

overload 89

FEBRUARY 2008 £3

Orderly Termination of Programs

Getting your code to work is one thing; getting it to stop is another

An Introduction to FastFormat

Matthew Wilson introduces a useful C++ formatting library

On Management: Caveat Emptor

More sage advice on managing software teams and projects

Measurable Value with Agile

Agile development is touted as the next silver bullet for software development. How can you tell if it's delivering value?

Measurable Value with Agile

Are you solving the right problem or simply solving the problem right? Ryan Shriver shows us that both are needed for value delivery.

Agile is one of the hottest trends in IT. It's the shiny new toy that's gone from underground movement to mass marketed panacea. It's done all this within the last ten years or so. Now agile is maturing and being marketed as 'delivering business value', but there's little agreement in the agile community on what this is and how to measure it.

This past summer I spoke at the Agile 2008 conference in Toronto where there were over 1,500 attendees and 400 sessions on everything you could possibly want to know about agile. There was an entire series of presentations on 'Customers & Business Value', yet amongst the presentations, none that I saw covered:

- What is value? What do we mean when we say agile delivers business value?
- How do you measure value?
- What do you measure it with (and when)?

After the conference I thought, 'For a community where nearly everyone talks about delivering business value and prioritizing by business value, I've seen very few specifics on how to implement this in practice'. Yes, organizations see features getting implemented and they track velocity, but they've got no real way to measure the value delivered by these features.

The challenge – are we delivering the right things, now?

In my experience, most IT project teams, even agile ones, rarely grasp the business objectives of their stakeholders investing in the project. Project leaders either don't understand, can't articulate or don't care what drives business value or how it aligns with business strategy. It's sad to witness the flurry of new project activities while nearly everyone fails to distinguish between:

Delivering the right things and delivering things right

This is especially acute in the agile community and it's setting a dangerous precedent. As methods like Scrum increase in popularity [VersionOne09], an overall focus on real value and delivering the right things becomes even more critical. Today, in practice, teams can be performing Scrum flawlessly (delivering things right) only to find out they were doing the wrong project all along (delivering the wrong things) because they didn't understand the real business objectives. The result is an investment that may result in running software that delivers no business value despite the (apparent) success of the agile process. Whoops!

But it doesn't have to be this way, as this article will demonstrate.

Ryan Shriver is a Managing Consultant with Dominion Digital, a Virginia-based process and technology consulting firm, where he leads the IT Performance Improvement Solution. With a background in systems architecture and large-scale Agile development, he currently focuses on measurable business value and systems engineering. He writes and speaks on these topics in the US and Europe, posting his current thoughts at theagileengineer.com.

Determining the right thing doesn't have to be costly and complex. In fact, it can be done without changing Scrum and without slowing teams down.

In this article the reader will learn that measurable value using quantified business objectives and Scrum can work together to ensure teams are both focused on the right things and delivering the things right. Used together, teams can move beyond feature-builders to value-delivers, measuring progress not in features built but value delivered using business-defined metrics.

Now is the time for agile teams and the agile community to seek out and embrace practical ways to demonstrate measurable business value. By engaging the business and quantifying their objectives, agile teams can ensure investments are aligned with strategies. The agile community, including you, can help IT transform from feature builders to value deliverers.

Value delivery approach

This article presents value delivery, a practical approach for measuring the stakeholder value delivered by teams. This approach directly aligns business strategies and stakeholder objectives using simple quantitative methods for clarity.

To understand this approach, we must first get to the root of the issue with defining value. It is, not surprisingly, communication. In most organizations a communication gap exists between the lofty prose used by leadership to describe strategic initiatives and the planning prose required by teams to deliver business value. Value delivery bridges this communication gap by transforming vague language into clear objectives that can be planned and measured. Teams that help stakeholders get closer to achieving these objectives are delivering tangible value to the business.

Value delivery advocates measuring value using quantified business objectives in alignment with business strategies. It does not advocate measuring value using features, functions, function points, epics, user stories or tasks. These are practically all too low-level for measuring value to be worth the investment.

Rather, to deliver value, people, process and technology must be properly blended so that stakeholders' objectives are met. Value delivery advocates a systems-thinking approach that encourages teams to think holistically about the problem space using numbers to assess the impact of designs on objectives.

Value delivery is a combination of existing principles and practices from the Evo [Gilb05] method that can be used in conjunction with the Scrum method. It is not the only method to measure value, but I believe it is a method that works well with the Scrum method. I believe value delivery can help agile teams show measurable value delivered in alignment with organizational strategies quickly and effectively.

Today's engagement

To show value delivery in action, we'll use a slightly altered real-world case study. You and I are going to consult with a non-profit client and we're

Now that we've identified our stakeholders and resources (time and money), that just leaves defining the objectives

going to help them adapt the value delivery approach to their existing Scrum process.

Our client, a leading non-profit research organization, recently completed its 2009 strategic planning sessions. Senior management's business strategy is:

In 2009, we are embarking on a strategy to increase charitable giving through improvements to our web site. We believe this strategy will help increase our market share for online giving while positively impacting our key customers: non-profit organizations.

The organization currently uses Scrum for developing their web-based application. The Vice President of Marketing and business sponsor, Nancy, has personally asked us if we could help her and the organization:

1. Establish a set of strategic objectives so value can be measured and managed
2. Make smart funding decisions with web site improvements so budget and risk can be managed
3. Identify the improvements with the best 'bang for the buck' for doing first so quick progress can be demonstrated to everyone.

