or social changes in a neighborhood. Therefore, I suggested using a scenario-based approach, similar to the techniques used for climate predictions.<sup>3</sup> Instead of forecasting a single future, a scenario-based approach forecasts a range of possible futures, depending on different possible developments among the controlling parameters (see Figure 2).

To avoid getting stuck in thousands of different scenarios, a scenario analysis focuses on only a handful of different settings, depending on the subject you want to explore. The absolute minimum consists of three scenarios:

- **A worst-case scenario** assumes that all parameters will develop in the worst imaginable way. For example, in the case of global warming, this scenario usually predicts a steady increase in greenhouse gases along the same lines as over the last 50 years.
- **A best-case scenario** assumes that all parameters will develop in the best possible way that still seems to be realistic. In the global warming discussion, this scenario predicts that all governments will take effective measures to reduce their climate load in the next few years.
- **A realistic scenario** lies somewhere between these two and reflects likely developments. In some cases, as with global warming, the realistic scenario is quite similar to the worst-case one.

## COMBINING SCENARIO ANALYSIS AND AGILE PLANNING

A standard scenario analysis can be a very cumbersome undertaking, because it involves identifying the key parameters and building a model showing how they influence the system. Fortunately this was pretty simple in our situation, as the single parameter we had to vary was release velocity. Reflecting the feeling in the team, we defined the current 150 feature points per release as our worst-case scenario. I don't recommend this approach, because it neglects the not too unlikely prospect that things could deteriorate even further, but it was agreed that it would not make much difference to TWeb's future whether the current velocity would continue or get worse. So we started with three scenarios: the 150-point scenario for the pessimists, such as me; a doubled velocity for the optimists, such as the lead developer; and a tripled velocity to beat even the most unrealistic optimist. After some discussion, we also added a fourth scenario of a 50% increase in velocity and agreed that this might be the most realistic one (see Figure 3).

Though this approach was fairly simplistic, it led to the discussion recounted at the beginning of this article and to the decision that TWeb would acquire no new clients who couldn't use the product pretty much out of the box. This decision effectively banned any new acquisition of major clients, because most larger organizations needed to customize the system to integrate it into their IT landscape.

Two months later, the director of sales quit his job.

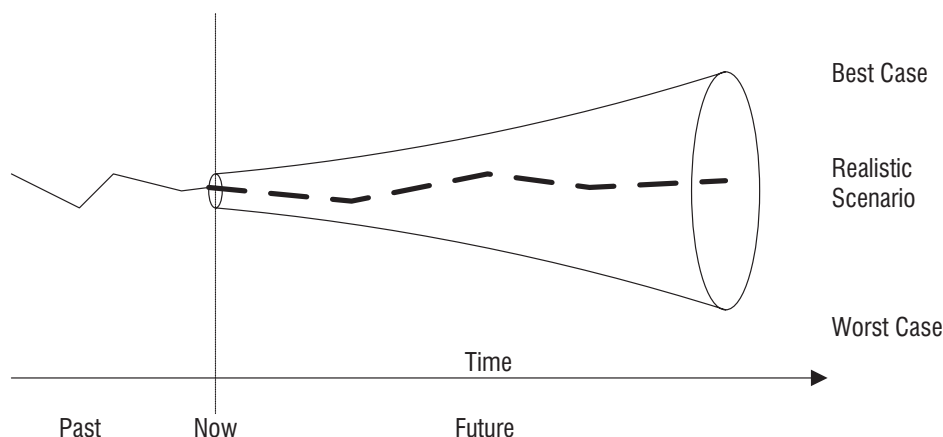


Figure 2 — In scenario analysis, you don't try to predict a single future, but a set of possible futures, symbolized as a funnel here. The best-case and worst-case scenarios form the boundaries of the analysis, while one or more realistic scenarios show the possible consequences of defined actions.

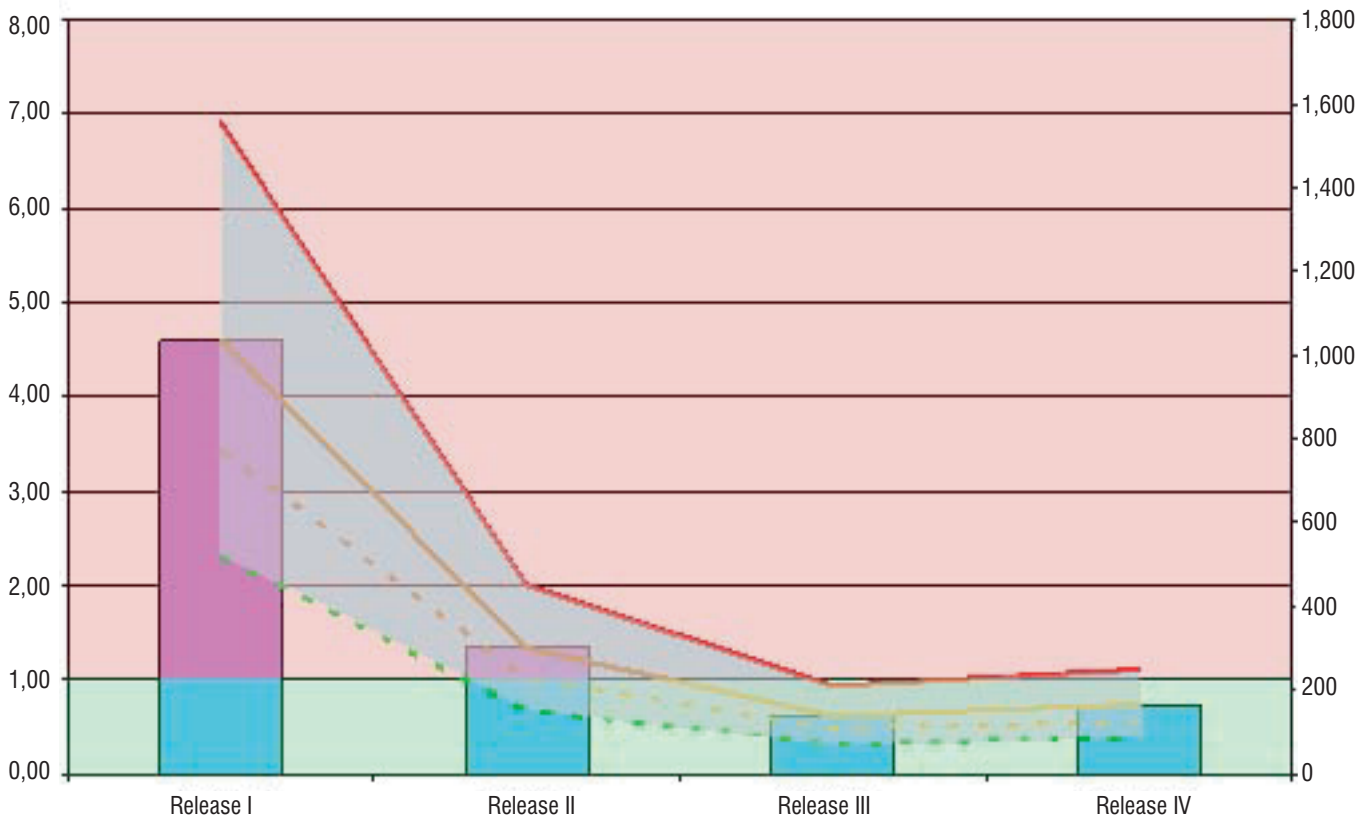


Figure 3 — The backlog of legally promised features for the next four releases (bars) and four different scenarios for the development of the velocity and the resulting team utilization (i.e., the “number of teams” of the current size you would need to do the job) (lines). A healthy utilization would be below 1 and decrease as you look further into the future. The graph shows that the promised goals for the next two releases were not feasible by any means and that without significant cuts in scope, no relief was in sight. Many promised features had to be shifted so far into the future that the clients would probably cancel their contracts.

