Is Your Project Healthy?

CastleBay Consulting

June 2009

Despite the ongoing terrible economic conditions insurance carriers, particularly in the property and casualty domain, continue to invest millions of dollars in the acquisition of new core insurance application systems and the subsequent replacement of fragile and brittle legacy assets. Core legacy replacement projects, the replacement of policy, claims, billing and reinsurance systems, are the most demanding, difficult and risky application systems initiatives that an insurance carrier can undertake. Given that most core legacy systems are twenty to thirty years old such replacement efforts are by definition once-in-a-generation undertakings. And it follows that once-in-a-generation projects are not well understood or necessarily well executed. Indeed, impartial studies have consistently concluded that failure rates for such projects are high. For example industry analyst Celent in 2006 concluded that “between 30% and 80% of all large projects fail, with most estimates coming in on the higher side of this range.[1]” The number we at CastleBay commonly hear anecdotally and observe from our own experience is well north of 50%. While there may have been a general improvement in project execution in recent years it is unrealistic to assume that these numbers have improved significantly.

Measuring the frequency and severity (from partial implementation to total loss) of such failures is hard for several reasons, the single biggest of which is that senior stakeholders are hardly going to advertise such outcomes given their embarrassing and career-threatening nature. Rather, we find the landscape littered with partial implementations which fit radically de-scoped completion criteria, ongoing rollouts which have been temporarily halted, “experiments” and proofs-of-concept which were “interesting” and the lessons of which are being absorbed, etc. So, how do you ensure that your mission-critical project doesn’t end up on this euphemistic scrap-heap, or if it is heading that way how do you identify the fact early enough to either save the project or kill it for a lot less than would otherwise be the case?

Much has been written about how and why projects fail and the post-hoc litany includes such “well knowns” as lack of management commitment, scope creep, changes of senior management, changes in business environment, etc. While these after-the-fact pointers may be true they don't answer the most important question, which is how do you stop a project from failing, or stated in the positive, **how do you keep your project healthy?**

A project is in many ways like a journey. It has a start point and a stated goal or destination. It is expected to reach the destination in a predicted amount of time and to pass through some major checkpoints along the way. And as anyone who has been on an extended journey knows all kinds of things happen along the way: for extended periods of time you are cruising at a steady 70mph on the interstate; then you exit onto the local business loop for food, gas and bathrooms; maybe you have a flat. On a long journey the weather changes, the road conditions change as does you speed and your level of safety. The important thing for successful completion of the journey is to know where you are in relation to your ultimate goal and interim checkpoints and to adjust accordingly. And here is a thought before we leave our analogy behind: although you end up at an exact location in about the time you expected, hardly ever during the journey are you travelling exactly towards your destination and neither do you often travel at the average speed required to meet your expected arrival time. Even an aircraft doesn't fly in a straight line at a constant speed to its destination.

There are various up-front planning principles which can significantly improve the likelihood of project success. Appropriate choices concerning vendors, resources, scope, sequence and project infrastructure greatly increase the project's viability. But even where good initial planning is evident there is a further dimension to be taken into account when considering the health of a project, and that is the element of time.

Given that core legacy replacement projects are complex and lengthy there is no single, point-in-time action which can ensure a successful outcome. As we illustrated with our journey analogy, some aspects of a project will almost certainly be better controlled or more adequately sourced than others and the momentum of the project will vary over time. At certain times the project will hit tough patches, lose focus and slow down, and at other times it will appear to move forward fairly smoothly and consistently. Given these characteristics

- How often should we check up on the health of our project?
- Who should take the measure of the project's health?
- What do we look at in taking stock of the health of the project?

When?

Project health checks should be built into the overall plan for every significant project. It is both possible and desirable to monitor project health with some regularity. Here are some general rules of thumb: first, perform a project health check at appropriate milestone in the project which fall somewhere between two and four month intervals; second, if no obvious milestone events suggest themselves, create them; third, don't be put off by false arguments that dismiss the importance of these checkups based on "ongoing progress monitoring" or "built-in reviews". If these activities were effective the landscape would not be littered with failures. Recognize that while the health check will not stop the project, it will slow it down to some extent. The exercise of assessing the projects health will place extra time requirements on key members of the project team and may briefly slow project execution.