We need to ensure our work integrates nicely with the existing Scrum process used for web site development.

At our initial meeting, Nancy asks if we can take this job on. 'No problem' I tell her. Then she shrugs a bit, 'And being a non-profit, we can't pay much at all.' I hesitate for a second, then respond, 'If you can ensure we get access to the right people, and provide us with someone to organize the meetings, we'll do the project pro-bono!'

'Wonderful!' Nancy exclaims. 'If it's ok by you, let's get started the week after next. That'll give me some time to lineup the right stakeholders.'

So with that, you and I are off to do some valuable work for a worthy non-profit. Ready to get started?

Step 1: Identify stakeholders, objectives and resources

When faced with problems like these, I like to ask myself three questions:

- Who are my stakeholders?
- What are their objectives?
- What resources are available?

Since we're looking to make improvements at the organizational level, we're certain board of directors, CEO and executive management are all stakeholders. Their customers: non-profit organizations and for-profit organizations that donate to non-profits, are also stakeholders. Other internal stakeholders include operations, development, marketing and management.

Our first day on-site we conduct interviews with key stakeholders (up to the CEO) and spend quite some time with Nancy. We meet a diverse set of individuals in marketing, sales and IT who provide us background on their roles and how the strategy will likely impact them. We cast a wide net to ensure that we don't leave anyone out.

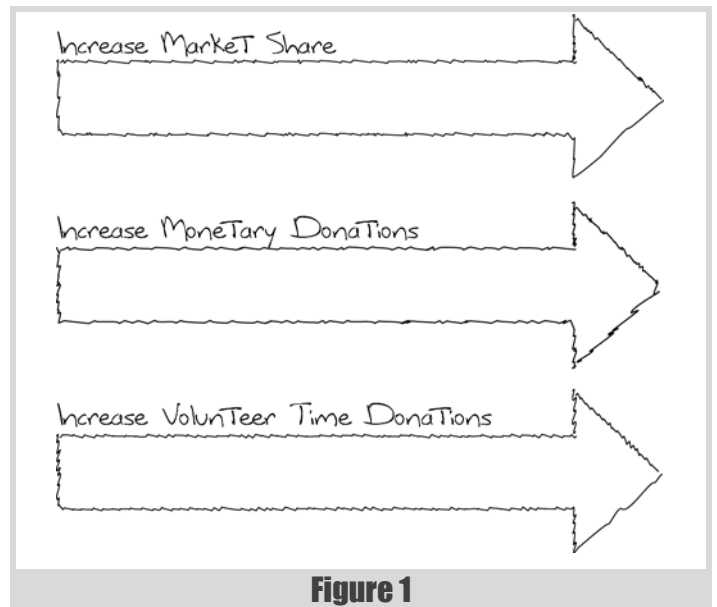


Figure 1

During our interview with Nancy, she says, 'The CEO recently agreed to provide \$1 million and 10 months for implementing the business strategy, but wants to see results quickly. My responsibility as business sponsor is to ensure this succeeds, but where do I start?'

After a bit more conversation, Nancy and I sit down together at the table and I continue, 'Now that we've identified our stakeholders and resources (time and money), that just leaves defining the objectives. This is by far the hardest question to answer, so let's take an iterative approach to creating our objectives. The first step is to identify each with a simple name.'

I turn my attention to a copy of the business strategy on the table, pull out my pen and underline the key themes I see:

In 2009, we are embarking on a strategy to increase charitable giving through improvements to our web site. We believe this strategy will help increase our market share for online giving while positively impacting our key customers: non-profit organizations.

After some further discussion with Nancy, we quickly identify the following objectives and write them on the whiteboard in the room (Figure 1).

The first two come straight from the strategy. Nancy tells us the last is a request from our non-profit customers. In addition to money, non-profits also value the time donated by volunteers to help them fulfill their missions. We decide these three objectives are enough to get started and provide the right focus for the team, so we capture these and move on.

Step 2: Quantify our objectives

With these objectives identified quickly, we're feeling pretty good about our progress. I say to Nancy, 'The next step is making them quantifiable.' Nancy, looking a bit puzzled, says, 'Why should our objectives be

It is through the process of trying to quantify objectives that we probe more deeply into what's really important

