CALCULATING ROI ON YOUR INVESTMENT

In requirements management tools

August 2006

Richard Denney

SERENA™

TABLE OF CONTENTS

Calculating ROI on Your Investment in Requirements Management Tools

SUMMARY

There's no question that a commercial Requirements Management tool is very useful; but, can it pay for itself at your company? In this article we'll look at a model to help you calculate ROI on Requirements Management tools.

OVERVIEW OF ROI

Return On Investment (ROI) is a popular method of measuring the success of process improvements and IT investments. It is a measure of the dollars returned on dollars invested. As Jeffery E. Payne points out in his article ''Quality Meets the CEO: How to Get Management Buy-in,'' ROI is an effective approach for arguing the need for, or demonstrating the success of, process improvements and IT investments.

Though there are a number of methods of calculating ROI, one straightforward, simple to understand method is the Benefit to Cost Ratio, which is simply dividing the benefits in dollars of process improvement or IT investment by the costs. Benefit to Cost Ratio = Benefits/Cost. So, a Benefits to Cost Ratio of say, three, would mean that for each $1 spent on the cost of process change and IT, $3 in benefits were realized.

In doing an ROI assessment, typical sources of cost include:

- IT investments, including ongoing maintenance
- Training staff in new processes and IT tools
- Consulting needed to assist process change and IT installation
- Recurring cost associated with new process and IT

Typical benefits that are considered in an ROI assessment include:

- Increased revenue, e.g. increased sales, or sales margins
- Retention of sales that would otherwise have been lost
- Reduction in operating expense, e.g. daily time savings, eliminated rework

Let's look at how to do a Benefit to Cost model for process change and IT investment of putting a requirements management tool in place at your company. Though this model was developed in conjunction with the rollout of a commercial tool, it should be readily adaptable to development and rollout of ''home grown'' tools.

REQUIREMENTS MANAGEMENT TOOLS

Requirements Management tools are to requirements what Defect Tracking Tools are to defects. They provide an environment for, and database approach to, managing large numbers of requirements related in potentially complex ways (traceability). Requirements tools provide continuity through time, across projects, through staff changes, and through company re-organizations. They are a corporate memory for requirements. In a project, setting a Requirements Management tool is used in a number of contexts:

- Planning the scope of a release, or a series of releases (key in iterative, incremental projects)
- Managing plan execution: who is responsible for what, when
- Tracking project status
- Change control of scope, particularly in evaluating the impact of proposed changes (this is where traceability of requirements is critical).

For a large company, dealing with thousands of requirements, moving to a database approach to managing requirements, allows a metrics-based management of requirements that is simply not practical with paper document-oriented approaches.

CALCULATING THE ROI

No question that a requirements management tool is very useful, but can it pay for itself at your company? In his paper *Calculating Your Return on Investment from More Effective Requirements Management*, Dean Leffingwell advances this model for estimating ROI (numbers are from his example):

Start by calculating the total staffing cost of a typical project at your company	$2,400,000
Then use a ballpark number based on the industry average of % total project cost that is typically rework cost	30%
For this example, cost of rework	$ 720,000
Next use a ballpark number based on the industry average of the % of rework that is typically requirements related	70%
For this example, cost of requirements-related work	$ 504,000

One can then make an argument that if the requirements management process was improved by say 10%, that is a cost savings of $504,000 x 10% = $50,400. If the cost of process improvement (e.g. training), plus tool purchase was a total of $19,900 (again, numbers from his example) then the Cost to Benefit Ratio is $50,400 / $19,900 = 2.53.

The model I will present here is a bit different from Leffingwell's; it will provide specific improvements one might expect from installing a requirements management tool. I will then try to quantify the benefits from that perspective. The model is based on one I developed for a company of about 500 R&D staff, 1000 employees overall. The tool, a commercially available product, was deployed in four cities. The ROI assessment was done about 1.5 years into the rollout.