## MEASURES TAKEN

After some discussion, the CEO sent the roadmap team back to their work with a new assignment to be done in one week: suggest how to cut the task load to 150 points for the next release and 250 for the following releases based on the developers’ estimations. “I will negotiate any sensible suggestion with our clients,” he declared. During this week, he cancelled the contracts of two of the clients who had the longest backlog of unfulfilled requirements, following the idea that it is better to lose a client who is upset anyhow than fight for them and upset the others. Two of the account managers sounded their clients out on deferring major production start dates by one year and at the same time reducing the scope. Finally, all participants came back with suggestions on how to slice features differently to make them cheaper.

The next week the group reconvened — without the director of sales, who had an “important” customer meeting — and agreed on a roadmap for the next five releases based on a velocity of 250 feature points per

release. This roadmap had cost two clients and further damaged TWeb’s reputation with several other clients, but for the first time in the company’s history they had a forecast for the next 20 months that everyone had agreed was realistic. (Well, to be precise, everyone had agreed but me.)

Of course, seen from a scenario analysis perspective, this was only half of the job. To conduct a full analysis, the team would have had to finish all four different scenarios and assign their features to the releases with a decreasing velocity for each release to reflect unexpected changes. However, the team decided that this would be an academic exercise that would not contribute to their actions anymore. They had seen enough, and they had reacted.

## 15 MONTHS LATER

It turned out that a velocity of 250 feature points per release *was* realistic due to additional staffing and increasing utilization of test-driven approaches. One of the two



clients whose initial launch had to be deferred was able to successfully launch the system 10 months after the planning game meeting — it was the first nearly “smooth sailing” launch TWeb had ever had. The second deferred client is just about to launch as I write this article, 15 months after the big meeting. This launch is proving much more problematic, because the account manager in question resisted cutting back the requirements to the extent necessary. Instead, the team reverted to their old habit of cutting expenses by skimping on testing and code quality. Though the first results of the launch look better than anyone expected — mainly due to a lower-than-predicted load — the team will still have to spend months cleaning up the mess. As with most practices, agile planning only works when you do it.

## CONCLUSION

Standard agile planning does a great job of predicting the short- to medium-term perspective. In the long term, though, a single plan is not able to represent any meaningful information to an agile organization. There are simply too many uncertainties connected to changes both in the team and in the clients’ behavior. Different hopes and fears as to how these parameters will change make it difficult, if not impossible, to get the team to agree on a realistic view.

Combining agile planning techniques with scenario analysis offers a strategy for including all perspectives within the team into a single model for discussion, thus providing a basis for sound decisions. Often, a

very simplistic adoption of scenario analysis will be sufficient: you have to vary only the team velocity to outline pessimistic, optimistic, and realistic scenarios. As with many techniques, the major gain comes from the discussion that occurs while setting up the plan, not necessarily from the completed plan itself.

There’s no doubt that the portfolio planning meeting at TWeb was tense and stressful. But in the end, that meeting probably saved the life of the company and about 100 jobs.

## ENDNOTES

<sup>1</sup>Cohn, Mike. *Agile Estimating and Planning*. Prentice Hall, 2006.

<sup>2</sup>The fact that the velocity does not rise linearly with the time mirrors the increasing uncertainty of the estimation process the larger the tasks become and the further they are in the future. In addition, tasks that are not planned in the release may still enter the monthly plan, and tasks not planned in the monthly plan may still make it into the weekly plan. So the different velocities reflect the need for agility in the different scopes.

<sup>3</sup>Schwartz, Peter. *The Art of the Long View: Planning for the Future in an Uncertain World*. Currency Doubleday, 1996.

*Jens Coldewey, based in Munich, Germany, is a Senior Consultant with Cutter Consortium’s Agile Product & Project Management practice. Mr. Coldewey specializes in deploying agile development in large organizations. For more than 10 years, he has worked with numerous large and medium-sized companies to introduce agile practices and agile management. Mr. Coldewey is a founding member of the non-profit Agile Alliance and served as a member of its board. He is also a regular speaker at agile and OO conferences. Mr. Coldewey can be reached at [jcoldewey@cutter.com](mailto:jcoldewey@cutter.com).*



DON'T CALL US, WE'LL CALL YOU

# Lean Portfolio Management at SciQuest

by Jamie Duke and Sam Bayer

## IMPROVING PROCESSES THE LEAN WAY

The first step in managing a portfolio of projects, lean or not, is to *have* a portfolio of projects.

When you're busy with the daily trappings of running a customer support organization for a \$37 million software business, it's easy to lose sight of the fact that responding to customer calls isn't your only job. In a lean organization, energy needs to be devoted to improving the processes by which you respond to the customer, or better yet, obviating the need for them to get in touch with you in the first place.

This case study will discuss how we in SciQuest's customer support organization juggled our responsibilities for providing timely resolution to customer incidents with the creation and management of a portfolio of process improvement projects.

First, let's set the context for our story by giving a little background information about SciQuest. We're a Cary, North Carolina-based software as a service (SaaS) company that was launched during the heyday of the dot-com era and subsequently relaunched in 2001. SciQuest provides purchasing automation and supplier enablement software and service targeted at vertical markets (higher education, life sciences, healthcare, and government). We have 165 employees and are growing at over 30% per year.

Two years ago, we embraced a set of lean/agile principles in our operational divisions as *the* key for continuing to meet or exceed customer needs and support robust top-line growth. The SciQuest lean principles are:

- Respect the individual.
- Focus on customer value.
- Eliminate waste.
- Continuously improve.

This lean commitment has led to a number of fundamental process improvements throughout the company as we've completely reengineered our customer implementation methodology, revamped our product development organization, and fully transitioned to a

Scrum-based, agile product development process. Applying lean to our customer support processes was a natural next step. We had established an industry-leading customer satisfaction position, as evidenced by world-class customer survey results (98% of customers stated in our 2007 survey that they would refer us to a colleague or friend) as well as near 100% customer renewal rates. A lean perspective would be critical to maintaining this position in a high-growth environment.

Lewis Carroll is quoted as saying, "If you don't know where you are going, any road will get you there." So it is with process improvement initiatives. If you don't know what you're striving for, it doesn't matter how well you're doing now, nor will any recommendations for improvement make sense.

The core of the SciQuest lean initiative's success over the past two years has been a set of well-thought-out and crisply defined corporate goals. They not only serve as vehicles for communicating the corporation's strategy to our employees, but they also provide a yardstick by which to measure progress. Most importantly, from an execution perspective, these goals are quantitative and represent meaningful portions of each individual's annual total compensation.

## THE CALL TO CUSTOMER SUPPORT

In the case of our customer support department, our mandate was clear. We needed to reduce the number of customer incidents reported per module per month by 20% in Q4 2008 as compared to Q4 2007. As a SaaS company, SciQuest provides end-to-end IT support for our customers. Reported incidents can involve a range of activities: activating new release functionality, implementing changes to support new processes/functions, addressing application defects, and so on. From a SciQuest corporate perspective, automating changes or providing customers with self-service capabilities increases our customer support department's capacity, allowing it to support our projected customer growth without adding staff. More important, from our customers' perspective, a reduction in logged incidents

reinforces SciQuest's commitment to delivering great service in addition to quality software solutions.

At the end of Q1 2008, we were startled to learn that not only had we not improved over the previous quarter, but in fact our performance had deteriorated significantly. A reported incident checkpoint revealed that from a 2007 year-end rate of 1.27 incidents per module per month, the reported incidents had grown 39% to 1.75. With bonuses on the line, this was a clear call to action for the department!

