

Enterprise-class systems deployment: A framework for success



Enterprise-class systems deployment: A framework for success

To even consider the installation or replacement of core insurance systems makes most of us shudder. Perhaps we've heard one too many horror stories, or we're overwhelmed by the variety of systems, or we're taken aback by the sheer time and money it takes to get these things up and running.

In any event, keeping pace with our competitors requires some hard work, and some of the hardest we'll ever be involved with can be the identification, selection, deployment and management of new core systems.

A MAJOR STRATEGIC INITIATIVE

To those contemplating wholesale installations of critical systems that impact the entire enterprise – including policy administration, claims management, billing, business intelligence and other major facilities used in daily insurance operations – the technology landscape is daunting. There are literally dozens of providers, large and small, that promise to be the next big thing: web-based, modular and quick to implement, the sales pitches are tempting, and even whittling down to a short list of prospective vendors could leave you with a dozen or so that match your business profile and base selection criteria. Your IT staff is excited about this stuff, and they should be.

The path of least resistance, of course, is to delegate the selection to them. The development of requests for proposal, vendor selection, requirements gathering and ongoing project management are often the exclusive domain of IT. This is, in my opinion, a critical mistake.

That's not to say that IT staff shouldn't have a lot of input into the selection, planning, implementation and management of your core systems – they most certainly should. However, we're talking about an event, and it's an event that impacts an insurance company profoundly and will continue to do so for the next five, ten or twenty years. The quality of operations – and as such, the quality of service and, ultimately, customer experience – is at stake. This is not a technology decision; it's a major strategic initiative, and should be treated as such.

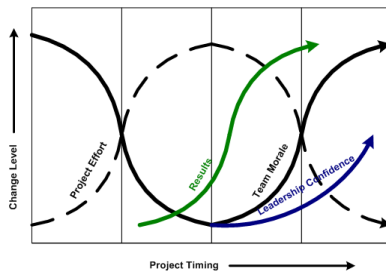
THE BUSINESS/IT DIVIDE

The dreaded divide between business and IT has been experienced by all of us. Surely, technology folks, reticent to labor through the run-of-the-mill implementations of traditional systems and inherently curious critters to boot, will often jump at the chance to work with new technologies and architectures. From this exuberance comes landmark ideas such as object-oriented technology, service-oriented architecture, web services, and the preponderance of acronyms about which we've all become so bitterly cynical that their meaning and importance is lost in a sea of indifference. We've seen and probably all-too-often experienced some degree of failure when we've allowed ourselves to be blindly led through the maze of technological marvels, only to see the vision collapse and, sadly, the occasional career destroyed.

But the world need not be that way. The line between business and IT is blurring. It's blurring at such a blistering pace, in fact, that the need to utterly defer to our technophiles in the absence of our own understanding of the way things work is being quickly supplanted by a more informed approach, easily embraced by those who traditionally limited their roles to pure "business" operations. The battleground between business and IT littered with the carcasses of what were once promising, multimillion dollar deployments that fell far short

Enterprise-class systems deployment: A framework for success

A note on expectations



The project lifecycle

Expectations are a major part of any large-scale endeavor. To illustrate the usual way in which a project gains organizational support, the graphic above depicts the project lifecycle, which is best characterized by the unsettling lag between the massive efforts required to effect change (i.e., getting a new system into production) and actual results (i.e., return on investment). Note that there's a disproportionate amount of effort early on, and the morale of an eager team begins to evaporate in the absence of instant gratification. Likewise, leadership confidence is slow in coming as, again, the failure to produce instant results often causes formerly supportive organizational leaders to retreat, lest they become inextricably attached to a potentially flawed – or worse, failed – project. This distancing is normal, and as it unfolds savvy managers will have already prepped key stakeholders so the inevitable drop in executive support is met with a simple recognition that the cycle is moving forward and in due course leadership confidence will pick up. Understanding and communicating this cycle is critical; successful deployments hinge heavily on expectations, establishing realistic goals and communicating early on the inevitable lag between implementation and results.

of expectations is giving way to a new cooperative spirit, where the business side of the divide is becoming intimately involved with and, in fact, leading what were formerly efforts exclusive to the technical domain. To take full advantage of this blurring of lines and facilitate this new type of cooperation, business managers would do well to become conversant with a few simple technical concepts.¹

Accordingly, provided here are introductions to a few traditionally technical management principles to help you, the insurance executive, to confidently gain more control over the process. To maintain the bridge between business and IT, we introduce the notion of the software development lifecycle, differentiate between project and program management methods, and describe best practice requirements gathering techniques. Further, we explain that the means by which a deployment is managed should be considered in terms of what we introduce here as the organizational context – a comprehensive approach to deployments that respects the relationship between the organization and the many things that influence its effective and efficient operation.

THE SOFTWARE DEVELOPMENT LIFECYCLE (SDLC)

Take note: different vendors will appear to promote different methods of accomplishing their development objectives, each claiming a superior approach in an attempt to differentiate themselves. However, their approaches – each, a software development lifecycle (SDLC) – is simply a method used to group the major activities involved in the creation of software in a way that brings to bear certain disciplines at the appropriate stage. Generally speaking, an SDLC includes a discovery stage, a design stage, a build stage and a launch stage. Sometimes these stages take place serially, sometimes concurrently and sometimes some combination of serial and parallel development takes place. These stages have been revamped and reworded to create “proprietary” frameworks by most vendors, but in essence, most share the same basic structure, and with each stage we find a common set of activities and objectives, knowledge of which helps us to cut through the sales rhetoric.