Who?

An in-depth project health check must be undertaken by an outside third party for several reasons. First, if the results are to be believed they must be formulated by an impartial third party. Just as programmers should not test their own code, those who run a project should not review it. Second, staff who are working on the project tend to be so busy and close to the details that it would be a very difficult task for them to stand back from the project sufficiently to see it from an overall perspective. Third, whoever does the health check, be it third party consultants or staff members from another area of the company, need to be competent enough in the project domain and in project execution to formulate an accurate, independent analysis.

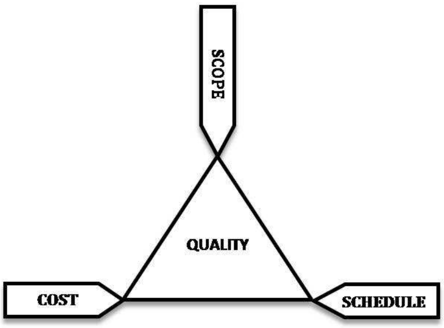
Please note that the internal audit department is not the right group to undertake this effort. As we will see later, a project health check is not a project audit. A health check does not focus on the administration of the project – time keeping and bill paying and the like – rather it focuses on those characteristics which are vital for success – effective execution and appropriate direction.

What?

A project health check should address the following questions:

- Will the project deliver the functionality that the sponsors expect?
- Will the delivery be to level of quality acceptable to the sponsors?
- Will the delivery be made in a timeframe that is acceptable to the sponsors?
- And will the delivery be made at a cost that is acceptable to the sponsors?

These four questions are fundamental to the success of the project and reflect the well known formulation that a project



is a triangular relationship between three variables – scope, schedule and cost (resources). Any two of these variables can change at one time and any change to one variable may cause changes to one of both of the other variables. For example, in order to hold to the schedule we may need to reduce scope and/or increase costs (resources). Note that quality sits in the middle of the triangle and is also affected by changes in the three variables.

In order to answer these fundamental questions about the health of the project the focus of the project health check should be the following four areas – business requirements, personnel, project management and, assuming there is one, the vendor. Let's take a brief look at each one of these.

Business Requirements:

The requirements statement is the "destination" for the project. Without clear and complete requirements no one knows when the project is done, and whether it delivered what it was suppose to. Business requirements includes both high level statements about the project's goals, end state vision and cost benefit as well as mid-level and detailed statements of specific functionality. At the initial review it is appropriate to focus on the project's mission statement, the end-state or to-be vision statement, and the cost benefit analysis. In addition to assessing the plausibility of these statements it is also important to ascertain the extent to which the project team understands and believes in these guiding documents. The following questions should be asked and their answers considered:

- Does the project have a clear, unambiguous vision statement or mission statement?
- Do all team members believe the vision is realistic?
- Does the project have a business case that details the business benefit and how the benefit will be measured?
- Does the project have a user interface prototype that realistically and vividly demonstrates the functionality that the actual system will have?
- Does the project have a detailed, written specification of what the software is supposed to do?
- Did the project team interview people who will actually use the software (end users) early in the project and continue to involve them throughout the project?
- Does the project have detailed, written architecture and design documents?

In subsequent reviews it is more appropriate to focus on the functional requirements which drive the periodic activities of the implementers. Here again, a broad interpretation of "requirements" can be used as appropriate to include functional requirements, look and feel, performance and workflow and process implications.

Personnel:

There are two fundamental questions with reference to personnel. These are 1. Does the project have enough resources? 2. Do those resources have the right levels of expertise and support? Warm bodies and unassigned resources do not make for successful projects. Expertise and experience are required in various key roles, including business subject matter experts, business analysts, system designers, software configurators and integrators, project management staff, quality assurance analysts, trainers and change management experts. Obviously, not all these roles are required at all times during the project – generally speaking requirements gathering comes before configuration/integration, comes before testing, comes before rollout, although this simplistic formulation is significantly modified in an agile project environment – so at different points in the project different groups of personnel will be the focus of the review. The kinds of issues to focus on are:

- Does the project team have expertise with the business domain in which the software will operate?
- Does the project have a leader capable of leading the project successfully?
- Does the project team have all the technical expertise needed to complete the project?
- Are there enough people to do all the work required?
- Does everyone work well together?
- Is each person committed to the project?