We quickly assembled a cross-functional team and scheduled a two-day offsite workshop. The workshop's goal was to identify specific steps (projects) required to immediately arrest the growth of the incident count and set us back on a trajectory that would have us achieving our annual goal by year-end.

Senior representatives from all departments in the Products & Delivery Division were conscripted to the team. This cross-functional team was necessary because while customer support owned the direct relationship with each client in production, many of the reported incidents would uncover defects whose origins could be traced back to:

- The quality of the product (product development)
- The rigor of the implementation process (professional services)
- The robustness of the production infrastructure (production support)
- The integration with suppliers (supplier services)

The workshop was designed to analyze the situation and take immediate action where possible, but at the very least to make recommendations for getting us back on track. Having members of each of these departments collaborating during the workshop was critical to developing, and gaining commitment to, a timely and meaningful response to the situation.

The facilitated workshop's agenda was simple:

1. Collectively review the corporate goal and current performance (i.e., understand the problem).
2. Do root-cause analysis (in breakout teams) on the prior six months' worth of data and recommend action plans (i.e., define projects).
3. Collectively analyze and rank the project portfolio for investment purposes.
4. Individually sign up for incident-reducing projects to be executed within the next three months.

The breakout sessions produced several investment-worthy project candidates, which were then brought back to the full group for further scrutiny. Each project would have a champion, who was responsible for putting together a "mini-business case" for it. Brevity was the watchword, since the goal of the exercise wasn't to produce extensive (wasteful) documentation; it was to provide enough meaningful insight to make action-oriented decisions. Since all of the decision makers were involved in the workshop, verbal communication was sufficient. However, everyone did use a similar template, which facilitated the ranking of the projects against one another. This template consisted of:

- A brief description of the concept of the project
- A simple ROI analysis that included the number of incidents to be avoided, additional potential (qualitative) benefits, and a high-level cost summary, along with other project considerations and risk
- A skeleton outline of a project plan (8-10 steps) with "who does what when" milestones
- A discussion of key risks, barriers, and unknowns

**At the end of Q1 2008, we were startled to learn that not only had we not improved over the previous quarter, but in fact our performance had deteriorated significantly.**

A spreadsheet was created to keep an inventory of all projects under consideration and to facilitate sorting them using a simple grading rubric. Each project was assigned a composite letter score ranging from A+ to C, which reflected the value of the overall business case presented. An A+ score was given to the project that had the highest potential to remove the greatest number of incidents in the shortest period of time with the least amount of cost and risk. Conversely, the C projects were those that simply didn't give an adequate return on their investment considering how much near-term improvement we were seeking.

## WORKSHOP RESULTS

In total, the teams came up with a pool of 17 candidate incident-reducing projects. Of those, 12 projects with the potential to reduce the overall reported incidents by 22% were earmarked for Phase 1 investments; the remainder were put on the back burner.

Nine of the 17 projects were easily rated A or A+ by the group and were projected to account for over 80% of the available incident reductions. In fact, the top three projects themselves could eliminate close to 42% of the pool of targeted incidents. Let's examine the top two projects in a little more detail.

**Fully 95% of customer-managed integration failures will be eliminated when a little more sophisticated logic is built into the integration monitoring process.**

### Integration Failures

An enterprise-class purchasing automation solution has to integrate with a customer's IT systems if it is going to deliver the maximum gains in operational efficiency. While the SciQuest system can help end users with their product selection and order management processes, we are not the accounting system of record. All of our clients, to a lesser or greater extent, integrate our system with their accounting system to ensure that the appropriate accounting codes are used, funds are available, and the appropriate ledgers are updated.

It is therefore critical that we monitor these various integration points in production and that alerts be generated when the points are not operating as designed. When the source of the malfunction is within the SciQuest infrastructure or systems, it's imperative that we generate an incident so that we can quickly mobilize our resources to deal with it.

However, many malfunctions can be resolved only by the client. For instance, their system may simply be unavailable to us. In this case, while it's of great value to alert the customer to the fact that a situation exists, opening up an internal SciQuest incident is of no value. In fact, it's actually a source of significant waste, because according to our standard operating procedures for all incidents, energy has to be expended to manage each one. They have to be opened, triaged, assigned, disposed of, and reported on. Fully 95% of these customer-managed integration failures will be eliminated when a little more sophisticated logic is built into the integration monitoring process.

### Punchout Failures

A very important capability of the SciQuest solution is that it allows shoppers to "punch out" of their shopping cart and go directly to a supplier's Web site in order to

find things to buy. Shoppers love this capability because they are assured of getting thorough and up-to-date information about products they'd like to buy from a particular supplier. Suppliers love this capability because it's the simplest and cheapest way for them to ensure that their shoppers have all the information they need to make an informed purchasing decision.

Nearly 150 suppliers who provide this punchout capability are on the SciQuest network, and customers take advantage of the capability approximately 250,000 times every month. Unfortunately, for reasons beyond SciQuest's control, these suppliers' Web sites are not always available 100% of the time.

When the status of these punchout Web sites is automatically monitored, suppliers can typically repair the situation before it affects any of our customers. Unfortunately, only 10% of supplier Web sites are currently monitored. When one of the remaining 90% of unmonitored Web sites goes down, which happens about 1.5% of the time, it's usually discovered by SciQuest customers, who then promptly file an incident report with our customer support department. We then initiate contact with the supplier in order to effect a prompt resolution to the situation.

If SciQuest could increase the coverage of actively monitored punchout Web sites, and automatically notify their owners when they were unavailable, a significant number of incidents would no longer be deposited in the customer support department's queue.

### MONITORING PROJECTS IN FLIGHT

The SciQuest procurement system offers a sophisticated workflow tool that enables customers to orchestrate very complex routings of purchasing documents based on any number of factors, such as product type, spending limit, organizational hierarchy, accounting code, and so on. A SciQuest technical consultant typically designs and implements these workflow rules during the initial implementation process.

Because of the complexity of some of the workflow rules, customers often have trouble understanding why certain documents are routed a certain way, get stuck in unexpected approval queues, or the like. When one of these situations occurs, customers usually file a customer support incident in order to seek help with debugging the situation.

Since about a quarter of SciQuest's more sophisticated customers were logging several of these incidents every month, the workshop team decided to explore something called a "workflow documenter and

debugger.” Due to a relatively higher development cost, the team had graded this project A-, but team members decided to invest in it because they felt the project could deliver a significant number of incident reductions.

The approach was to “productize” the internal tools that SciQuest’s consultants and customer support representatives use to analyze these situations themselves. The goal was to at least provide enough training and documentation on the tools so that the more sophisticated customers could find value in investigating and perhaps resolving these “errant” workflow documents themselves.

The team selected a pilot group of customers to collaboratively engage in this project. The hope was that success with these customers would then fuel a more general rollout of the workflow tools. The major project risk was that the effort required to productize these tools, such that a non-SciQuest businessperson could derive value from them, would far outweigh the return on the investment.

Unfortunately, after the first two customers got hold of the workflow tools, they told us that the tools introduced more complexity and consumed more of their time than they were willing to tolerate. The ROI of this particular project was not going to be realized, so we cancelled it after the first iteration.

We were all disappointed that we weren’t able to complete the project and get this longstanding workflow debugging monkey off of our back. But since the customer feedback was loud and clear, we were happy to redeploy resources to other higher-ROI projects in the portfolio.