- **DISCOVERY.** This is the stage where investigative work is done. Typically, during discovery user requirements are captured and documented by business analysts. It's the job of the business analyst to have at least a basic understanding of your business; otherwise they may fail to ask the right questions and receive the proper answers that drive an effective development effort. The product of discovery includes business, functional, technical, legal and usability requirements and often includes draft documentation. These requirements documents are handed off to the development team for use during subsequent stages.
- **DESIGN.** During the design stage, software architects develop the context within which the systems being developed will fit, and the means by which the various components will interact. The design stage also includes the creation of formal requirements documentation and project schedules for use by development teams.

Much attention should be paid to a vendor's preferred approach. Be especially wary of vendors who claim their own "best practice" methods absent solid documentation and a proven track record.

- **BUILD.** The build stage includes the actual programming of software code, the loading of rates, rules and forms necessary for policy output, billing scenarios, and other system configuration input. This stage also includes intensive testing for accuracy. For comprehensive policy system implementations, the number of possible policy combinations can be enormous and, as such, lend themselves to automated testing – a technique whereby base test scripts run automatically through multiple policy scenarios, each for a different state/LOB/endorsement/declaration page combination. There are many automated testing applications on the market that are available at relatively modest cost. Accordingly, no vendor should be without one.
- **RELEASE.** Too often overlooked, a formal release should involve all stakeholders and include comprehensive user training and rolled deployment of the fully developed, tested systems. Rolled deployment means lines of business are brought up successively to work out any kinks in the system prior to a complete transfer of business from legacy systems.

These stages are variously called envisioning, planning, developing, stabilizing (Microsoft Solutions Framework); inception, elaboration, construction, transition (Rational Unified Process); planning, requirements analysis, design, coding, testing, documentation (Agile); listening, designing, coding, testing (eXtreme Programming); and requirements analysis, design, implementation, testing, integration and maintenance (Waterfall). Regardless of the SDLC methodology employed, however, the result should be the same: a software product that meets or exceeds the requirements of users. As such, I won't advocate for one over another; they all have their pros and cons, with some more appropriate than others given the size, scope and scale of the implementation being undertaken. The point is to understand that the SDLCs described above (among others) represent best practices developed over many years through thousands of projects. As such, much attention should be paid to a vendor's preferred approach. Be especially wary of vendors who claim their own "best practice" methods absent solid documentation and a proven track record.

PROJECT, PROGRAM AND PORTFOLIO MANAGEMENT

PROJECT MANAGEMENT

Project management involves more than simply putting tasks and projected completion dates into a project plan and periodically assessing how far off course the project is. Adherents to the Project Management Institute's Project Management Body of Knowledge (PMBOK) know that project management is a profession that involves nine separate disciplines: integration management, scope management, time management, cost management, quality management, human resource management, communications management, risk management and procurement management.² Employing project management best practices means proficiency is demonstrated in most, if not all, of these disciplines throughout the deployment.

Project management is a profession that involves nine separate disciplines: integration management, scope management, time management, cost management, quality management, human resource management, communications management, risk management and procurement management.

- **INTEGRATION MANAGEMENT.** Project integration management includes the creation and validation of project plans, project plan execution strategies, ongoing management tools (including the creation of dashboards and periodic reports) and other techniques employed to successfully deliver the system being developed into production.
- **SCOPE MANAGEMENT.** The place where many projects get into trouble is the dreaded “scope creep.” Scope creep is a sort of death by needles, where one or two breaches won’t kill you, but tens, hundreds or thousands of changes and enhancements send project teams reeling and cause the delays and cost overruns that plague poorly run implementations. Good scope management means accurately defining precisely what is to be accomplished and gaining consensus among all stakeholders regarding the completeness of the scope statement. Once defined, unless absolutely critical to address regulatory mandates or competitive threats, the scope should remain fixed. Enhancements should be queued up for future releases and governed by a strictly-enforced change control procedure that requires any requested change to be evaluated for impact on project schedule and cost prior to being approved.
- **TIME MANAGEMENT.** Another major cause of project delay is inadequate time management. Like so many other project management skills, time management is all too often taken for granted. Plugging dates into Microsoft Outlook or a project schedule and monitoring them is not time management. Time management is a systematic method of assessing the time required to complete individual tasks while respecting the impact each has on all others. Time management involves discipline and focus. Ironically, the use of the ubiquitous Blackberries, Treos and other communications devices often flies in the face of good time management. In a popular book on the topic, *Finding Time: How Corporations, Individuals and Families Can Benefit from New Work Practices*, author Leslie Perlow posits that interruptions – email, phone calls, meetings – those “necessities” that inundate our days – provide the worst types of infringement on good time management practices.³ To compound this point, Mihaly Csikszentmihalyi has written in his seminal work on creative productivity, *Flow: The Psychology of Optimal Experience*, that those who get the most done, too, express far greater volumes of output when they are left alone and are able to get into “flow” – where time slips away as they toil diligently “in the zone.”⁴ We’ve all experienced it; it’s those modern conveniences that disable our ability to ever achieve that state of flow, of optimal experience, and actually infringe on our ability to get things done.
- **COST MANAGEMENT.** How many development teams actually pay attention to the bottom line? What is the anticipated return on investment? How much awareness is there among the stakeholders regarding the implementation budget and expected returns? In addition to outright system costs (software licenses, hardware purchases, professional services), has the cost of internal staff been considered? Have

Enterprise-class systems deployment: A framework for success

Employing project management best practices means proficiency is demonstrated in most, if not all, disciplines throughout the deployment.

the post-implementation support needs been identified? At what cost is internal staff utilized to assist the vendor(s) in their implementation, before, during and after the deployment? These are real costs that are often overlooked.

