Building a Business Case for Detailed Specifications

David Gelperin
LiveSpecs Software

This paper describes a cost-benefit analysis of using detailed product specifications. The presentation begins with assumptions about the requirements process and then provides a formula for the Return on Investment (ROI) in moving from one specification strategy to another. We demonstrate the formula by comparing the minimal specing in Extreme Programming (XP) to detailed specing using the ClearSpecs Requirements techniques [www.livespecs.com].

Assumptions about the requirements process:

1. Requirements development is an instance of goal-directed human communication. Requirements information is communicated from *requirements suppliers* (e.g., customers, users, subject matter experts, conformance authorities, designers, testers) to *requirements users* (e.g., project estimators, user interface designers, product architects and developers, acceptance & system testers, technical writers & trainers).

2. Requirements users use the information to perform various tasks that are central to their responsibilities. The quantity and quality of the communicated information is a critical enabler of user efficiency and effectiveness.

3. Detailed specifications enable more accurate schedule and cost estimates, more effective user interface designs and product architectures, preventative test design, and can save significant requirements-based failure and rework costs caused by ambiguity.

Based on these assumptions, the goal of requirements development is to provide requirements users with task-adequate, accurate, and easily-accessible information.

To calculate the benefits and costs of a specification approach, we use:

Operational benefits –

Value of Automation ($V_{Automation}$) -- refers to the enabling of automatic support processes such as verification, simulation, and generation. Generation includes tests, documentation, interface designs, and implementation (e.g., loading business rule engines). This value is zero unless such automation is actually used.

Value of Reuse (V_{Reuse}) -- refers to the financial impact of specification reuse both within and outside the project, modified by the likelihood of that reuse.

Operational costs –

Cost of Coping (C_{Coping}) -- requirements users provided with inaccurate, incomplete, inconsistent, or poorly organized information cope with these shortcomings by exerting extra effort. They seek and communicate with information sources and translate the information into new forms. This leads to seeking answers to requirements questions and review of the recast information. Coping costs only include the extra effort resulting from spec shortcomings, not the entire cost of the follow-on activity (e.g., test). Coping is often uncoordinated and therefore there is a significant coping cost for each user. The cost of user coping refers to the sum of these individual user costs.

Cost of Failure and Rework ($C_{Failure} + C_{Rework}$) -- refers to the economic loss caused by requirements-based failures, including recovery, and the associated cost of system correction.

Cost of Specing ($C_{Specing}$) -- refers to the direct cost of requirements specification, calculated as the number of person-weeks multiplied by the fully-burdened pay rate.

Setup costs --

Cost of Learning ($C_{Learning}$) -- refers to the cost of training and experimenting with new specification techniques, calculated as the number of person-weeks multiplied by the fully-burdened pay rate.

Cost of Automating ($C_{Automating}$) -- refers to the direct cost of developing or buying support automation and the cost of learning to use it effectively, calculated as the number of person-weeks multiplied by the fully-burdened pay rate. This value is zero unless such automation is actually acquired.

Using these terms, we calculate the benefits and costs (direct and indirect) of a particular specification strategy as:

Benefits $B = (V_{Automation} + V_{Reuse})$

Costs, Operational $CO = (C_{Specing} + C_{Coping} + C_{Failure} + C_{Rework})$

Costs, Setup $CS = (C_{Learning} + C_{Automating})$

We calculate the ROI percentage in changing from one specification strategy to another with the formula:

$$[(B_{New} - B_{Former}) \; / \; (CO_{New} + CS_{New} - CO_{Former})] \times 100$$

When the setup cost has been fully amortized, the ROI formula becomes:

$$[(B_{New} - B_{Former}) \ / \ (CO_{New} - CO_{Former})] \ x \ 100$$

As an example, we compare a current strategy of specing in Extreme Programming (i.e., development of user stories and tests) to a new strategy that uses the ClearSpecs techniques extensively. Made-up values (i.e., not measured) for this example appear in the table below.

Terms	XP Specing Strategy	ClearSpecs Strategy
$V_{Automation}$	0	40
V_{Reuse}	0	80
$C_{Specing}$	20	200
C_{Coping}	100	10
$C_{Failure}$	60	15
C_{Rework}	40	10
$C_{Learning}$	0	25
$C_{Automating}$	0	40

The ROI formula $[(B_{ClearSpecs} - B_{XP}) \ / \ (CO_{ClearSpecs} + CS_{ClearSpecs} - CO_{XP})] \ x$ 100 yields: [120 / 80] x 100 = 150%. After fully amortizing setup costs over three projects in the first year, the ROI becomes 800%.

There are situations in which the numerator, the denominator, or both may be negative. For example, using the values in the table above, but reversing the transition (i.e., to XP) would result in both terms being negative. A negative numerator (benefit) is a cost, while a negative denominator (cost) is a saving. Therefore, [-120 / -100] means that 100 units of saving yields 120 units of cost.

Additional benefits, more difficult to assess in monetary terms and therefore not reflected in the ROI formula, are:

1. Increased customer confidence throughout the project because they can easily see that their needs have been understood;
2. Decreased frustration among project stakeholders due to decreased ambiguity;
3. Increased quality of communication, coordination, and control by project leaders;
4. Increased product quality and decreased cost due to early detection and management of problems such as inconsistency and disagreements (i.e., early detection of the devil in the details).