## REAPING THE REWARDS OF THE PORTFOLIO

Now let’s fast-forward a couple of months to the end of Q2 2008. By that point, several of the portfolio projects had yielded their projected benefits. Overall, the number of reported incidents per month had shown a 22% improvement over the previous quarter, running at 1.37 incidents per module. This was still quite far away from our year-end goal of 1.02 incidents per module, but the momentum had shifted in the right direction.

We kept working on these incident-improving projects over the summer and continued to demonstrate progress. In September 2008, we reconvened the entire project portfolio team to reassess where we were and repeat the workshop process developed earlier in the year. However, in this iteration, we split up our working teams by customer. Our hypothesis was that a

customer-oriented view might yield additional insight into the issues specific customers were having that we might then target for improvement. With this approach, we reprioritized our portfolio in order to focus on those projects that offered the greatest likelihood of making our corporate goal.

In addition, for the remainder of the year, we increased our progress reporting frequency from monthly to weekly. We also evolved the tone of these review meetings from “status reviews” to “progress celebrations” (where appropriate). This gave each project champion an opportunity to gain public recognition for his or her process improvement efforts while simultaneously cross-fertilizing the rest of the project teams with new ideas, techniques, and results.

## LEAN LESSONS LEARNED

SciQuest has enjoyed great success over the past three years due in major part to the injection of lean thinking into the corporation. We’ve wholeheartedly embraced lean/agile in our customer implementation projects and incorporated it into our product development and management processes. In this article, we’ve recounted our efforts to extend lean principles and practices into the customer support organization, specifically to improve customer responsiveness and internal efficiencies.

We’ve relearned many important lean lessons working with the customer support team over the course of this year:

- A specific, attainable, measurable, and customer-linked goal is critical to fueling a well-grounded process improvement program. If 20% of our bonuses had not been linked to reducing the reported incidents to a specific level by year-end, focus on this problem would not have been as intense.
- Executive sponsorship, both in deed and words, is necessary to sustain the programs. From funding the bonus pool that is directly linked to the business goals, to participating in the process improvement workshops, to allocating the resources to staff these projects, to participating in all of the review meetings, executive sponsorship is critical to success.
- Cross-functional collaborative teams clear the way for real results to emerge. That’s because they not only tackle the difficult but high-potential cross-organizational boundary problems, they also garner personal commitment to execution.
- For the lean organization, cancelling projects in a portfolio is just as important as sponsoring the right

ones (albeit more difficult). Sometimes great ideas just don't pan out. It's better to pull the funding early and redirect the resources to other more worthy projects than to feed a dead horse.

- In-flight projects need regular reviews to revalidate their goals, assumptions, resources, commitment, and progress. In a fast-moving organization, priorities can change quickly. Revalidation and reprioritization are critical.

## WHAT'S NEXT IN SciQUEST'S LEAN JOURNEY?

As 2008 comes to an end, the results of our customer support lean projects have been significant. At year-end, we're at .81 customer incidents per module per month, a 36% decrease from the same time last year. Obviously, the results are extremely motivating — what remains to be seen is how we can use this excitement to feed ongoing improvements next year (and beyond).

We've already begun discussions focused on continuing our lean drive. Key goals and objectives for 2009 were to be committed in January, after vigorous discussion and debate throughout the organization. And we're certain many of the tactics used in our customer support improvement efforts in 2008 will be applied again. That said, other improvement opportunities require additional focus:

- **Goal setting.** We need to continue to balance our goals between achievable targets (typically based on year-over-year improvement) and breakthrough "game changer" targets (from competitive or best-in-class benchmarking).
- **Gaining goal ownership.** Top-line goals shared across all operational divisions are great for sparking cross-functional process improvement efforts. Our approaches for achieving goals, however, have been widely varied. While there isn't a "one size fits all" approach that will work, we do need to more formally align teams and resources earlier to better achieve our goals.

- **Developing a process improvement project portfolio roadmap.** To be developed early in the year and anchored by our goals, this roadmap will be modeled after our customer support project portfolio successes and monitored/reprioritized regularly.
- **Ongoing skills development.** Our continuous improvement efforts are still rather new, and we need to continue to develop our people's skills, especially around Kaizen workshops and lean project administration. This will facilitate lean project identification and rationalization events and ongoing project portfolio management (so that people know how to identify, and secure, needed resources).

SciQuest continues to exceed the top-line growth goals expected by our investors. We understand, however, that last year's success doesn't guarantee next year's. Our commitment to lean/agile is critical to building a successful and sustainable company.

*As COO at SciQuest, Jamie Duke drives the planning and management of product design, launch, and enhancements for all of SciQuest's technology solutions, including procurement applications, materials management solutions, and scientific research applications. Mr. Duke most recently served as CIO of BuildNet, a solutions provider for the construction materials industry. His prior experiences include roles as VP of Sales and Marketing at GE Capital Mortgage, Group VP for Technology and Alternative Channels for First Citizens Bank, and a management consultant with McKinsey & Co. A graduate of Duke University, he also has a master's degree from MIT's Sloan School of Management. Mr. Duke can be reached at [jduke@SciQuest.com](mailto:jduke@SciQuest.com).*

*Sam Bayer is a Senior Consultant with Cutter Consortium's Agile Product & Project Management practice and a contributor to that advisory service. He is the founder of MarketAcuity, a consultancy established in 2001 dedicated to bringing agile management principles and practices to high-tech organizations. Over his 28-year career, Dr. Bayer has held management positions at IBM, Amdahl, AGFA, Sapiens, and Haht Commerce. In addition, he was the founder of Axiom Systems and successfully oversaw its IPO; was President of Datacraft Solutions; served as the acting CEO for Memex's US operations; and is currently the CEO of b2b2dot0, a SaaS startup in the B2B e-commerce market. Dr. Bayer also sits on the advisory boards of the Association of IT Professionals, 6th Sense Analytics, Nipendo, Ltd., and Memex. He is a published author and a frequent presenter in the agile community. He earned a PhD in chemistry from the University of Florida. Dr. Bayer can be reached at [sbayer@cutter.com](mailto:sbayer@cutter.com).*



# Transformative IT: Creating Lean IT Portfolio Management ... or Not

by Bob Benson and Tom Bugnitz

The call for papers for this issue of *Cutter IT Journal* stated:

Many experts in the agile space believe there is now a significant misalignment between the way agile projects are run and the way IT projects are governed in general. IT program and portfolio management, in particular, seem to be at the root of many of these alignment issues....

How can program and portfolio management be improved?<sup>1</sup>

The solution put forward is “lean portfolio management,” but is this a practical idea for IT organizations? We’ll conclude: maybe yes, maybe no, and it depends on two issues.

The first is that lean portfolio management touches on the fundamental issue in IT decision making: justifying and reconciling conflicting demands for resources. This may require more structure and deliberation than lean portfolio management is prepared to provide. The second is that effective business management participation in this decision making may be hard to come by.

## LEAN PORTFOLIO MANAGEMENT REQUIRES BUSINESS MANAGEMENT PARTICIPATION

In a recent Cutter Consortium *Executive Report*, Guest Editor Sanjiv Augustine and coauthor Roland Cuellar identified the following lean/agile portfolio management principles:<sup>2</sup>

- Align continuously.
- Manage project throughput. (This focuses on one project, one team at a time, purging “sick” projects to reassign project resources.)
- Manage system constraints. (This focuses on continuously tracking progress and identifying bottlenecks often.)
- Applying these principles to project portfolio results in practices that are:
  - Always focused on the highest priorities
  - Responsive to the business in managing changing priorities

- Always delivering the highest possible value to the organization
- Always in alignment with strategy
- Delivering projects at the greatest possible speed
- Maximizing investment returns

According to these principles, lean/agile techniques will improve portfolio management when they:

- **Reduce the number of projects worked on at a given time.** This increases project efficiency and, by extension, will produce more completed projects.
- **Reduce the scale of individual projects.** They accomplish this by making intermediate project decisions to go forward (including funding decisions), in effect breaking up big projects into little ones.
- **Constantly reprioritize and optimize the assignment of resources to projects.**
- **Ensure that the projects addressed are the highest business priorities.**

For lean portfolio management to succeed, the decision and review processes — IT governance in other words — must be capable of achieving these goals. The portfolio management process needs to accept the idea that fewer projects underway at a time is better than more; dedicated project teams are a good practice; smaller projects are better than big ones; and constant review of business priorities, individual project performance, and the overall portfolio is needed. For example, in a company of reasonable size, the existing portfolio management process might have 100 projects prioritized and scheduled. We’re expecting that a lean/agile process will prioritize and schedule, say, 10 at a given time and dynamically make decisions about the projects as events unfold.