- **QUALITY MANAGEMENT.** Quality guru Philip Crosby wrote a book many years ago entitled *Quality is Free*. The central premise of the book is that taking the time to ensure the output from work processes (in our example, an enterprise-class system deployment) is of high quality is well worth the added expense.⁵ Why? The rework, bug fixes and “bad will” generated by inferior implementations take away far more than any expense added to ensure a quality result. As such, the basic tools of quality management, including quality function deployment, failure mode effects analysis and process mapping should be utilized extensively throughout the implementation. These, too, are easily acquired skills that are often taken for granted. The absence of their use begets faulty systems, increased costs, unhappy users and, ultimately, dissatisfied customers.
- **HUMAN RESOURCE MANAGEMENT.** The appointment of team members, their indoctrination into their respective project teams, training, motivation and compensation practices all weigh heavily on the effectiveness of the team and the success of the implementation. Human beings are the machines charged with getting your implementation done, and like most humans, they probably need the same motivation and management skills applied to them as anyone else. As such, pay attention to how your vendor manages their internal staff and how your own company management deals with implementation team members.
- **COMMUNICATIONS MANAGEMENT.** Here, again, is an area that tends to be an afterthought. The absence of a well-documented communications strategy leads to ad hoc meetings, missed conference calls and general misunderstandings. Good communications management begins with a simple communications plan – a document containing the names of all stakeholders, their contact information, their roles and responsibilities, the type, frequency and location of meetings, conference call and online meeting instructions and the web address of a collaboration space (i.e., a place where all project and program-related documents are located) to which everyone involved has access.
- **RISK MANAGEMENT.** At the outset of the deployment and at various times throughout stakeholders should assemble and brainstorm all possible problems that, left unchecked, might derail it. In these sessions anything goes; a scribe in attendance should diligently write down all ideas that flow from the group. These ideas should then be categorized into affinity groups (for ease of management) and, using a scale of one to ten, each major affinity group assigned a ballpark probability of occurrence (1 = unlikely to occur; 10 = very likely to occur), severity (1 = not severe, 10 = very severe) and ease of detection (1 = easy to detect, 10 = difficult to detect). The product of those three numbers – the Risk

While a project comprises a set of related activities whose collective completion yields a specific deliverable, a program is a set of interdependent projects that yield multiple deliverables.

Priority Number or RPN – helps the team to focus its efforts on the most likely project failures that are most difficult to detect that would have the greatest adverse impact on the implementation. For each major risk category, a contingency plan should be developed, and that risk monitored by the project staff to help detect it should it occur.

- **PROCUREMENT MANAGEMENT.** Finally, few things frustrate a team more than having to wait for a piece of hardware to show up in order to install and test code in a production environment. Software, peripherals, and other ancillary components necessary for the efficient operating of the new system should be itemized, budgeted and ordered such that they're delivered to the implementation team in a timely manner. Further, third-party services required for project completion, including information feeds and professional services, should be researched and contracted well in advance of their actual need. Should your chosen vendor work with third parties in their implementation projects, gain a solid understanding of and comfort with their request for proposal (RFP), vendor selection and related processes.

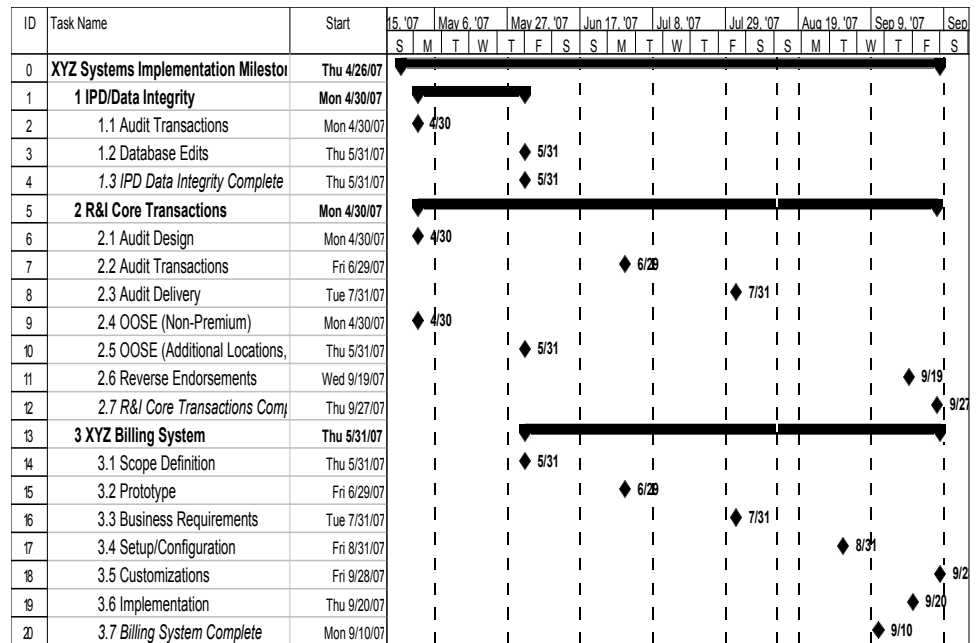
PROGRAM MANAGEMENT

A major, enterprise-class systems deployment (ECSD) will ultimately involve many different pieces. All too often I've seen large-scale systems implementations fall apart for a relatively simple reason: the failure to differentiate between project management and program management. While a project comprises a set of related activities whose collective completion yields a specific deliverable, a program is a set of interdependent projects that yield multiple deliverables.