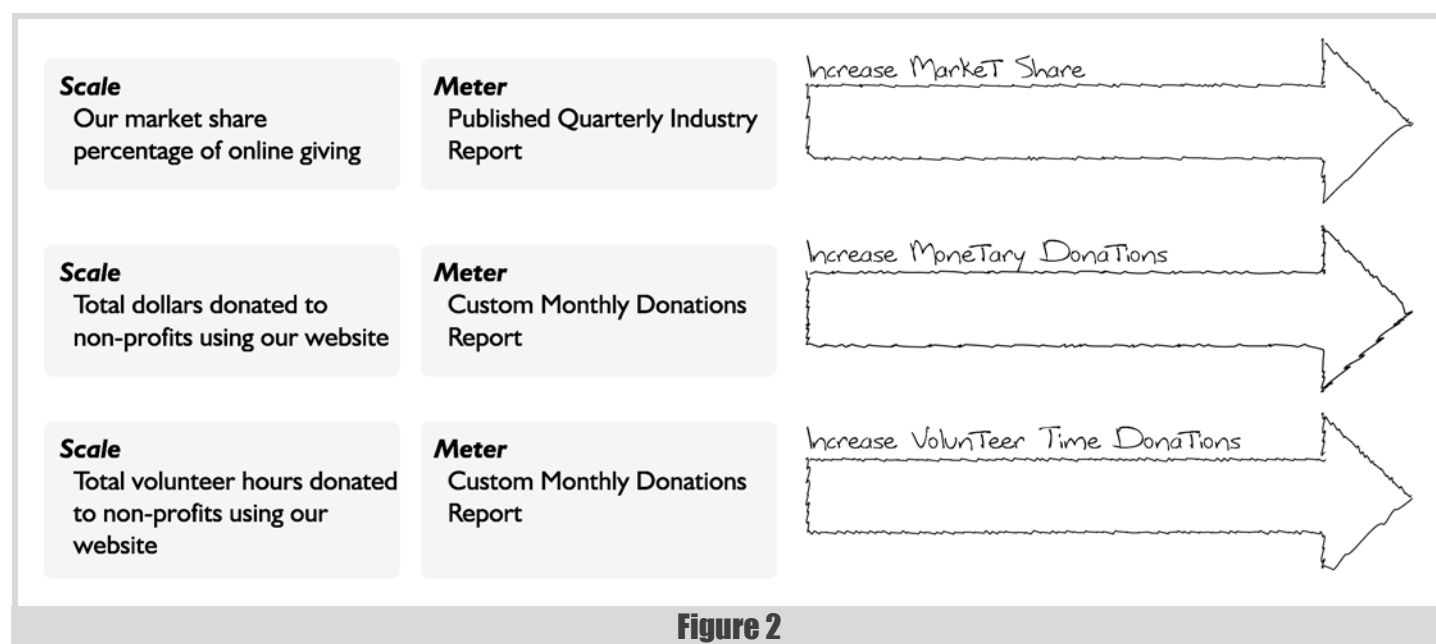


Figure 2

quantified?’ I respond that contrary to what she may think, ‘The main purpose of quantification isn’t to measure and track. The main purpose is to provide clarity in requirements. Tom Gilb [Gilb] says it best:’

The fact that we can set numeric objectives, and track them, is powerful; but in fact is not the main point. The main purpose of quantification is to force us to think deeply, and debate exactly, what we mean; so that others, later, cannot fail to understand us.

I continue, ‘It is through the process of trying to quantify objectives that we probe more deeply into what’s really important.’

Nancy responds, ‘But how do we do this? Remember, we don’t have much time to start showing progress. I’m not looking for an academic exercise, I need results!’

‘It’s OK’, I say. ‘There’s actually a way we can quantify these objectives pretty quickly using Planguage [Gilb05]. With our objectives identified, we’ll next add a Scale (*what to measure*) and then a Meter (*how to measure*).’

Nancy and I return to the board and update our objectives (Figure 2).

Step 3: Identify targets, constraints and benchmarks

Nancy is again happy with the progress but asks, ‘Now I see quantifiable objectives, but without knowing where we are today or going in the future, what use is this?’ She’s right.

‘Time to tell you about Targets, Constraints and Benchmarks!’ I respond.

Targets, as the name suggests, are the performance levels the team is striving to achieve. It’s the level of performance that, when reached, everyone agrees is success. Stakeholders agree to provide the necessary

resources to achieve these levels and technologists agree to design systems to meet these levels. Target levels are not simply edicts laid down by stakeholders absolutely. Rather, setting them requires collaboration and agreement from the implementation team to ensure the levels are achievable (with an estimate of what resources it may take to get there).

Constraints are the levels that must be avoided. In practice, these could be contracted Service Level Agreements (SLAs) identifying the minimal performance levels before penalties are assessed. They may also be the minimum levels needed to ship the product.

In our case, these are the levels below which senior management recognizes things didn’t go well (and perhaps bonuses would be impacted!). Just like targets, setting constraint levels requires collaboration from all parties.