If that’s the case, then lean portfolio management cannot operate in a technical context devoid of active business management participation. Lean portfolio management demands a significant amount of middle and upper management activity in a lasting manner.

## LEAN PORTFOLIO MANAGEMENT NEEDS BUSINESS MANAGEMENT PARTICIPATION IN SPECIFIC WAYS

For any IT governance to work, we need the active participation of business management in these decision-making areas:

- **Project approval** (e.g., project sponsorship)
- **Decision-making processes** (e.g., setting the ground rules for making prioritization decisions)
- **Prioritization decisions** (Which are the most valuable projects in terms of impact on the business?)
- **Confirming prioritization decisions** (Which projects should actually be done? What are the “mandatory” or “compliance” projects?)
- **Scheduling decisions** (Which projects should be done first?)
- **Content participation** (e.g., agreements on business strategic intentions, business goals, business change management)
- **Monitoring and mitigation** (e.g., developing and agreeing to the mitigation plans — including project stoppage, deferment, etc. — as risks are recognized in a project or in the portfolio)

While we describe these participative roles here with respect to managing the project portfolio, we can develop similar “need” lists for the other IT governance activities, such as IT planning, financial management, and cost containment. In all these areas, we need business management to focus on balancing functional silos with enterprise needs and on enforcing standards and process definitions.

*The challenge is that business managers don't want to spend the time and energy or risk the political exposure to participate in this way in an ongoing fashion. We often find that business management's views of how governance works is limited to these perspectives:*

- “Approval” is a rubber-stamping review of individual projects.
- Senior managers say, “This is what we hire you to do,” leaving IT to make the decisions.
- Business managers see their role in IT governance as shepherding their individual projects, not making decisions for the good of the enterprise.
- “Prioritization” is treated as a matter for individual business units and individual projects. Business managers ask, “When will my project be done?” rather than “What is the optimal set of projects to be done?”<sup>3</sup>

- Few business managers care whether projects for managers in other business units are high priority. The siloization of business is very apparent. Every organization struggles with cross-silo prioritization, and most fail, with the result that IT resources become silo-specific.

How many of us have sat through steering committee “dog and pony shows” in which everything is approved? (“After all, how can we turn down his project knowing that he’s about to vote on ours?”) How many of us have gone to governance meetings only to encounter, not the designated manager, but her “representative” — even though that representative didn’t attend previous meetings or bother to read the support material? These situations arise partly because current IT governance schemes focus only on participation, not on actual decision making. They are also partly due to the zero-sum-game aspect of project governance decisions; they require saying no to other parts of the business, which is not easily done.

How do we effectively change all this in lean portfolio management? Over the last 25 years, the most pressing problem our clients have faced is getting business management to engage in an *ongoing* fashion in IT portfolio management and IT planning. Oh sure, everyone gets participation for the first cycle, but then it declines from there. The upshot is that IT managers end up driving the portfolio management and IT planning processes — and making the decisions.

The basic dilemma is that it is easy to say we need to establish effective governance roles, engage business management in them, and demonstrate that the result is better than what we have now. What’s hard is doing so in ways that encourage and reward participation and gain enthusiastic support. Simply telling management to participate has never worked.

## IS LEAN PORTFOLIO MANAGEMENT COMPATIBLE WITH CORPORATE IT DECISION-MAKING PROCESSES?

Portfolio management is just one of several highly interrelated IT governance processes. And what is IT governance about? Simply, it’s about deciding where to spend money (whether on projects, “lights-on” support, or infrastructure) and assessing the performance and bottom-line impact of that IT spending. IT governance is IT decision making from the business perspective, both making the decisions and determining *who* makes them.

There is a sense in this issue of *Cutter IT Journal* that the company’s IT decision-making processes are the problem and that changing the management culture to



agile principles may be our best hope. Before we get too excited, though, we feel compelled to point out that agile/lean portfolio management principles may be incompatible with the practical requirements of IT decision making overall.

We often use the diagram in Figure 1 to describe the elements that go into IT decision making. The diagram refers to basic practices the CIO and the IT staff perform in the context of managing IT for the business and, more significantly, maximizing the business value/impact of the IT expenditures. The key point is that every one of these IT governance practice areas is about justifying and reconciling conflicting demands for resources.

Figure 1 shows the three factors applicable to all these IT governance practices:

1. **Strategic intentions** (the framework we use to describe an enterprise’s basic strategies and management goals)
2. **Portfolios** (including projects as well as applications and infrastructure)
3. **Services** (the actions and resources the IT organization actually delivers to the business organization)

Every one of the governance practices shown in Figure 1 requires confirmation of appropriate business priorities, responsiveness to changing priorities, delivering highest value, alignment with business strategy, effectiveness and productivity, and maximizing IT’s returns — exactly the “agile” principles listed above. And these

are resolved through the use of strategic intentions, portfolios, and IT service assessments. So the problem really isn’t that agile principles are inconsistent with broad IT decision-making practices. As we’ll see, the difficulties lie elsewhere.

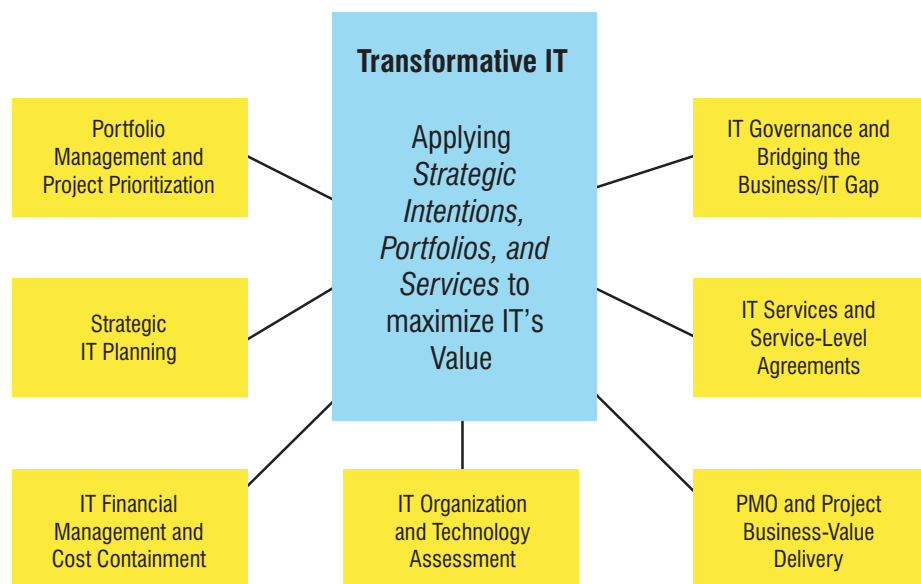
### Business Management Participation Is Required in All Areas

Substantial business management participation is required in all the governance practice areas shown in Figure 1. This participation is reflected in attention to business strategic intentions (and hence alignment and priorities), portfolios (and hence value, cost containment, alignment, etc.), and services (hence their cost to the business, performance, and value). In every case, the basic IT decision making is about where to spend IT resources and the resulting performance and business impact.