To be sure, a major ECSD at an insurance company might involve policy issuance, claims management, billing, accounting, document management and other systems that together provide core operational functionality. Further, there's documentation to be developed, training to take place and a formal launch to occur. As such, a single ECSD might comprise five, six, seven or more projects, each of which is dependent upon one or more other projects in order to realize the vision of a consolidated, comprehensive insurance operations system.

Just as each player in an orchestra works from her own sheet music, each discrete project deserves its own project plan, and the collection of project plans representing the components of the overall system should be managed via a program plan.

Enterprise-class systems deployment: A framework for success



A program plan

The program plan is a high-level schedule of milestones and project dependencies that is the central management tool for any well-managed ECSD, and is analogous to the orchestra conductor's score. A conductor who attempts to lead an orchestra by simultaneously reading the parts of every player will have a hard time queuing the right players at the right time. Likewise, an ECSD managed without a central program schedule will be terribly difficult to manage – and far more prone to failure.

PORTFOLIO MANAGEMENT

Finally, the collection of projects and programs being undertaken by an organization have varying degrees of importance, and as such are subject to prioritization. Project Portfolio Management, or PPM, is a relatively new and fast-growing discipline that applies the base concepts of Modern Portfolio Theory traditionally used for investments. The guiding premises of PPM are balance and focus: the portfolio should represent a good balance of projects that address the organization's needs, while focus or priority should be given to those deemed critical to the attainment of the organization's larger strategic objectives.

REQUIREMENTS GATHERING

The means by which requirements are gathered can make or break any system deployment. A best practice involves the creation of use cases – narrative descriptions of the step-by-step processes users will follow when working within the system under development. Each user is represented by one or more actors, and each actor has a goal. The steps leading to the goal represent the main success scenario. Alternative paths to the goal – due to system failures, errors or unexpected results – are termed extensions to the use case.⁶ Gathering requirements in this manner provides an extremely coherent, intuitive

Project Portfolio Management, or PPM, is a relatively new and fast-growing discipline that applies the base concepts of Modern Portfolio Theory traditionally used for investments.

A best practice involves the creation of use cases – narrative descriptions of the step-by-step processes users will follow when working within the system under development.

path for developers to follow, and the writer of use cases need not have any real technical knowledge, other than how users interact (or desire to interact) with the system under development.

Use Case: Take Application

Actor: Agent

Goal: Completed Application

Main Success Scenario

1. Agent logs into system
2. System presents menu of choices
3. Agent selects "application" from menu
4. System presents "application" submenu
5. System presents initial input screen
6. Agent inputs customer information
 1. Customer name
 2. Customer address
 3. FEIN
 4. Contact telephone
 5. Email address
7. Agent clicks "submit" button
8. System returns review screen
9. Agent verifies information
10. ...

Extensions

- 9a. Information is incorrect
 1. Agent clicks "edit" button
 2. System returns input screen
 3. Agent corrects information
 4. Go to step 7

A typical use case (abridged)

Use cases are typically complemented by functional and technical requirements, where page load times, ease of use, system capacity and other aspects of the development effort are documented. The use case format provides an excellent framework for test plans, as those conducting post-development system tests can utilize the use cases as a step-by-step guide to realizing specific system goals, and indicate whether each step either passes or fails based on the testers' experience.

There is rarely a need to complete all requirements prior to commencing the development effort. However, in some instances, including offshore development initiatives and complex projects with widely distributed project team members, completed formal requirements help to close the gaps in time and space that can otherwise disable geographically dispersed teams, by providing each member with an identical set of detailed instructions.

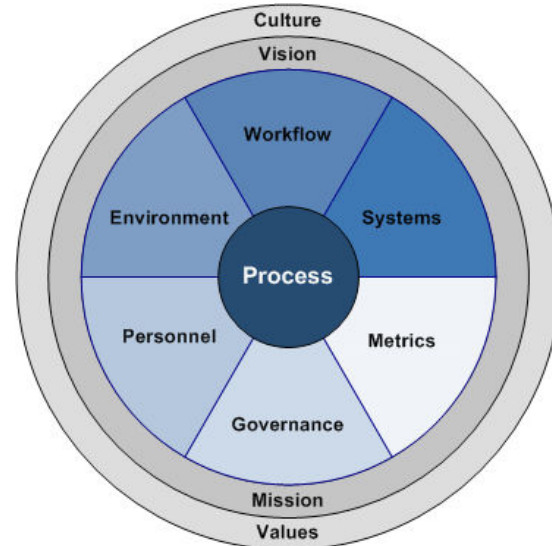
THE ORGANIZATIONAL CONTEXT

If this article were a short story, this section would be the climax. The fundamentals have been described thus far – concepts that any vendor should embrace in some form or another about which you should be aware. How those tools are utilized within the context of the entire organization is another

Enterprise-class systems deployment: A framework for success

No deployment takes place in a vacuum; as a major strategic process it has a profound reciprocal relationship with the organization as a whole, including every one of its constituents.

matter entirely. No deployment takes place in a vacuum; as a major strategic process it has a profound reciprocal relationship with the organization as a whole, including every one of its constituents. In planning the implementation, therefore, it's critical to pay attention to those aspects of the organization that enable or constrain its successful completion – aspects I refer to as process influencers.