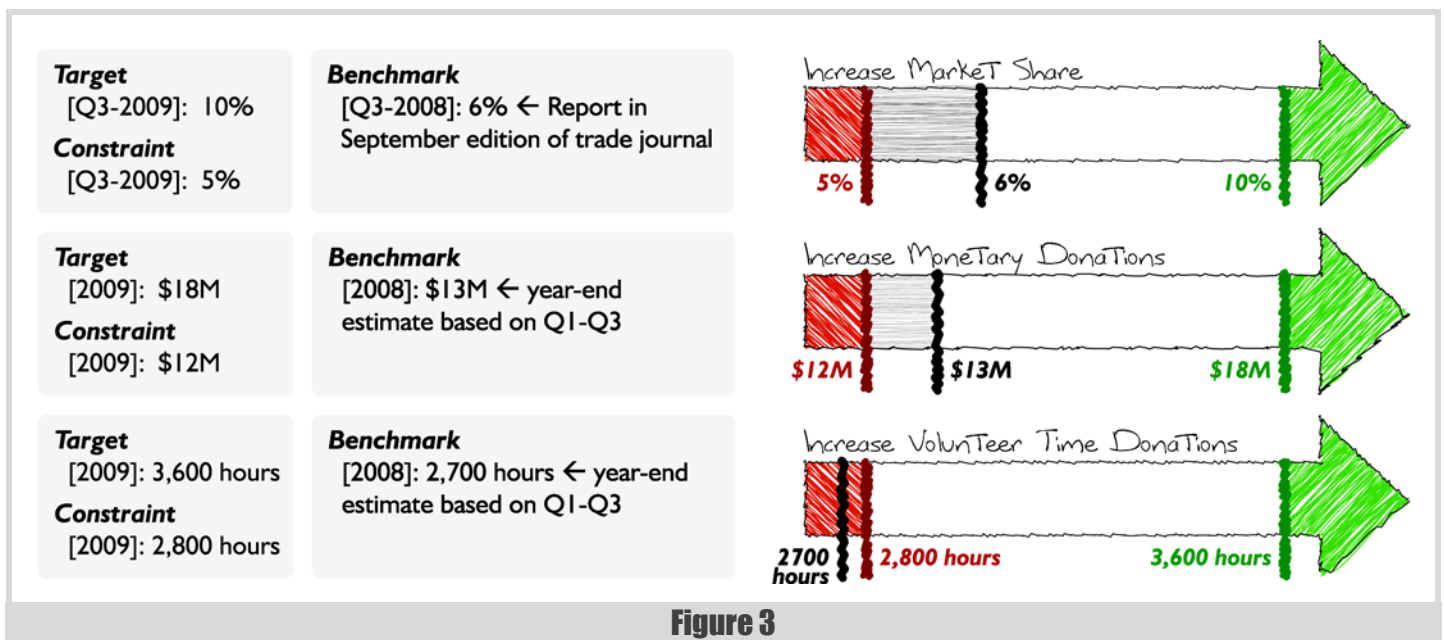
Finally, benchmarks are the levels achieved today or what’s been achieved in the past. Benchmarks enable an understanding of the current state and assessing how close (or far) we are from achieving the target levels of performance. In practice, it’s often easiest to start with identifying current benchmark(s) and using this to set appropriate target and constraint levels.

After listening Nancy responds, ‘Why do we need constraints? Can’t we just set targets?’

I remind her, ‘As important as setting levels for success is, it’s often more important to set levels for failure. Everyone in the project needs to understand clearly what’s success, what’s failure, and where the organization is today. With that understanding, we can begin honest discussions about what to do next.’

Nancy takes a guess at target and constraint levels and we do follow-up interviews with developers and testers to gather benchmark data. Pulling

A design idea is a potential solution that moves a team closer to achieving the stakeholder's objectives



this all together, we update our whiteboard and show Nancy our results the following morning (Figure 3).

As we walk Nancy through this, we also explain the last two Planguage concepts: Qualifiers and Sources.

Qualifiers are the variables in brackets [] and they help specify under what conditions the levels apply. Qualifiers are typically dates, places or events. Examples could include: 2008, Q1-2009, UK only or Release 3. These are user-defined and can be anything that makes sense for a particular situation.

Sources are the text to the right of the arrows ← and they help convey where information originated. Sources can be applied to any piece of information to add transparency, credibility and traceability.

We've introduced Nancy and the organization to a lot of new concepts, so let's briefly recap before moving forward:

- **Scale** – What's measured (units)
- **Meter** – How it's measured (method)
- **Targets** – Levels aiming to achieve
- **Constraints** – Levels trying to avoid
- **Benchmark** – Current or past performance levels
- **Qualifiers** – Dates, places or events useful for clarification
- **Sources** – Origin of information for transparency and credibility

I tell Nancy, 'Think about our progress right now. We have quantifiable objectives for our business strategy that will fit on a single PowerPoint slide and can be communicated and understood by all project team members and stakeholders, including the CEO!'

She responds, 'Wow, *now that's powerful!* I think we're ready to share this with the other senior managers and stakeholders, I'll set up a working session for this Friday so we can get validation before moving forward.'

Step 4: Brainstorm design ideas

During Friday's working session the business sponsor, product owners, leaders, analysts, developers, testers and ScrumMaster are all in attendance. Agreement is reached on the target and constraint levels we previously established with Nancy. With the team itching to start designing solutions, I explain 'Next comes the really fun part: creative brainstorming of design ideas.'

'What's a design idea?', Steve, one of the architects asks. I respond, 'A design idea is a potential solution that moves a team closer to achieving the stakeholder's objectives.'

In order to ensure the brainstorming is productive, I tell the team, 'The objectives have been established and validated, so now let's find the solutions. But I'd request that each of you focus the brainstorming session on finding design ideas that will:

1. Increase Market Share
2. Increase Monetary Donations
3. Increase Volunteer Time Donations

Good design ideas will positively impact one objective, but **great design ideas** will positively impact all three objectives with a single design idea!'

The team knows their current web site is pretty basic with no fancy Web 2.0 stuff. There's a basic search and the ability to view reports of non-profits. Users can also make donations directly to non-profits on the web site by clicking a 'Donate Now' button.

The process of identifying design ideas is a creative one that encourages out-of-the-box thinking engaging the entire team

Nancy offers up to the team, ‘The CEO is looking for ideas that would kick off an implementation project and are achievable within a \$1 million budget and 10 months. Those are the constraints of our brainstorming today.’

The process of identifying design ideas is a creative one that encourages out-of-the-box thinking engaging the entire team, not just executives and product managers. In practice, good ideas often build off one another. Teams that do this before starting their projects achieve a greater shared vision of what they’re being asked to deliver and understand the real measures of success that translate to value to the organization.

During the session lots of ideas are generated and captured. Some of the more interesting ones include:

1. Setup recurring payments for members so each month a donation is automatically made to the non-profits of their choice.
2. Create a Facebook application that integrates the web site to the 100+ million Facebook users so they can get connected to non-profits organizations of their interest.
3. Create ability for Non-Profits to upload images and videos of their charitable work to the site to help solicit donations.