ASSUMPTIONS AND CONVENTIONS

We begin by making a number of working assumptions and conventions that will be used throughout the ROI model. The model is developed in an Excel spreadsheet.

- Cells with black text are ones for which you must input values that make sense for your company.
- Cells with red text are ones with values computed via some formula.
- Cells that are gray filled have values that are subjective.

These fields are the main source of uncertainty in the model. In interviewing staff as part of the rollout, the values for these fields varied widely.

We begin by calculating the cost of a fully burdened employee, per work day, per hour, and per minute. By "fully burdened" we simply mean the total cost to the company: salary + total benefits. As a rule of thumb, the fully burdened cost is usually about 1.5 times the salary.

$120,000	Average fully burdened cost per employee per year
230	Work days per year
$ 522	Employee cost per day
$ 65	per hour
$ 1.1	per minute

Next, we capture the actuals in terms of numbers of requirements that were entered into the requirements management tool as of the date of the ROI assessment; about 1.5 years after initial rollout. Of the approximately 5000 requirements entered, about 1000 had already been implemented and shipped as part of some project, and some 200 had been rejected, meaning a decision had been made that these requirements would never be addressed in any future release.

5,000	Requirements entered to date
1,000	Requirements implemented thus far
200	Requirements rejected
3,800	Requirements remaining open for future projects

THE COST

An ROI assessment must be done for a fixed period of time; both the cost and the benefits must be calculated for the same fixed period. This study was done about 1.5 years into the rollout of the new requirements management process and tool.

Consulting cost was for onsite support beyond classes, i.e. additional support that was purchased to aid in the rollout of the tool and process. The Rollout Team was a core set of staff that spent a portion of their working time in support of the process and tool rollout. This number is probably a bit low as it doesn't account for travel expenses.

Software	**$ 45,000**	Server and client licenses, plus maintenance to date
Training	7	Number of training classes
	$ 8,100	Cost per class including instructor airfare, hotel, meals, car, etc.
	15	Number of students per class
	2	Duration of class in days
	522	Fully burdened cost per day per employee
	$ 23,752	Total per class
	$166,265	Total for all classes
Consulting	**$ 25,000**	On-site support beyond training classes
Rollout Team time	1.50	Elapsed years of rollout to date
	1.00	Number of staff working at 25% time on the rollout
	3.00	Number of staff working at 10% time on the rollout
	0.83	Staff years of effort
	$ 99,000	Total cost for rollout team

Another cost we account for is *tool use overhead*. Just as the proper entry of a defect into a defect tracking tool requires some time, the proper entry of requirements into a requirements management tool requires more time than the capture of requirements on the back of a napkin, or excel spreadsheet. As noted above, the values entered in the gray cells are highly subjective.

Tool Use Overhead		
	5	Overhead (staff minutes) per requirement to use the tool beyond what would be done in say excel, word, etc.
	5,000	Requirements entered to date
	25,000	Staff minutes
	$ 27,174	Total

Finally, we account for the added rigor in requirements management that we are asking teams to make. For example, requirements that would otherwise not be recorded, or recorded on a white-board in an office, are now entered into a public record. This increase in easy-to-access requirements leads to additional review, discussion, test planning, and change control. The cost of doing the job right is, nevertheless, a cost, and is captured in the ROI model here.

In Ed Weller's *Calculating the Economics of Inspections* (StickyMinds, Jan 2002) he states the recommended rate for preparation and inspection of a requirements specification is about seven pages an hour. This seems a reasonable heuristic for estimating the *average* cost of additional review and rigor on requirements that were actually implemented. The requirements that were entered, but not implemented, receive much less scrutiny, here modeled as one-tenth the effort, or a factor of 10 increase in speed at which they are reviewed. For the model we will assume 1 requirement = 1 page.