Process in the organizational context

A typical approach to a deployment includes an analysis of workflow patterns and supporting systems required to enable them, yet fails to acknowledge the role the many other influencers play in the effectiveness or efficiency of the process. Key performance indicators (metrics), policies and regulations (governance), personnel issues (hiring, training and compensation practices) and working environment all must be considered in light of the mission, vision, values and culture that shape the organization and the implementation team. An understanding of these factors provides a comprehensive foundation that considers the deployment in its “organizational context” rather than approaching it as a standalone undertaking.

- **WORKFLOW.** How workflow is impacted lies at the heart of the deployment. The way people interact with the new system will inevitably differ from the way they were used to working. Accordingly, modeling existing workflow (“as-is”) and improved workflow (“should be”) must include the input of those who will daily use the system. Nothing brings a deployment to a grinding halt faster than a workforce that refuses to work within the bounds of the new system, even when those bounds are far wider than they were previously. Including staff early in the process (e.g., during requirements gathering) secures their ownership and ensures their buy-in of any change in the way work is performed.
- **SYSTEMS.** How is the implementation impacted by existing systems? How will the implementation impact existing systems? Will there be

Enterprise-class systems deployment: A framework for success

An understanding of all relevant factors provides a comprehensive foundation that considers the deployment in its “organizational context” rather than approaching it as a stand-alone undertaking.

interfaces with any part of the legacy infrastructure? How will those interfaces be managed? Are custom programming or third-party tools required? What’s the integration plan? Who’s responsible for effecting it? Answering these questions helps bridge the gap between old and new systems.

- **METRICS.** There’s the adage “you can’t manage what you don’t measure” and nowhere is this thought more fitting than in new systems deployments. Be sure to accumulate key statistics prior to the implementation that demonstrate the current state of the process, including cycle time (e.g., application to issuance for policies), error rates and costs, then measure them afterwards. Illustrating the direct impact of the new system with hard numbers will help to secure the buy-in that facilitates management, even from those who would otherwise rather do things “the old way.”
- **GOVERNANCE.** Policies, both internal and mandated by legislation, can have a major impact on deployments. Are internal policies unnecessarily constraining? Which regulations must you respect (e.g., Sarbanes-Oxley) or which frameworks do you choose to respect (e.g., COSO, Six Sigma) for compliance purposes? How are existing systems helping or hindering your organization’s compliance efforts? How can new systems facilitate compliance without unduly burdening staff or increasing costs?
- **PERSONNEL.** Hiring practices, training programs and compensation plans will profoundly impact the way in which employees approach the new system deployment. Hiring practices should always consider the skills, temperaments and commitment required for major change initiatives. Training should be focused and always include project management principles as part of the curriculum. A proper alignment of organizational objectives (in this case, getting the new systems into production as quickly as possible) and employee rewards is critical to the efficient mobilization of staff working together toward the realization of operational goals.
- **ENVIRONMENT.** The physical location of project teams and their juxtaposition to organization staff on whom they are relying for critical input must be taken into account as well. For major systems implementations, you’ll want a few onsite representatives from the vendor, especially during the early stages. As important as where the team members are located is how they’re located; providing a work environment that’s conducive to creative collaboration helps tremendously. Be sure to have accommodations that enable rather than constrain effective working teams.

FOUNDATIONAL ASPECTS

In working with systems vendors and integrating their efforts with those of your own organization, an understanding of those foundational aspects that motivate people to action is an important underpinning to a successful deployment.

Enterprise-class systems deployment: A framework for success

Mission, vision, values and culture all play a part in guiding the inter-company teams that emerge during major deployments.

Mission, vision, values and culture all play a part in guiding the inter-company teams that emerge during major deployments. Interestingly, these teams evolve their own missions, visions, values and culture; understanding what they are and how they fit within the enterprise is an important part of managing the deployment in the proper organizational context.

- **MISSION.** A mission statement is defined as a concise statement of the reasons for an organization's existence, its functions, its target market, and the means by which it intends to fulfill its purpose. It's focused on day-to-day operations, is generic enough to cover all strategies and broad enough to cover the complete area of operations. In assessing the mission statement of the deployment team, we seek to answer four primary questions:
 - What functions does the deployment team perform?
 - For whom does the team perform these functions?
 - How does the team go about filling this function?
 - Why does this team exist?

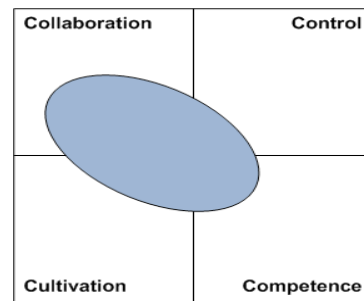
The mission statement is a logical starting point for any ECSD.