Here’s the rub: to improve portfolio management by adopting lean/agile practices, we are asking that the portfolio management process dynamically make resource allocation decisions that are normally made in other IT governance activities and in a less-than-agile time frame. Business managers will wonder why and/or will resist the duplicate efforts.

### Competing IT Decision-Making Processes

While there are conceptual commonalities in the IT governance practices (namely, strategic intentions, portfolios, and services), there are also considerable in-the-trenches process and decision-making



©2009 Robert Benson

Figure 1 — IT governance processes.

interlinkages among them. Projects, for example, often come from the decisions made in a strategic planning exercise. Projects may also come from decisions made in cost containment or service-level assessments. Budgets are a sum of all the decisions made. A company's strategic intentions apply to all, and as they change, also affect all. And so forth.

These commonalities and interlinkages throw sand in the works. Much of it has to do with timing, which of course is one of the complaints from the agile community. Strategic planning and budgeting are almost always annual processes, and annual processes have a way of bogging down project planning and prioritization processes as well. It comes down to who makes the basic underlying decisions (e.g., establishing what the company's strategic intentions and priorities are), when they are made, and whether the decisions "stick" in all the related IT decision-making activities.

### **Siloization**

The siloization of companies infects practically everything. For example, it makes dealing with the timing questions so much more difficult. In many companies, the sequence of events is strategic corporate business planning, then individual line of business (LOB) planning, and then budgeting. In the midst of all this, IT is expected to prepare budgets that affect each LOB, as well as a project plan that helps define the overall budget. Dealing with siloization is also at the heart of reconciling resource requirements; deciding the priority of projects from different silos is enormously difficult for most companies.

### **CAN LEAN PORTFOLIO MANAGEMENT ALONE JUSTIFY AND RECONCILE CONFLICTING DEMANDS FOR RESOURCES?**

The challenge lies in the "reconciliation" part. As we'll discover in the culture section below, companies do differ substantially. A major difference is the degree to which competing demands for resources occur — for example, between two LOBs, between two functional areas of the business (e.g., manufacturing vs. marketing), and certainly within each LOB or functional area. In the project portfolio, this forms the basis of prioritization, which is the selection of one project over another. Agile and lean methods make strong cases for quick and simple decisions and constant reassessment of priorities — and that's extremely laudable. But in many companies, it is also very complicated and politics-laden. (This is why

the common elements of strategic intentions and portfolios are so vital in all the IT decision-making practices.)

But the problem is, can the practices of lean portfolio management be successful when the primary means of resource allocation — and the means of resolving competing demands — are found in other IT decision-making processes, such as budget, capital budgets, IT strategic planning, or corporate business planning? One possible conclusion is that, rather than quicker decisions, we need slower, more thoughtful decisions when addressing cross-business resource requirements. While a business context with simple resource allocation requirements is perfect for lean methods, it is not clear that they would be helpful in more complex environments (e.g., a multiple-mission government agency). It really does depend on culture, as we'll discuss later on.

The current economic situation makes these problems more challenging. In today's economic climate, IT organizations are expected to keep expenses level at best, or even reduce them overall year over year. This means that project money is scarce and the object of intense competition. That competition is premised on knowing what the projects are — which gets in the way of the lean idea of making project funding commitments incrementally. But consider the reality that all this is a zero-sum game. From our 25-year experience, we know managers understand that if we approve one project, we can't do something else. Everything is a tradeoff. From a governance process perspective, this creates winners and losers. This is the case even though the business overall is the winner — assuming project selection is based on bottom-line impact — and even though reasonable people can differ on that bottom-line impact.

### **BUSINESS MANAGEMENT MAY NOT BE INTERESTED IN PARTICIPATING**

Complicating matters further is that fact that business managers in general appear to dislike and distrust IT decision-making activities. This results partly from the exhaustion factor — we try to engage the same managers in many IT decision-making processes. It also results from the reality of the economic pressures. But it is a real fact: it is difficult to get quality business management participation on a continuing basis.

In 2008, we surveyed companies about IT governance as part of a *Cutter Benchmark Review* issue on dynamic IT, or the ability of companies to respond quickly to turbulent business conditions.<sup>4</sup> Table 1 shows how 103 companies characterized as being in turbulent

industries and/or particularly concerned about IT's ability to respond to change felt about IT governance.

The results shown in Table 1 are consistent with our field experience: senior corporate and business unit managers simply do not believe in existing IT governance processes. Figure 2 shows the percentage of participating companies that thought various IT governance processes actively improve their ability to be dynamic and the percentage that thought they actively hinder that ability. Not surprisingly, agile development was the process considered by most companies (70%) to actively improve IT's ability to respond to change effectively. Other IT decision-making processes also were thought to help; for example, 56% of the 103 respondents thought that portfolio management actively improves their ability to be dynamic. On the other hand, nearly one-third (29%) of the companies thought that steering committees significantly inhibit their ability to change, and one in five (20%) thought the same of portfolio management.

Taking Table 1 and Figure 2 together, the data shows overall that:

1. Most companies do use most of the IT decision-making practices.
2. Business managers and corporate executives generally don't think they all work well.
3. Many believe IT decision-making processes interfere with the achievement of dynamic IT (of which agile/lean techniques are a part).

We find that business management is exhausted, jaded, and/or confused as to why the same issues (such as

business priorities) come up again and again. So often the complaint is heard, "Why are you asking the same questions you asked before?" Or "We settled that issue last month — why does it come up again [in a different IT decision-making context]?" The result is that business managers may simply not be interested in participating in IT decision making in general and lean portfolio management in particular.

### THE SUCCESS OF LEAN PORTFOLIO MANAGEMENT DEPENDS ON RELATING TO COMPANY CULTURE

It is clearly desirable for companies to adopt such lean portfolio management practices as focusing on the highest priority at any given time, delivering the highest possible value, and aligning their efforts with corporate strategy. But as we've also seen, adjudicating competing claims for IT resources is tough to do, and the multiple demands for business participation in IT decision-making mean that most such requests are met with resistance.

We'd like to overcome these challenges with a common lean portfolio management approach, but as we all know, "one size does not fit all." Companies are different one from another in critical governance ways. In our experience, two separate approaches can help organizations define a successful lean project portfolio management process.

#### The "IT Importance" Approach

This first approach is based on Warren McFarlan's work in early 1980s. McFarlan was among the first to formalize the concepts of IT portfolio management,<sup>5</sup> and he

Table 1 — How Managers at Various Levels Feel about IT Governance

Question: Is IT Governance Effective?		
Senior IT Managers	Senior Corporate Managers	Business Unit Managers
52% Yes	43% Yes	42% Yes

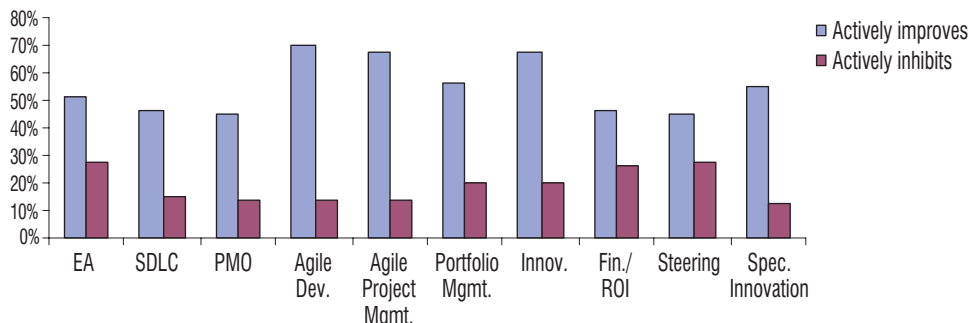


Figure 2 — Percentage of companies that say governance processes improve vs. inhibit dynamic IT.

later proposed a quadrant analysis of the use of IT in companies.<sup>6</sup> He suggested that companies fall into one of four quadrants depending on how important IT currently is to the company and how competitively important IT would be in the future. Figure 3 shows how these two factors combine to form four quadrants.