Added Review and Rigor	
1,000	Requirements implemented
7	Requirements per hour per person
5	Number of persons on a team that use a requirement for direction in their work to plan, develop, test, develop user documentation and training materials, etc.
714	Staff hours
$ 46,584	
4,000	Requirements entered, but not implemented
70	Requirements per hour per person (requirements not currently being implemented receive 1/10th the scrutiny)
5	Number of persons on a team that use a requirement for direction in their work to plan, develop, test, develop user documentation and training materials, etc.
286	Staff hours
$ 18,634	
$ 65,217	Total for added review and rigor

BENEFITS

Let's move to calculating the benefits of the requirements management tool. Of the two broad categories of benefits discussed above, increased sales and reduced operating costs, this one will focus solely on reduced operating expenses for the company. This is fairly common for most IT investment ROI models. In the case of a requirements management tool one could probably make an argument that increased sales, or at least retention of sales are realized due to increased customer satisfaction resulting from better requirements management.

COST SAVINGS FROM STAFF WORKING MORE EFFICIENTLY

We begin by calculating the savings realized from staff on projects having a readily available, always-up-to-date, common source of requirements upon which they can base their work. It is a cost reduction from staff working more efficiently to plan, develop, test, document, and develop training materials for a product.

Scenarios where inefficiencies occur when requirements are not readily available to staff include:

- Requirements exist only in Joan's head, so each and every staff member that needs that information to plan tests or write a user manual makes a trip to Joan's office
- Requirements are recorded on Joe's laptop, but Joe is at the client site for the next two weeks
- Bob writes the user manual based on the woefully outdated hardcopy requirements specification document he has on his desk, requiring significant rework later
- Requirements kept on Sue's laptop are lost when laptop is stolen at the airport!

Note that this part of the model is not about doing a better job; *it's about doing the same job more efficiently.*

1,000	Requirements implemented
5	Number of persons on a team that use a requirement for direction in their work
5	Time saved in minutes, per person, per requirement, because they have a documented requirement, readily available in a company-wide repository upon which they can base their work
25,000	Total minutes of saving for all staff
$ 27,174	**Total $ savings**

AVOIDING THE COST OF LOST REQUIREMENTS

Staff churn is a common problem for projects. Staff quit the company; staffs move to other projects or are promoted; whole projects get restaffed through company reorganization. When requirements are not recorded and managed in a companywide system, they are subject to loss due to staff churn. The loss means that the requirements must be ''rediscovered'' and re-engineered again and again. Human Resource groups report that 12 percent staff churn per year is not uncommon. Requirements kept on laptops are subject to loss when the laptop is stolen at the airport!

5,000	Requirements
1,000	Already implemented so loss is not a problem
4,000	Pending implementation and subject to being lost
12%	Percent staff churn… assume X% churn of staff causes X% churn of requirements
480	Estimated requirements that fall through the cracks and are lost due to staff churn and other causes
3	Staff days to re-engineer each lost requirement… this could be a huge effort for some requirements
1,440	Total staff days of work re-engineering lost requirements
$751,304	**Total $ savings**

AVOIDING COST OF UNNECESSARY DEVELOPMENT

One of the benefits of a companywide requirements management tool is the increased visibility that requirements receive. On Projects employees are able to see what each other is doing; redundancy is spotted; priorities are better managed. The result: requirements get rejected. This leads to cost savings in avoiding unnecessary work.

200	Requirements that were rejected
25%	Percent of these, which had they not been recorded and subsequently rejected, may have gone forward
50	Requirements that may have gotten implemented
20	Total staff days spent implementing the requirement: coding, testing documentation, etc.
1,000	Total staff days for all unstopped features
$ 521,739	**Total $ savings**

REDUCING THE COST OF REQUIREMENTS RELATED DEFECTS

Finally, let's look at the cost savings in terms of fixing requirements-related defects. For this part of the model we'll use a few concepts also used in doing inspection ROI assessments, e.g. Ed Weller's ''Calculating the Economics of Inspections'' (StickyMinds, Jan 2002).