- **VISION.** Vision is a future state of the enterprise, without regard to how it's achieved. It represents the ultimate state the enterprise would like to achieve. While the mission is translated into strategies and tactics, the vision is translated into goals (resulting from strategy), and objectives (achieved tactically). The truly compelling part of an effective vision is a view of the future, embraced by all, toward which they are collectively moving. Getting this down on paper and having stakeholders refer to it frequently keeps the effort focused and purposeful.
- **VALUES.** Vision begins with an understanding of values common to members of the team. A brief poll that asks what each team member considers to be their most important values can be quite revealing. Integrity, honesty, providing value, creative expression, professionalism and open communication are common values that are typically uncovered during such a poll. These core values indicate things that are universally important to team members and their respective organizations. Identifying them, exemplifying them and reinforcing them will create a resonance among team members as they work daily toward the fulfillment of their common goals. Any organization that lives by the collective values of its workforce empowers its employees to work with purpose – a critical characteristic of excellent operations and a hallmark of successful deployments.
- **CULTURE.** The culture that emerges by melding project teams from multiple organizations profoundly influences the focus, intensity and effectiveness of team members. Cultures characterized by finger-pointing and blame over cooperation and acceptance of responsibility are doomed to failure. As such, understanding the culture is a foundational piece of our discovery process. (We acknowledge the work of Geoffrey

Enterprise-class systems deployment: A framework for success

The culture that emerges by melding project teams from multiple organizations profoundly influences the focus, intensity and effectiveness of team members. Understanding the culture is a foundational piece of our discovery process.

Moore and TCG Advisors in their identification of four major culture types.)⁷



Major culture types

While organizations exhibit one dominant cultural type, they typically demonstrate qualities of all others as well.

- A **COLLABORATION** culture exhibits synergy, equality, unity and involvement, and is driven by the need for affiliation.
- A **CONTROL** culture exhibits certainty, systemization, objectivity, stability, standardization and predictability, and is driven by the need for power and security.
- A **COMPETENCE** culture exhibits professionalism, meritocracy, continuous improvement, accuracy and autonomy, and is driven by the need for achievement.
- A **CULTIVATION** culture exhibits growth, development, commitment, creativity, purpose and subjectivity, and is driven by a need to realize potential.

No team or organization will fall precisely into any one of these categories, rather, the culture is typically a mix of types. The figure above illustrates the result of a culture assessment that reveals a strong Collaboration culture that “dips” into each of the other three culture types, though far less so. While Collaboration is the dominant type, the culture depicted exhibits qualities of the others as well. Where the stakeholders and their respective organizations fit in this model provides a key insight into how best to approach the management of the effort.

CASE STUDY: A SMALL BUT FAST-GROWING INSURER

A recent engagement by our firm involved a mid-sized insurance company struggling with a seriously delayed policy administration system implementation. The project, some six months behind schedule, had already eaten 14 months and \$1.5 million with no end in sight. In addition, the company’s business development group was busily acquiring books of business that had to be brought online quickly, however, each new program implementation was taking somewhere around 150 days. A major component of the system had to be

The company was seriously considering pulling the plug and seeking a replacement vendor. Before taking that drastic step, they engaged our firm to provide an objective assessment of the situation.

rebuilt from scratch resulting in the need to bill policyholders manually. Many simple functions had severe limitations (e.g., a four-state limit for multi-state workers' compensation policies). The company was seriously considering pulling the plug and seeking a replacement vendor (and a lawyer!).

Before taking that drastic step, they engaged our firm to provide an objective assessment of the situation. Over four days, we reviewed mountains of documents, conducted staff interviews at both the company and the vendor and did a deep dive into the technology and the vendor's practices. The first word that comes to mind when I think back on that engagement is defensiveness. The vendor was certain I was "throwing them under the bus," and the company staff was, to say the least, unsettled by my presence. The head of IT at the company blamed the vendor, the vendor blamed the company, and no one seemed willing to accept responsibility, evaluate the situation fairly and get things back on track. Progress reporting to senior management involved a black and white photocopy of a high-level system diagram with percentages written next to each component indicating how much work had been done to date ("all guesses," according to the head of IT). There was no form of project planning evident and the only requirements documentation available was a technical specification produced by the company in a desperate attempt to provide some structure for the virtually non-existent (but promised) billing module.

Our written assessment detailed the results of staff interviews and technology evaluations and concluded that the most pressing issue was the abject lack of cooperation between the company and the vendor. There were few signs of collaboration; instead, each interview with company staff became a prolonged diatribe about the shortcomings of the vendor, and you can probably guess that each with the vendor's staff became a rant about the inadequacy of the company's staff. This, of course, highlights the importance of cultural considerations when choosing a vendor; the company and vendor had distinctly different (and incompatible) culture types. This led us to conclude that the problem began at the vendor selection stage – that the selection process failed to consider candidates in the "organizational context," and as such, critical selection criteria were overlooked.

The next big issue was the lack of any formal project management practices. A project plan was nowhere to be found. The entire implementation – including 12 individual components – was undertaken as a mass development effort with no vision, no differentiation of objectives and no ownership over any particular piece of the complex puzzle the company and vendor were attempting to assemble.

Next, there was no identifiable best practice employed. I'm a big fan of "rolling requirements," meaning, there is generally no need for completing requirements documentation prior to the commencement of work (unless the job is completely outsourced, especially offshore, as indicated earlier). However, the vendor's development methodology, while modeled on an "Agile" framework that supports this notion, was undocumented and largely ad hoc. Further, the Agile approach was thoroughly inappropriate, as an Agile development process demands a committed, highly collaborative, co-located team that can

Our written assessment detailed the results of staff interviews and technology evaluations and concluded that the most pressing issue was the abject lack of cooperation between the company and the vendor.

quickly produce working prototypes about which immediate feedback from multiple stakeholders (including marketing, business development, executive and other non-technical folks) can be applied. In addition, there were no traditional Agile tools being used (e.g., scrum teams, burn-down charts, etc.), and the vendor had apparently knitted together their own hybrid development framework that borrowed from various best practices but committed to none. The environment simply wasn't right for this approach. Once again, had the vendor selection process considered the vendor's development methodology (provided the selection committee was somewhat educated about such methodologies), this impediment may have been noticed earlier and many of the problems associated with inadequate development processes mitigated.