McFarlan’s point is that IT decision making is vastly different for companies in each quadrant. For example, the primary drivers for decision making for companies in the Support quadrant are cost containment and operational excellence, whereas the primary decision-making drivers for companies in the Strategic quadrant are strategic business support and strategic effectiveness. In other words, decision making about scarce IT resources will be driven by very different factors — and performed by different people — for companies in different quadrants (see Table 2).

Such analysis is fine as far as it goes, but the reality is that companies typically do not fall easily into just one quadrant. A company with multiple LOBs often has one

or more in each quadrant. Or a company may have functional areas (e.g., manufacturing, finance, marketing) in two or more quadrants.

To the extent that LOBs or functional areas are in multiple quadrants, then the problem of reconciling conflicting demands for resources is significantly more difficult. In our experience, it is unlikely that the portfolio management process can successfully deal with resource allocation when large components of the business are in different quadrants.

### The Cultural Assessment Approach

The second approach to defining an appropriate lean portfolio management process, which is based on our research and consulting experience over the last 25 years, is an assessment that examines the culture of business governance and planning/management. The two factors to consider are the current governance culture for the business as a whole and the way the company times its basic planning and budgeting processes.

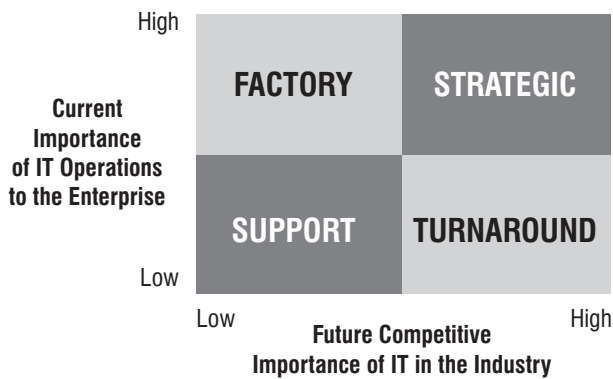
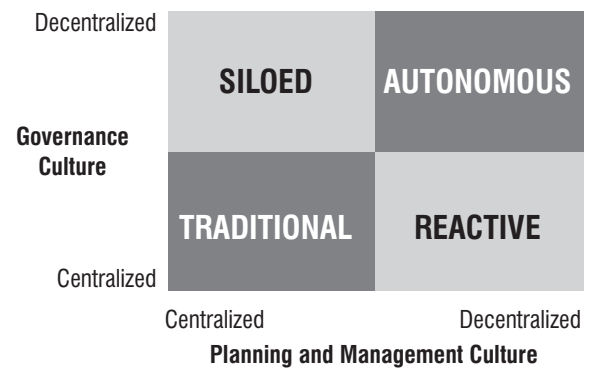


Figure 3 — The current and future importance of IT to the company.



©2009 Robert Benson

Figure 4 — Management culture quadrants.

Table 2 — Portfolio Management Decision Making by “IT Importance” Quadrant

IT Importance Quadrant	Who is ideally involved in lean portfolio management?	What is the most important basis for making basic portfolio decisions?
Strategic	Senior leadership committee	How well does a project support the competitive strategies of the company?
Turnaround	Senior leadership committee	How well does a project support the competitive strategies for the company, plus the strong financial (ROI) justification?
Factory	Senior leadership of business units	How well does a project improve the operational excellence of the company?
Support	Managers and directors	To what extent does a project contribute to cost containment and/or reduction?

The first factor examines how the corporate executives view the management of the many parts of the company — its LOBs and its functional areas. A “centralized” culture operates as a single business; major decisions are made in a single process, perhaps by the senior leadership team. The “decentralized” culture puts most major decisions in the hands of each LOB or functional area, typically treating only financial performance issues at the highest corporate decision-making level.

The second factor examines the timing of management processes. “Annual” companies look at budgets, financial results, competitive decisions, and the like in a yearly management process. “Episodal” companies look at basic decisions as the need occurs, often monthly or quarterly or as events dictate.

Figure 4 displays the quadrants that result. “Siloed” companies are annual, but most decisions are made at the business unit level. “Autonomous” companies are collections of business units, each with its own decision-making processes. “Traditional” companies (our term) are centrally managed and operate with annual cycles. “Reactive” companies are centralized but respond to circumstances as they arise. Table 3 shows how the portfolio process itself may be different according to the management culture quadrant.

The relationship of lean portfolio management to corporate planning processes, to budgeting, to capital budgeting, indeed to all of corporate governance, is determined by this cultural assessment. For example, in the autonomous quadrant, the principles of lean management are *much* more easily adopted than, say, in the traditional quadrant. The reason is that, in the former, none of the disadvantages (annual corporate processes, attempts to allocate resources across silos, etc.) apply. And the specific approaches will be different for each business unit in the autonomous quadrant.

But as with the IT importance quadrants above, a reasonably large company may exhibit more than one cultural profile. Again, any company with multiple LOBs or separate functional units (e.g., marketing, manufacturing, finance) may find that the culture and characteristics of the various units are in different quadrants.

So the challenge is to find approaches that satisfy our requirements — for business engagement, for real decisions to be made, for effective decision making using lean and agile methods — that accommodate the multiple profiles described above. Taking a uniform approach in the face of the cultural diversity we’ve just sketched would doom the effort to failure. Table 4 characterizes the application of lean portfolio management in multiple-profile companies.

Table 3 — Portfolio Management Decision Process by Management Culture Quadrant

Management Culture Quadrant	Who is ideally involved in lean portfolio management?	What is the scope of the portfolio decisions?
Siloed	Senior leadership of business units	Separate portfolios by business units
Autonomous	Senior leadership of business units	Separate portfolios by business units
Traditional	Senior leadership committee	Enterprise-wide portfolio
Reactive	Ad hoc	It depends

Table 4 — Applying Lean Portfolio Management in Multiple-Profile Companies

Quadrant Characteristics	A reasonable approach to the scope of lean portfolio management
A single profile for both IT importance and management culture	This is the sweet spot for lean portfolio management at the enterprise level.
Single culture profile, multiple IT importance profiles	With care in the selection of decision-making parameters (like strategic intentions) and with strong support, an enterprise-wide approach is possible — but there are risks.
Multiple profiles	The scope should be carefully determined, likely bounded by separate profiles for separate business units.

## CONCLUSION

We said at the outset that this article would address whether program and portfolio management can be improved by the application of lean/agile techniques. Lean portfolio management is a main component of IT governance and has to be undertaken with that in mind, taking the following steps:

- Build on the idea that all IT governance deals with justification and reconciliation of competing demands for resources. It's a question of where these decisions are to be made.
- Understand that lean methods are achieved only by direct engagement of business management.
- Do a self-study to determine the company's profile. Add to this the analysis of the requirements for governance. This will entail a cultural assessment as well as understanding the exact process interlinkages among your IT governance activities.
- Align processes, decisions, and governance with what's important to business management. This activity, too, is based on the cultural assessment.
- Discourage business management from treating portfolio management as a means for shepherding their individual projects.
- Emphasize that business management participation has to result in real decisions.