We'll do this by:

- Building a baseline model of what we believe was the cost of defect detection and removal *before* we added the new process and tool
- Recalibrate the baseline model with improved defect detection and removal, the cost of which we paid for as *Added Review & Rigor* above
- Our cost saving is cost of baseline-1 subtract cost of baseline-2

It's almost an industry cliché that the later a defect is found in the development lifecycle, the higher the cost to fix. The relative cost in the increase varies from industry to industry. Here, I use a simple three phase defect removal model:

- Removal of requirements defects anywhere before coding (peer review, analysis)
- Removal of defects once they are committed to code (unit test, system test)
- Removal of defects once they are released to the customer

The relative cost of fixing defects will be 1 staff day, 5 staff days, and 25 staff days; an increase of a factor of 5 from phase to phase, well within the industry averages that are cited in the literature.

Now let's look at the model. We start by estimating the number of requirements-related defects. Unless we are willing to acknowledge the possibility of a perfect requirement, it's safe to assume that all of the defects have at *least* one defect. If you are not comfortable with that assumption, adjust the percent as needed.

1,000	Requirements implemented
100%	Percent of requirements that initially have at least one defect
1,000	Initial number of requirements defects

Next, we estimate the number of requirements related defects removed prior to commitment to code, and the associated cost. A removal effectiveness of 50% means that of the total population of defects, 50% were caught and fixed. We'll assume that on average a defect at this stage can be found and fixed in 1 staff day. Changes that don't involve code are simply cheaper to make.

50%	Removal effectiveness for this stage based on ballpark industry average
500	Requirements related defects removed
1	Staff days expended to find and fix one defect at this stage
$260,870	**Cost of defect removal before defects are committed to code**

Next, we estimate the number of requirements related defects removed from the code itself (e.g. unit, integration and system test) and the associated cost. The calculation starts with the number of defects that remain undetected and unfixed from the previous stage. Because we are now dealing with code, the cost of finding and fixing a defect rises from 1 staff day per defect to 5.

500	Defects remaining that slip through to code
80%	Removal effectiveness for this stage based on ballpark industry average
400	Requirements related defects removed
5	Staff days expended to find and fix one defect at this stage
$1,043,478	**Cost of defect removal from code, prior to commercial release**

At this point let's review the total removal effectiveness of the first two stages.

1,000	Total number of requirements defects initially
900	Total number of defects removed by previous two phases before release
90%	Combined defect removal effectiveness of previous stages

Finally, the cost of defects shipped with the product to the customer. At this stage "finding" the bug is not so much a factor in the cost; the customer does that for you! Here, the cost of defects is determined by factors such as customer support calls, loss of sales from unhappy customers, and, of course the increased cost to patch software in the field. The cost of defects at this point will vary greatly depending on the industry, safety-critical products being an example where the cost can be very high.

100	Defects remaining that slip through to the customer
25	Staff days expended to support and fix one defect at this stage
$1,304,348	**Cost to support and fix remaining defects in the field**

Our baseline cost for finding, supporting, and fixing requirements-based defects is $2,608,696. Then we rerun the baseline, but with a 10% increase in defect detection and removal at each of the first two stages. This is the added review and rigor we paid for previously (see above under costs).

1,000	Requirements implemented
100%	Percent of requirements that initially have at least one defect
1,000	Initial number of requirements defects
10% increase 55%	Removal effectiveness for this stage based on ballpark industry average
550	Requirements related defects removed
1	Staff days expended to find and fix one defect at this stage
$286,957	Cost of defect removal before defects are committed to code
450	Defects remaining that slip through to code
10% increase 88%	Removal effectiveness for this stage based on ballpark industry average
396	Requirements related defects removed
5	Staff days expended to find and fix one defect at this stage
$1,033,043	Cost of defect removal from code, prior to commercial release
1,000	Total number of requirements defects initially
946	Total number of defects removed by previous two phases before release
95%	Combined defect removal effectiveness of previous stages
54	Defects remaining that slip through to the customer
25	Staff days expended to support and fix one defect at this stage
$704,348	Cost to support and fix remaining defects in the field
$2,024,348	**New cost**

Our new cost is $2,024,348; a cost savings of $584,348.