Finally, company management had virtually no visibility over the status of the implementation. They relied entirely on the previously mentioned photocopied system diagram and estimated percentage completions indicated on it for status which, on further examination, proved to be wildly inaccurate.

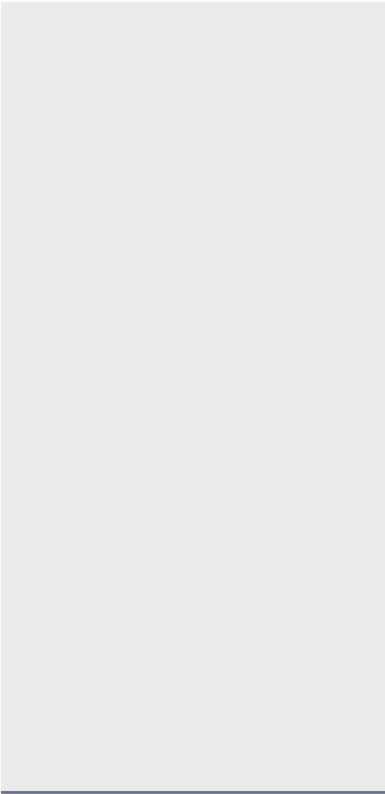
SOLUTIONS

Going through vendor selection again was not an option, given the investment of time and money already sunk into the implementation. This was a classic project rescue, and the steps to accomplish that were obvious.

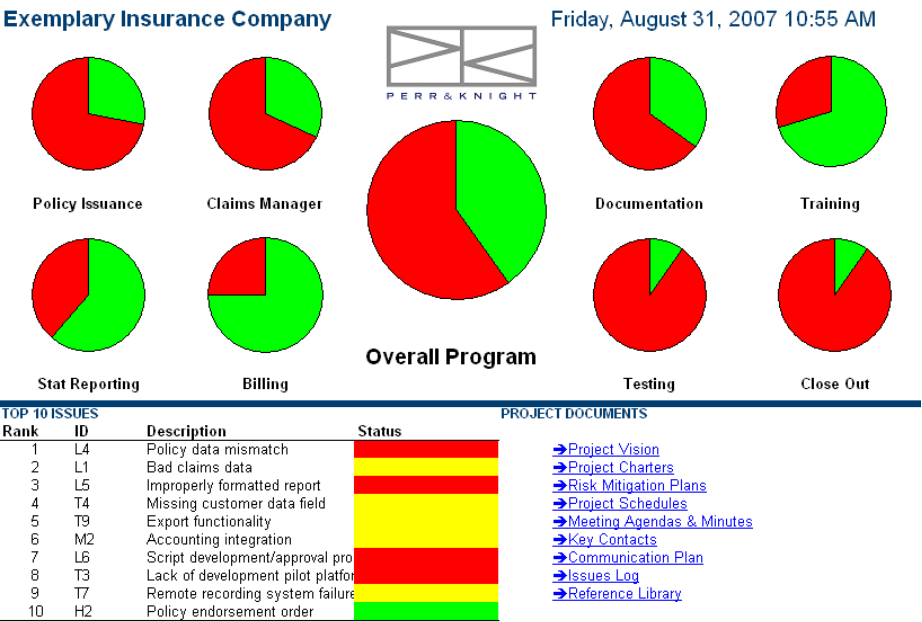
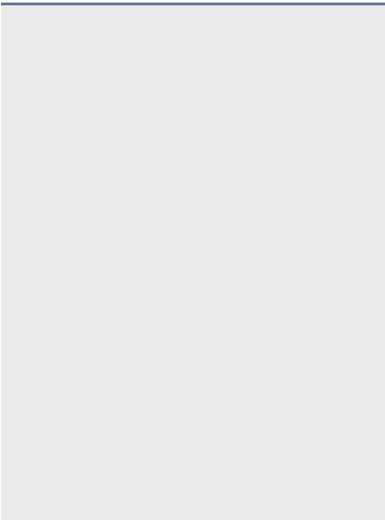
To get things back on track, the four most important modules being developed were made priorities and for each a separate project plan was created. In addition, for each of these "core" projects, a charter was drafted that provided on one page the project owner, team members, key milestone dates, project objectives and project risks. The milestones from each core project were added to a central program schedule to which the head of IT – the "orchestra conductor" – at the company could refer.

Next, the head of IT left the company. Surely, this was more symbolic than anything as blame could not be placed entirely with that one person, but there were fundamental flaws in the vendor selection and project management processes, and the move by one individual to accept responsibility at once demonstrated the gravity of the situation and the seriousness of senior management's commitment to deal with it. That one staff change practically eliminated the finger-pointing that characterized the relationship previously.

Next, a program dashboard was created to provide senior management with a web-based view of the progress of the overall program, each core project and several ancillary projects. Progress was indicated for each by a small pie chart that, when clicked, would launch a detailed project schedule. The dashboard also provided the status of the top ten open issues facing the implementation, and access to a newly defined program vision document, individual project charters and a communications plan that listed all stakeholders and their contact information, specified regular meeting days and times and included log in information and passwords for teleconferencing and online demonstrations. In addition to providing senior management with visibility, all team members had access to the same set of progress reports, project schedules and project-related documents in one place.



To date, some five months after the conclusion of our engagement, 16 new programs have been brought online, and the average time to bring a new program online has dropped by 40% from an average of 150 days to just 90.