*Can portfolio management be improved with lean methods?*

Yes, if the issues of resource reconciliation and business management participation are successfully addressed.

*Can lean/agile methods be applied to the other IT decision-making processes?*

We have been discussing lean portfolio management as it might be applied alongside the other IT decision-making processes. But perhaps the relevant questions are:

- Can lean portfolio management replace any of the other IT decision-making processes and thus resolve the problems?
- Can lean/agile principles be applied in those other processes?
- Can a CIO resolve the issues described above by changing all the interrelated IT decision-making processes in favor of lean/agile methods?

If we are to achieve success in any of these objectives, we propose these four ground rules:

1. **Any initiative to move more broadly to “adaptive governance” should focus on project portfolio management as the centerpiece.** Planners should understand all these related governance areas (recall Figure 1) and how project decision making fits in. This particularly applies to budgeting (and/or chargeback). The tendency in all of these areas is to focus on the “plan” and an annual cycle. To be successful in changing portfolio management, any lean portfolio management initiative should deal with how the portfolio and the related planning processes relate to these annual cycles. At the same time, the initiative must also deal with the reality that each business unit generally has different interests and priorities — and as suggested in the previous section, different IT criticalities and management cultures.
2. **The adaptive initiative has to produce real decisions at every step.** Whether these decisions involve prioritization, or budgets, or project approvals, the process must be *the* decision-making process for individual projects and the project portfolio overall.
3. **The process has to be transparent.** That is, the data that leads to decisions has to be apparent and the ground rules agreed to.
4. **Projects — whether conventionally “troubled” or just of lower priority — must be rejected or stopped as a result of the lean portfolio management process.** Without the possibility that projects can be stopped, management will have no stake in the process.

## ENDNOTES

<sup>1</sup>We are concerned primarily with business organizations that are users of IT rather than sellers of IT. For them, IT projects are intended to change how the organization does business through its internal processes and Internet customer interactions. This is a more complex management problem than managing IT projects resulting in a product to be sold, because to be successful, governance decisions must focus almost exclusively on business management and achieving the expected changes to process and customer interactions.

<sup>2</sup>Augustine, Sanjiv, and Roland Cuellar. “The Lean-Agile PMO: Using Lean Thinking to Accelerate Agile Project Delivery.” Cutter Consortium Agile Product & Project Management *Executive Report*, Vol. 7, No. 10, 2006.

<sup>3</sup>It is very important in this discussion to keep this in mind. Organizations for which IT is mostly a part of the products being sold are different from organizations for which IT is mostly part of the production and management process. This difference lies in how the organizations deal with process change and change management. This is a minor issue for "IT as a product" companies and a major issue for "IT is part of the production/management process" companies. For the latter, process change and change management are important factors in prioritization.

<sup>4</sup>Benson, Robert J., and Tom Bugnitz. "Linking IT Budgeting, Governance, and Value." *Cutter Benchmark Review*, Vol. 8, No. 7, 2008.

<sup>5</sup>McFarlan, F. Warren. "Portfolio Approach to Information Systems." *Harvard Business Review*, September-October 1981, pp. 142-150.

<sup>6</sup>McFarlan, F. Warren. "IT Changes the Way You Compete." *Harvard Business Review*, May-June 1984, pp. 98-103.

*Robert J. Benson is a Senior Consultant with Cutter Consortium's Business-IT Strategies practice and a Principal with The Beta Group. He applies more than 40 years of academic and corporate experience to assist companies and government agencies with understanding the business value of IT, strategic and financial IT management, strategic IT planning, effective IT application development, and IT governance. Since 2001, he has written numerous Cutter Executive Reports and Business-IT Strategies E-Mail Advisors and has consulted for and conducted workshops with Cutter clients in the US, Mexico, and Poland. Mr. Benson has been instrumental in the development of portfolio management methods and strategic and financial management methodologies based on Information Economics that have been used by companies and consulting organizations around the world. He has conducted executive seminars and management courses on these subjects globally and has consulted with more than 100 companies and*

*organizations in 20 countries. Mr. Benson taught computer science and information management at Washington University in St. Louis for 40 years, where he also served as Associate Vice Chancellor for Computing and Communications and in other executive positions. He also has taught information management at Tilburg University (the Netherlands) for 20 years and is a member of its faculty. He is coauthor of several books and numerous articles and monographs, including From Business Strategy to IT Action: Right Decisions for a Better Bottom Line, Information Economics: Linking Information Technology and Business Performance, and Information Strategy and Economics: Linking Information Systems Strategy to Business Performance. He can be reached at [bbenson@cutter.com](mailto:bbenson@cutter.com).*

*Thomas L. Bugnitz is a Senior Consultant with Cutter Consortium's Business Technology Trends & Impacts practice and Business-IT Strategies practice. He is also President of The Beta Group. His specialties include business and organization transformation through information systems, IS organization management, technology planning, the impact of new and emerging technologies, the impact of technology on business strategies, IS operations, strategic systems applications, and leadership for executives. Mr. Bugnitz is closely associated with Washington University in St. Louis and participates actively in the development and execution of research in information management. He is coauthor of several books on computers and computer programming. Since 1974, he has worked in the field of business and information management. He has consulted around the world with many companies and government organizations. Mr. Bugnitz brings practical experience to bear on his consulting and teaching assignments, having worked at all levels of IS organizations, including 10 years managing large data centers and telecommunications operations. He has also served as an expert witness for the last 20 years in the field of software acquisition and development. He can be reached at [tbugnitz@cutter.com](mailto:tbugnitz@cutter.com).*

## About Cutter Consortium

Cutter Consortium is a truly unique IT advisory firm, comprising a group of more than 100 internationally recognized experts who have come together to offer content, consulting, and training to our clients. These experts are committed to delivering top-level, critical, and objective advice. They have done, and are doing, groundbreaking work in organizations worldwide, helping companies deal with issues in the core areas of software development and agile project management, enterprise architecture, business technology trends and strategies, enterprise risk management, metrics, and sourcing.

Cutter offers a different value proposition than other IT research firms: We give you Access to the Experts. You get practitioners' points of view, derived from hands-on experience with the same critical issues you are facing, not the perspective of a desk-bound analyst who can only make predictions and observations on what's happening in the marketplace. With Cutter Consortium, you get the best practices and lessons learned from the world's leading experts, experts who are implementing these techniques at companies like yours right now.

Cutter's clients are able to tap into its expertise in a variety of formats, including content via online advisory services and journals, mentoring, workshops, training, and consulting. And by customizing our information products and training/consulting services, you get the solutions you need, while staying within your budget.

Cutter Consortium's philosophy is that there is no single right solution for all enterprises, or all departments within one enterprise, or even all projects within a department. Cutter believes that the complexity of the business technology issues confronting corporations today demands multiple detailed perspectives from which a company can view its opportunities and risks in order to make the right strategic and tactical decisions. The simplistic pronouncements other analyst firms make do not take into account the unique situation of each organization. This is another reason to present the several sides to each issue: to enable clients to determine the course of action that best fits their unique situation.

For more information, contact Cutter Consortium at +1 781 648 8700 or [sales@cutter.com](mailto:sales@cutter.com).

## The Cutter Business Technology Council

The Cutter Business Technology Council was established by Cutter Consortium to help spot emerging trends in IT, digital technology, and the marketplace. Its members are IT specialists whose ideas have become important building blocks of today's wide-band, digitally connected, global economy. This brain trust includes:

- Rob Austin
- Ron Blitstein
- Christine Davis
- Tom DeMarco
- Lynne Ellyn
- Jim Highsmith
- Tim Lister
- Lou Mazzucchelli
- Ken Orr
- Mark Seiden
- Ed Yourdon