Nancy says, ‘Based upon the team’s intuition, these design ideas sound like the best candidates. Let’s move forward.’

She turns to us and says, ‘Now, how do we get down to just one that we can do?’

Step 5: Select the next best design

‘Arm wrestling works well’ I quip.

‘Get serious’, responds Nancy.

‘OK, we could vote on each option using a secret ballot. Or we could just let the CEO pick. There are more options, but I think they would be irresponsible, given the time and energy we just spent creating measurable objectives.’

I continue to explain to Nancy and the team, ‘We should use Impact Estimation (IE) to help us with this problem. IE is a very simple tool for calculating cost/benefit of design ideas and identifying the one with the best return on investment or *bang for the buck*’.

To help Nancy and the team understand the concept, I draw Figure 4 on the board to show the structure of an IE table with objectives and resources as the rows and design ideas as the columns. The last row is the benefit to cost ratio (value delivered) which will help determine the best design idea. This is a simple ratio of the sum impact of objectives divided by the sum impact on resources (i.e. sum objectives impact / sum resources impact).

After a little explanation, it appears the team gets the gist of IE enough so we can get started.

It’s been a very productive session but I feel the energy slipping on Friday afternoon. ‘Let’s break for the week and next week we’ll start to fill in the impact estimation table with the help of the team’, I say. Everyone agrees and heads back to their desks and home for the weekend.

	Design Idea #1	Design Idea #2	Design Idea #3	Total Impacts
Objectives	Impact on Objective	Impact on Objective	Impact on Objective	Total Impact on Objective
Resources	Impact on Budget	Impact on Budget	Impact on Budget	Total Impact on Budget
Benefits to Cost Ratio	Ratio	Ratio	Ratio	

Figure 4

Nancy approaches us afterwards, ‘I really want to thank you, I have never seen our team identify so many good ideas so quickly. They were amazing! Because we had focus on our objectives, they really embraced the process and the result was great collaborative energy in the room. I’m anxious to see which design idea we end up doing!’

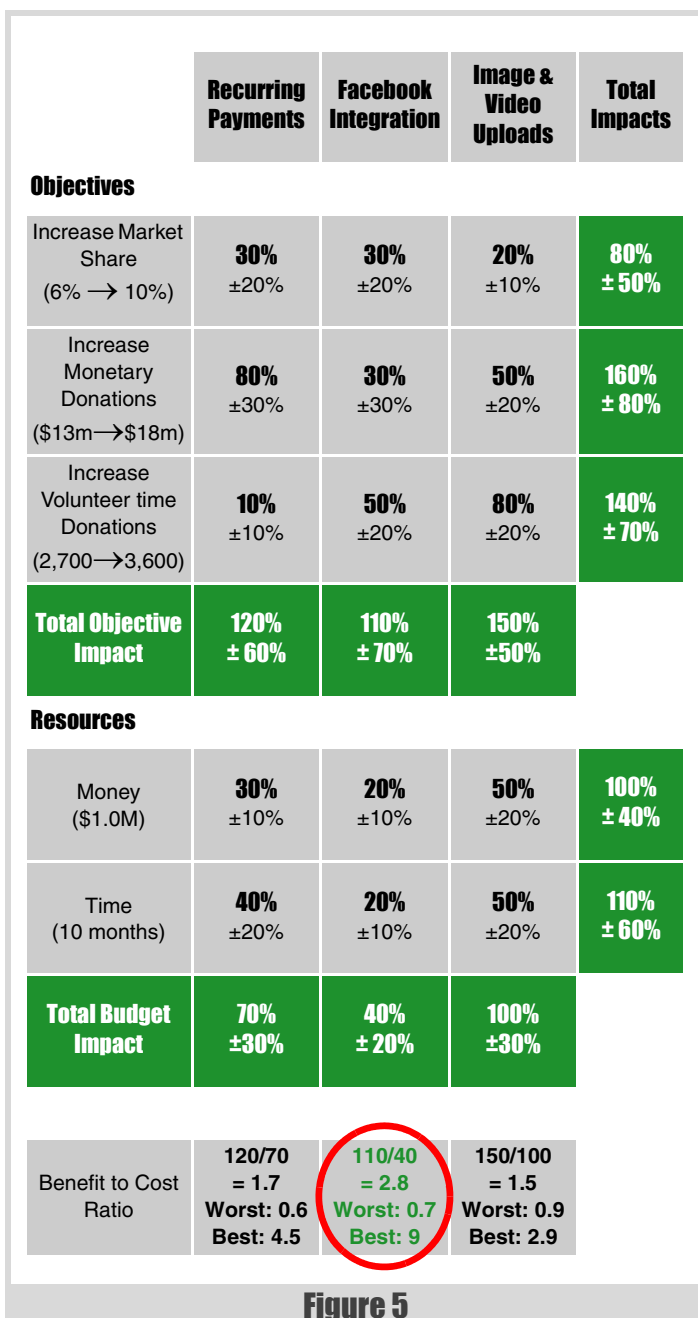
The following week the team meets again and puts the top three design ideas into the IE table and estimates each ones impact on objectives and budget. I encourage the team not to try to get too precise; instead simply try to make a first pass using the best information at their disposal. To show uncertainty, with each value best and worse case is provided using ± notation. At the end of the session the team has identified the Facebook Integration as the best design idea and circles it in the IE table (Figure 5).