A program dashboard

Finally, the vendor agreed to provide onsite management at least one full day per week until the deployment was complete.

RESULTS

The findings of our assessment and subsequent recommendations applied by the company had a profoundly positive effect on the implementation. Once the head of IT left, the team quickly re-aligned, project and program plans guided the effort, meetings occurred with more regularity and purpose, and the web-based dashboard provided senior management with excellent visibility into the progress of the overall deployment. To date, some five months after the conclusion of our engagement, 16 new programs have been brought online, and the average time to bring a new program online has dropped by 40% from an average of 150 days to just 90.

The financial benefits were immediately evident. First off, the implementation was saved and the time (14+ months) and money (approximately \$1.5 million) was not spent in vain. The company is solidly on its growth plan as the implementation continues due to the flow of new books of business being acquired.

SUMMARY

Often when speaking in public forums, I can see eyes starting to glaze over when I begin to discuss the “soft” aspects of management, like culture and values and vision statements. As a critical underpinning to any effort involving the coordination of multiple, often diverse team members, these soft aspects are of paramount importance. By all means, get to know both your own company culture and that of any vendor with which you place so much time, money and trust. Further, considering the entire organization – embracing an implementation as a critical strategic process influenced by multiple factors – enables the

The tools and methods described here, while not exhaustive, provide a solid foundation for any major system initiative.

realization of a multitude of benefits sorely lacking from less comprehensive approaches.

The “harder” aspects of an implementation, including project and program management skills and software development and requirements gathering techniques are all too often left to chance. But how many “proprietary software development methodologies” do we have to hear about before they become so obviously similar? How many “project managers” do we have to meet who can’t return a phone call or produce a program schedule? How many “business analysts” do we have to deal with who don’t take the time to know our unique businesses or can’t write a simple use case? Do not let these pretenders infiltrate your deployment and risk its failure; educate yourself, learn what to look for among all team members and call them on it when they appear to fall short.

Finally, no one method or approach is perfect. There are far too many variables to arrive at a single “best practice” regardless of the technology being deployed, the team members involved or the company being impacted. However, the tools and methods described here, while not exhaustive, provide a solid foundation for any major system initiative. Yes, applying these principles is time-consuming. It takes time to evaluate vendors, to understand their approaches, to plan and manage projects and to test the results of team efforts. There are the inevitable downturns that bruise egos and destroy morale. There are the false starts and outright failures that make our days grow long and our bodies weary. But we all know that an ounce of prevention is worth a pound of cure – and the committed consumption of the concepts you’ve just read about – a diet of best practices rife with rewards – worth every bit of effort it takes to eat ‘em up.

REFERENCES

- ¹ Berg, R. (2007, Summer). Using the whole brain: Bridging the business/IT divide for implementation success. *The Interpreter* (IASA), 8 – 11.
- ² Project Management Institute. (2000). *Guide to the project management body of knowledge* (PMBOK). PMI.
- ³ Perlow, L. (1997). *Finding time: How corporations, individuals and families can benefit from new work practices*. Ithaca, NY: Cornell University Press.
- ⁴ Csikszentmihalyi, M. (1997). *Flow: The psychology of optimal experience*. New York: HarperPerennial.
- ⁵ Crosby, P. (1980). *Quality is free*. New York: Signet.
- ⁶ Cockburn, A. (2000). *Writing effective use cases*. Upper Saddle River, NJ: Addison-Wesley.
- ⁷ Moore, G. (2002). *Living on the fault line, revised edition: Managing for shareholder value in any economy*. New York: HarperCollins.

ABOUT THE AUTHOR

Rob Berg is a Principal and Director of Perr&Knight's Insurance Technology Group (ITG). In addition to managing Perr&Knight's internal development initiatives, Rob and the ITG team work with insurers to provide solution design, requirements gathering, vendor selection, business case development, project planning, risk assessment, business process design, IT governance, data management and staff augmentation services for their technology initiatives.

Over a career spanning more than twenty years, Rob has led or advised executive teams in the financial services, consumer retail, software and telecommunications industries. His expertise includes strategic planning, organizational design, project management and process improvement methodologies, including workflow design, analysis and simulation. Prior to joining Perr&Knight, Rob was Chief Operating Officer of SmartReply, a leading retail marketing firm that enjoyed three consecutive years of triple-digit growth during his tenure. Prior to SmartReply, Rob was Chief Executive Officer of Ecor Solutions, a New York-based IT consulting firm that counted Deutsche Bank, W Hotels and American Legend among its clientele.

In addition to holding a Bachelor of Arts in Economics from Stony Brook University and completing graduate work in technology management, decision theory and organizational behavior, Rob holds credentials from the American Society for Quality (Six Sigma Black Belt) and Stanford University (Advanced Project Management). He is a member of ISACA and a Senior Member of the American Society for Quality, a frequent speaker at industry trade events, and has been quoted or published in multiple industry trades including Best's Review, IASA Interpreter, InsuranceNewsNet and the Journal of Insurance Operations, of which he is Editor-in-Chief.

ABOUT PERR&KNIGHT

Perr&Knight is a leading provider of insurance support services and a strategic resource that companies utilize to reduce their fixed costs while increasing the efficiency and value of their insurance operations. Perr&Knight's insurance support services include Actuarial Consulting, Competitive Intelligence, Data Services, Insurance Technology and Regulatory Compliance.

CONTACT US

PHONE: 310.230.9339

E-MAIL: sales@perrknight.com

WEB: perrknight.com

© 2010 PERR&KNIGHT