Project Management:

The plans, approach documents, risk mitigation strategies and monitoring and reporting are the strategy and tactics for executing the project successfully. Project management also includes the project review actions we are discussing here. The project management review focuses initially on the organizing documents which set up the initial structure of the project – project charter, approach statements, team and governance structures and initial phase plans. In subsequent reviews the focus will be more on the execution and monitoring aspects of the project such as the detailed work break down structure, communications, risk and vendor management plans and the frequency and accuracy of monitoring and reporting against plans. Other project processes such as change management must also be assessed. Later in the project quality assurance, training, rollout and change management plans and procedures will become the areas of focus. The items to focus on include:

- Is a single key executive, with decision-making authority responsible for the project, and does the project have that person's focus and active support?
- Does the project have well-defined, detailed milestones ("binary milestones") that are considered to be either 100 percent done or 100 percent not done?
- Can a project stakeholder easily find out which of these binary milestones have been completed?
- Does the project have a feedback channel by which project members can anonymously report problems to their own managers and upper managers?
- Does the project have a written plan for controlling changes to the software's specification?
- Does the project have a Change Control Board that has final authority to accept or reject proposed changes?
- Does the project have a detailed, written Software Development plan?
- Does the project's task list include data conversion, integration with third-party software, meetings with the customer, and other "minor" tasks?
- Were the schedule and budget estimates officially updated at the end of the most recently completed phase?
- Does the project have a detailed, written Quality Assurance Plan that requires design reviews in addition to system testing?
- Does the project have a detailed Staged Delivery Plan for the software, which describes the stages in which the software will be implemented and delivered?
- Does the project have an overall integrated plan which includes keys task and dependencies between the vendor and client teams?
- Does the project's plan include time for holidays, vacation days, sick days, and ongoing training, and are resources allocated at less than 100 percent?
- Does the project have a Risk Plan which lists current risks to the project? Is this plan reviewed and updated regularly?
- Does the project have stated mitigation strategies for identified major risk?
- Does the project have a "risk officer" who is responsible for identifying

and creating mitigation strategies for new risks?

The Vendor:

If the project which is subject of the review is a core system replacement where the legacy system is to be replaced by a vendor “package” then at least three major aspects of the vendor’s offerings should be closely reviewed: the ability of the software to support the requirements; the expertise and effectiveness of the vendors team in executing their part of the project and interfacing with the wider client team; and the appropriateness and robustness of the vendor’s implementation methodology to organize the core activities of the project in an effective manner. Areas of interest include:

- Is the vendor competent in the business domain?
- Does the vendor have a comprehensive implementation methodology?
- Has the vendor provided a dedicated, competent and adequate team?
- Is the software a good fit for the requirements set (or is significant enhancement required)?
- Is the software mature and stable?
- Does the software make extensive use of rules and data such that system behavior can be modified without significant coding effort?
- Is this project a first of a kind for the vendor, either due to line of business, technical features, software releases or the like?

The Prognosis:

Having asked all these questions and formulated an overall understanding of the project’s health the next task is to characterize the current likelihood of success and to propose concrete actions which will improve the likelihood of success. There are different ways of doing this. One way proposed by Steve McConnell[2], whose general approach is supported here, is to assess findings by attaching numeric values and totaling those values – the project’s health score, rather like a blood pressure reading, characterizes the health and therefore likely success of the project. This level of formality can be instrumental in getting sponsors attention but is not necessary to identify and describe the key areas of potential improvement.

Conclusion:

As we noted earlier it is unlikely that all aspects of the project will be equally well supported at any specific point during the project. It is therefore important to view the project health check process as an ongoing part of the project which is aimed at producing practical recommendations which improve the overall likelihood of project success. Therefore as we also said earlier, not only may each review have a different focus as the project progresses, it should also review the results of recommendations and actions from the previous review. Used regularly and rigorously project health check is a powerful tool for maximizing project success.

References:

1	Celent	The 18 Month Rule: Avoiding The Endless Project		2006
2	Steve McConnell	Code Complete 2	Microsoft Press	2004
		Software Project Survival Guide	Microsoft Press	1997