In doing this exercise, I point out the following things to the team members, so they see how we arrived at our decision:

- **Recurring Payments** have the biggest impact on the Increase Monetary Donations objective, but relatively low impacts on the others. Its estimated cost is \$200K–\$400K (30% ± 10% of monetary budget) and its expected to take 2–6 months to complete (40% ± 20% of time budget).
- **Facebook Integration** has the lowest sum impact on objectives (110%) of any of the ideas, but its low cost (\$100K–\$300K and 1–4 months) means it has the best benefit / cost ratio of all design ideas. So it’s the idea we’ll go with first.
- **Image & Video Uploads** has the biggest sum impact on objectives of any of the design ideas, but it also costs the most, resulting in the lowest benefit/cost ratio.

These are the obvious observations, but there are other more subtle things that are also interesting. I present the following to the group for consideration:

If we're wrong, at least we'll know quickly and can change direction



- Similarly, if meeting the Increase Monetary Donations objective is the highest priority, then Recurring Payments must be one of the design ideas implemented because it alone gets us 80% there.
- The Facebook Integration design idea has the most uncertainty associated with it (widest range of benefit/cost ratios: 0.7 to 9). Although this looks like the best design idea now based on its benefit/cost ratio (2.8), more research may be required to bring this uncertainty down or get the stakeholders to consciously accept this level of risk before moving forward.
- Implementing all three design ideas would use 100% of the money budget and 110% of the time budget, meaning the organization would likely not have time to implement all three design ideas in the next 10 months.

Nancy steps into our team session at the end of the day and I explain to her that the team now has:

1. Quantified the impact of the three best design ideas against the objectives and budget
2. Determined the design idea that has the best benefit/cost ratio
3. Identified the risk and uncertainty associated with each design idea, prompting more research to reduce the uncertainty or acknowledging the risk comfort level and moving forward

The team's choice of which improvement to pursue first is now less guesswork and more fact-based using quantified data. The IE table easily explains to all stakeholders why the Facebook Integration is the best project to pursue and can back it up with real numbers including expected benefit and cost compared to the other design ideas.

Nancy agrees, 'This is perfect. I can now present the Facebook Integration project to the stakeholders and the CEO for approval. When they question why this project was chosen, and I know they will, I can show them how we measured the impact of the best design ideas against our objectives and budget.'

Nancy continues, 'I'll set up another working session for this Friday with the stakeholders and the team so we can get feedback and make a decision on the project choose so we can move forward.'

On Friday, Nancy presents the Facebook Integration idea as the first initiative the team will pursue. After some initial questions, the stakeholder's agree and give Nancy and the team the green light to forward.

The CEO explains to Nancy and the team, 'Although I realize we could spend more time reducing uncertainty and getting more refined estimates, I accept the level of risk in order to move forward and start seeing results!'

He continues, 'If this were a bigger project I'd expect more details, but what I've seen here today makes me comfortable the team has assessed our options well. If we're wrong, at least we'll know quickly and can change direction. Best should not get in the way of better. Let's go with the Facebook option and see how quickly we can impact those objectives!'

Later that afternoon the CEO drops by Nancy's office. He asks how things are going and congratulates her on today's working session. He's clearly

- Assuming the design ideas are mutually exclusive, implementing all three would not likely help the organization meet their Increase Market Share objective (Total Impact is 80%). If meeting this objective is the most critical, then more design ideas must be identified.

we now have a process for quickly identifying design ideas and assessing their impacts

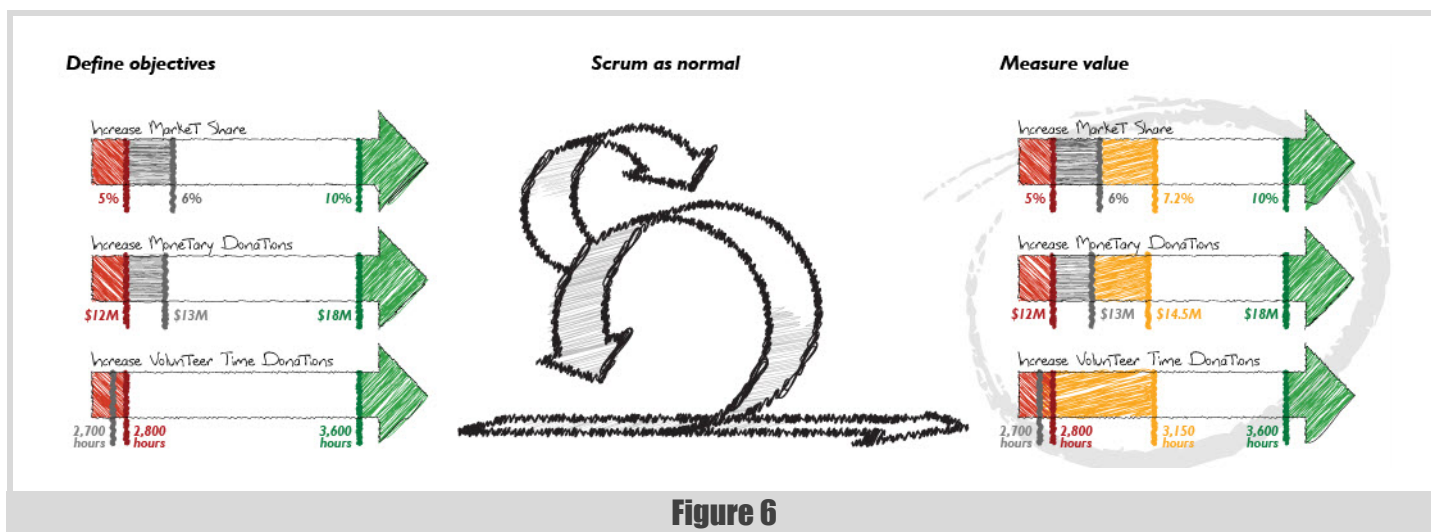


Figure 6

happy with how quickly the strategy and team are progressing and quips, ‘you’ve got a real team of movers and shakers working for you!’