BOTTOM LINE

All that is left is to tally the Benefit to Cost Ratio, and as they say, ''your mileage may vary.''

Estimated savings: $1,884,565

Estimated cost: $427,657

Benefit to Cost Ratio: 4.4 to 1

Recall, that cost at this point, includes one-time expenses such as initial software purchase, and initial on-site consulting and training. The ratio should get even better once those one-time expenses are out of the way.

SUMMARY

As noted previously, cells that are gray filled are ones where the values entered are highly subjective, and vary depending on your company and industry. I found that a useful approach to using this model in talking with people is to have them provide values for these cells, kind of a Wideband Delphi approach. In that way they can see if the values they are willing to buy into result in a good benefit to cost ratio. After you've talked with a number of people, use their responses to build a minimum and maximum version of the model. If you want to get a bit more sophisticated, you can even run the model with a Monte Carlo simulation tool. This would provide a probability distribution function of the Benefit to Cost Ratio.

And while this model was developed to demonstrate ROI after the fact, the ideal use of such a model is to help in selling the process change in the first place, and to forecast that point at which benefits would begin to outweigh the costs of the effort.

WANT TO LEARN MORE?

David Gelperin, *Maybe We Shouldn't Write Requirements*, StickyMinds Original Article, October 2002.

A brief discussion-and lot's of reader comments-on how best to record and manage requirements: paper documents vs. spreadsheets vs. database etc.

Jeffery E. Payne, *Quality Meets the CEO: How to Get Management Buy-in*, STQE magazine, May/June 1999.

David F. Rico, *How to Estimate ROI for Inspections, PSPsm, TSPsm, SW-CMM®, ISO 9000, and CMMIsm*, StickyMinds, Sept 2002.

Ed Weller, *Calculating the Economics of Inspections*, StickyMinds, Jan 2002.

Dean Leffingwell, ''*Calculating Your Return on Investment from More Effective Requirements Management*'' http://www.rational.com/media/whitepapers/roi1.pdf.

ABOUT THE AUTHOR

Richard Denney, based in Austin, Texas, is the author of *Succeeding with Use Cases: Working Smart to Deliver Quality*, part of Addison-Wesley Publishing's Object-Technology Series. See www.software-quality-consulting.com for details on his book. Richard has 25+ years experience in software development and process management including as a principal in software process improvements for the oil industry's two largest suppliers of software solutions and several Fortune 500 companies.

This article is copied courtesy of Richard Denney and was originally published as a Software Quality Engineering StickyMinds "Original Article," June 2003, and in IT Management News, August 2003.

ABOUT SERENA

Serena Software, the Change Governance™ leader, helps more than 15,000 organizations around the world—including 96 of the Fortune 100 and 90 of the Global 100—turn change into a business advantage. Serena is headquartered in San Mateo, California, and has offices throughout the U.S., Europe, and Asia Pacific.

CONTACT

Learn more about the enterprise-wide power of Serena products by visiting www.serena.com or contacting one of our sales representatives in your area.

Serena Worldwide Headquarters
Serena Software, Inc.
Corporate Offices
2755 Campus Drive
Third Floor
San Mateo, California 94403-2538
United States

800.457.3736 T
650.522.6699 F
info@serena.com

Serena European Headquarters
Serena Software Europe Ltd.
Hertfordshire
Abbey View Everard Close
St. Albans
Hertfordshire AL1 2PS
United Kingdom

+44 (0)800.328.0243 T
+44 (0)1727.869.804 F
ukinfo@serena.com

Serena Asia Pacific Headquarters
Serena Software Pte Ltd
360 Orchard Road
#12-10
International Building
Singapore 238869

+65 6834.9880 T
+65 6836.3119 F
apinfo@serena.com

SERENA™