The normally reserved Nancy can’t help but beam with pride. She, too, is proud of what her team has accomplished in two short weeks. She recaps for her boss that because of their efforts, the organization has:

1. Established a set of strategic objectives so each sprint the value delivered by teams can be measured and the project properly managed. Not just that, but the entire team is aligned with the objectives, something that’s never happened before.
2. Leveraged the Impact Estimation process to ensure smart funding decisions are based on quantified information. This allows budget to be managed and risk to be mitigated.
3. Identified the Facebook Integration as the best ‘bang for the buck’ and therefore the one that’ll be done next. Other design ideas will be evaluated and scheduled for future releases using the Impact Estimation process during planning.
4. Started to update the web site product backlog so the initial Facebook Integration features are rolled out in the next release of the web site in six weeks.

‘I tell you what. Take the team out after work today for a celebration. I think they deserve it.’ responds the CEO.

Over drinks at the local pub after work, the team celebrates their success and their growing camaraderie. Nancy comes up to me and says, ‘After going through this process I’m very confident we’re focused on the right problem. Like I told my boss, even if the Facebook option isn’t the best choice, we now have a process for quickly identifying design ideas and assessing their impacts. We’re focused on the results and if this one doesn’t work, we can quickly find the right ideas and make them happen.’

But then she pauses, ‘Before we celebrate too much, we still need to ensure we can do this project using our Scrum method.’

‘Don’t worry’, I say. ‘Scrum will fit right into this process. In fact, we’re not going to change Scrum at all. We’re simply going to do a few steps before and after each sprint. It’ll have a minimum disruption on the team, you have my assurances.’

‘OK, I’ll trust you on this.’ Nancy responds. Then, after a few seconds, she says, ‘But I’m not too worried. Our team is accustomed to delivering projects using Scrum, so long as you can items prioritized and into their backlog, we should be good.’

Step 6: Agile integration

Now that Nancy and her team are focused on right problem, it’s time for them to solve the problem right. This is where an agile method like Scrum comes in. Some of the benefits of using agile to implement design ideas include:

1. Short iterations and inspect-and-adapt philosophy get us working solutions quicker and reduce our delivery risk
2. Collaborative nature of cross-functional teams with feedback from stakeholders.
3. Team communication and collaboration creates a positive, supportive work environment where teams feel ownership of the process.

The ways in which agile feels different with the value delivery approach include:

1. Incremental funding is based on real value delivered. If, after a few iterations, teams are not making progress on the prioritized objectives, stakeholder’s can call ‘stop’ and pursue a different design idea with their remaining budget.
2. Entire teams are aligned to the business value of the project and know how success is measured.
3. In each sprint, progress towards objectives is measured to show the value delivered in the sprint, as shown in Figure 6.

Teams can still develop user stories for functional requirements and prioritize them in the backlog as normal

The value delivery approach rewards teams for delivering value, not just building features. It raises the level of measurement up from feature-centric velocity to value-centric objectives. And because these objectives are aligned to key business strategies, we're ensuring we're not just solving the problem right, we're solving the right problem.

The value delivery approach integrates with Scrum's product backlog in three ways:

1. Teams can still develop user stories for functional requirements and prioritize them in the backlog as normal. User stories are useful for documenting 'what' the system will do. User stories should be prioritized by business value, meaning how they help the organization make progress towards their prioritized objectives. User stories use the following format:

As a <user>, I would like to <do some function>, so that I can <achieve some goal>

2. Teams should also prioritize quality requirements (aka non-functional requirements) alongside functional requirements. These are useful for ensuring critical system qualities such as availability, scalability, usability, response time, etc. can be met. Quality requirements should be defined using the Planguage concepts introduced earlier including Scale, Meter, Target, Constraints and Benchmarks so expectations are clearly stated (with numbers) about how 'well' the system must perform. Because meeting target levels for quality requirements often requires multiple rounds of improvement, quality stories can help ensure:

As a <stakeholder>, I would like to improve <some quality dimension> from <current level> to <desired level>, so that I can <achieve some goal>

3. Both user stories and quality stories are specified such that teams can use story point estimates to assess their relative implementation effort. This helps communicate to the product owner what can be done each sprint and how long it will take to get meet specific objectives.

Value delivery uses traditional Scrum activities such as release planning, sprint planning, story point estimates, burn-down charts and stand-up meetings all without modification. In practice this minimizes the change curve and ensures existing processes that are working aren't disrupted.

The difference is that in addition to the normal end-of-sprint activities such as demos and retrospectives, value delivery teams report the value delivered (progress achieved that sprint or release towards stakeholder objectives). While every sprint may not result in measurable business value delivered, making this reporting part of the process ensures everyone stays focused on what's really important and not focused on simply building features. It also helps focus the release planning process on delivering measurable value if product owners and teams know they will be asked to show the value they are delivering with each release.

Value delivery adoption

Just like Nancy and her team, organizations need some guidance with adopting value delivery. Adoption is ideally done from the start of new projects. However, because value delivery doesn't change the mechanics of functioning agile teams, organizations can transition to value delivery on existing projects. The value measuring steps before and after each sprint will hopefully change the team's focus, but shouldn't interrupt their natural team dynamics.

Although this approach is fairly new, I hope it will slowly start to gain practice and will evolve with feedback from the community. I have noticed it does follow an emerging pattern of combining practices from multiple methods to achieve better results. Scrum co-inventor, Jeff Sutherland, in 2008 published the positive effects of combining CMMI + Scrum over implementing pure Scrum alone [Sutherland08]. I believe value delivery is on this same trend, seeking to get better results by complimenting Scrum with established practices from the Evo method.

If you would like to try value delivery on your next project, in addition to this article there are free tools to assist you available on theagileengineer.com. I am actively recruiting individuals who would like support using value delivery on their next project. Please contact me for guidance and coaching on applying these concepts in your organization.

Case studies

The value delivery method described here is essentially selected practices from the Evo method combined with the Scrum method. Both of these methods are well tested and have large numbers of case studies behind them (Evo starting in the 1970s and Scrum starting in the 1990s).

In 2007, I started applying the value delivery approach on projects ranging from custom software development to Microsoft SharePoint implementations. As of 2008, three clients in the United States have used the method and all reported they enjoyed the focus they achieved on business value and reported their teams felt more aligned to stakeholder objectives. However, publishable case studies have yet to be done with the value delivery method, although that remains a goal.

In putting this approach into practice, I observed the following hurdles to adoption:

1. Introducing the concepts of measurable value may be very challenging if an organization is struggling to get the basic Scrum process working and think their issues are with Scrum adoption.
2. When Scrum adoption is a 'bottom-up' approach, developers will often not appreciate the concepts of measurable value delivered (at least not initially).
3. Progressive business leaders or experienced agile coaches who recognize the value this brings will be required to help drive adoption and maintain focus.
4. If you don't have support from business stakeholders to continually measure value, teams won't think it's important to measure their results.

Are we delivering the right things, now?

5. If business stakeholders want to 'place an order with IT' and not be engaged in defining objectives and measuring value, adoption will be practically impossible.
6. Making value delivery stick requires discipline in not only creating the initial objectives, but also following through with actual results and using these in project management. Organizations can start with value delivery, but unless management is committed to asking for the value delivered from their investments, interest can quickly fade.

Others are starting to integrate Evo and Scrum, but as of yet have no published results. Recently, Jens Egil Evensen of Norway has been reporting success using Evo with Scrum in a method he calls 'Avegility' for his customer's projects. Evo is used for project management and to prioritize the backlog, Scrum is used to develop the software and Planguage is leveraged to write the requirements. [Evensen09]

Kai Gilb is teaching and coaching management and Scrum product owners at an international organization how to define value-results, and how to link the business, stakeholder, product and solution value-results to product backlog items. He reports, 'This process enables them to create a product backlog that is optimized and justified all the way through from product values to stakeholder values all the way up to business values. Everything the Scrum development team does is justified all the way up the value chain, it gives early, frequent, measurable, highly leverage benefits at all levels. Management can manage the value creation, and Scrum developers can deliver it.' [Gilb08]

Summary

To return to our challenge, 'Are we delivering the right things, now?' hopefully you can see there is an approach for answering 'yes' with confidence that's lightweight and agile! Nancy has learned it and so has our team. For our non-profit client, the value delivery approach:

1. Defined what 'delivering value' means within their teams
2. Measures value as progress towards stakeholder objectives
3. Prioritizes design ideas according to ROI
4. Integrates with their existing Scrum method

In summary, value delivery's philosophy is that teams and organizations should use whatever agile method they prefer for delivering things right. But they must also focus on delivering the right things with a laser focus on business value. Only by covering both of these perspectives will organizations ensure their outcomes are results driven and not feature driven. ■

Thanks to Chris Allport, Kai Gilb and Jimmy Chou for early feedback.

References

- [Evensen09] Evensen, J. E. (2009) Based on private email conversation
 [Gilb] Gilb, Tom, www.gilb.com
 [Gilb05] Gilb, T. (2005) *Competitive Engineering*, Oxford: Elsevier Butterworth-Heinemann
 [Gilb08] Gilb, K. (2008) Based on private email conversation.

[Sutherland08] Sutherland, J., Jakobsen, C., Johnson, K. (2008) Scrum and CMMI Level 5: The Magic Potion for Code Warriors. Retrieved 14 Jan 2009 from <http://jeffsutherland.com/scrum/Sutherland-ScrumCMMI6pages.pdf>

[VersionOne09] VersionOne, '3rd Annual State of Agile Development Survey', retrieved 11 Jan 2009 from <http://www.versionone.com/agilesurvey>



C++ Libraries and Tools to Simplify Your Life

- > **POCO C++ Libraries:** free open source libraries for internet-age cross-platform C++ applications
- > **POCO Remoting:** the easiest way to distributed objects and SOAP/WSDL Web Services in C++
- > **POCO Open Service Platform:** create high performance component-based, manageable and dynamically extensible applications in C++
- > and much more: Fast Infoset, WebWidgets, ...
- > available on many platforms – highly portable code
- > scalable from embedded to enterprise applications



appliedinformatics

Free Download and Evaluation: appinf.com/simplify