Quality & Risk Management Challenges When Acquiring Enterprise Systems

Ying Ki Kwong, PhD, PMP
Office of the State Chief Information Officer
State of Oregon
ying.k.kwong@oregon.gov

Philip Lew, PhD, PMP
CEO, XBOSoft
philip.lew@xbosoft.com

Abstract

Much has been written about quality and risk management from the perspectives of organizations that design, develop, and implement software systems. However, many enterprises no longer build software systems internally but, instead, acquire them from contractors. For an organization that acquires software systems, project management needs to be done from a different perspective, and quality and risk management needs a different focus and emphasis than the organization responsible for building the systems. This paper marks the tenth anniversary of a PNSQC 2008 presentation entitled “Management of Outsourced Projects.” Many of the topics discussed then continue to be relevant today. The major difference between then and now, however, is the popularity of agile development methodologies. Agile, with some of its primary tenets including focus on short iterations, needs to be adapted in order to work successfully in an environment where organizations acquire software built by a third party. In this paper, the authors categorize problems and issues that arise when fitting Agile into these environments and discuss areas where Agile needs to be adapted in order to be successful. We conclude with the observation that system development life cycle (SDLC) methodologies like Agile is one facet of delivering enterprise system projects successfully in large organizations.

Biography

Ying Ki Kwong is the Statewide Quality Assurance Program Manager in the Office of the State CIO in Oregon state government. Prior to this role, he was IT Investment Oversight Coordinator in the same office and was Project Office Manager of the Medicaid Management Information System Project in the Oregon Department of Human Services. In the private sector, Dr. Kwong was CEO of a Hong Kong-based internet B2B portal for trading commodities futures and metals. He was a program manager in the Video & Networking Division of Tektronix, responsible for worldwide applications & channels marketing for a line of video servers in broadcast television applications. In these roles, he has managed software based systems/applications, products, and business process improvements. He received the doctorate from the School of Applied & Engineering Physics at Cornell University and was adjunct faculty in the School of Business Administration at Portland State University. He holds the PMP certification since 2003.

Philip Lew, CEO of XBOSoft, oversees strategy, operations and business development since founding the Company in 2006. In a span of 25 years he has worked as a developer, product manager, and held roles at the executive level both in USA and Europe. He also serves as an Adjunct Professor at Alaska Pacific University. He has spoken at conferences such as STPCon, PNSQC and Better Software East-West, StarEast-West while his papers have been published in ACM, IEEE, Project Management Technology, Network World, Telecommunications Magazine, Call Center Magazine, TeleProfessional, and DataPro Research Reports. Philip Lew is a certified PMP and holds a BS and a Masters Degree in Operations Research and Engineering from Cornell University and a Ph.D. in Computer Science and Engineering from Beihang University. He loves bicycling and has traveled the world visiting more than 60 countries to quell his passion for exploration and learning.

1. Introduction

In many organizations, enterprise software systems are acquired and not built internally. In these projects, the software development is outsourced to contractors and may span all aspects of the system development life cycle (SDLC). Even when a prime contractor is experienced and has a good track record of delivering projects successfully, a project that involves the acquisition of an enterprise system is a complex undertaking and so high risk for an organization acquiring the enterprise system ("acquiring organization").

Many times, there may be conflict between contracting model and SDLC. This is especially true with Agile, where requirements are supposed to be discovered and clarified as a project progresses, yet the acquiring organization wants a product delivered by a specific date that meets stated requirements and business needs.

From the perspective of the authors, a root cause of this problem is that Agile development methodologies, without adaptation, may not be compatible with procuring software systems from contractors. Remember that Agile, first proposed in 2001 via the Agile Manifesto [Beck 2015], is an alternative to development methods that emphasizes nailing down detailed requirements up-front with detail documentation throughout a project's life cycle. Challenges of adopting agile methods in the public sector can be found in the literature [Nuottila 2016], including papers that focus on systems acquisition and contracting [SEI 2016 and USGAO 2014].

Understanding these issues is critical to solving the problem of poor or failed system delivery projects. As such, this paper categorizes and describes challenges in quality & risk management from the perspective of an acquiring organization:

- Requirements & SDLC Congruency
- Organizational readiness;
- Contractual considerations;
- Business aware testing;
- Organizational dynamics

The authors believe a better understanding of these challenges would be the first step toward solving this problem. We will leverage lessons learned from the authors' experience in the public and the private sectors; with emphasis on improving quality, reducing risk, and minimizing technical debt in Agile or Agile influenced projects. Whether dealing with internal staff or with contractor staff, our main theme will be "trust but verify". As such, the reader can leverage lessons learned in each section depending on their specific context.

2. Context for Major IT Projects in Large Enterprises

A description of the context of major IT projects in large enterprises may be useful. Of particular importance would be the general solution approaches available and the challenges in large complex enterprises.

General Solution Approaches. Major IT projects are major capital investments for the acquiring organization. For each project, management makes its decision to invest, typically on the basis of a business case. Such a business case is usually supported by high-level business requirements and an analysis of available solution approaches in terms of their relative costs, benefits, and risks for the organization. Available solution approaches emphasize system integration and generally fall under three categories:

- Commercial-off-the-shelf (COTS) system with customization. This is usually based on a major software product or a suite of software products. Examples include enterprise resource planning (ERP) system or electronic document management system (EDMS) that have been implemented for similar enterprises in the same or similar vertical industries.
- Transfer system with customization. This is usually based on a system or a suite of systems that have been implemented for a specific enterprise.
- Custom developed system. This is often viewed as higher cost and higher risk, but may be the only viable approach if no COTS or transfer system based solution is available.

As an example, at any one time in Oregon state government, there are usually about 20 major IT projects in execution, with prime contractors on board and working. As of May 2018, only one of these projects selected a fully custom developed approach. The Top 5 projects by budget are in the following state agencies:

1. Department of Human Services - $335 million, transfer system with customization
2. Department of Justice - $130 million, transfer system with customization
3. Department of Transportation - $90 million, COTS system with customization
4. Oregon Health Authority - $28 million, COTS systems with customization

5. Department of Administrative Services - $20 million, COTS system with customization

The scope of customization may include integration or interfaces with existing or legacy systems, as well as conversion of data in systems being replaced. In addition, configuration, scripting, and custom coding are usually necessary. Even for a COTS or transfer system based solution, there may be significant coding efforts to deliver functionality not in the base system. As such, a "gap analysis" between the functionalities of the base system and the system to be delivered is of great importance during procurement and during development.

Challenges in Large Enterprises. Major IT projects are usually change initiatives that aim to significantly improve or transform an enterprise. For the acquiring organization, challenges include:

- Insufficient participation of subject matter experts - Cross-functional processes, business rules, and applicable regulatory requirements are often too numerous and complex for a few individuals to fully represent them. Yet, it is often impractical for the most qualified individuals to participate when needed, because these individuals may be indispensible in their normal duties.
- lack of familiarity with "to be" business processes or new operational paradigms - Personnel within the acquiring organization are experts of the organization's "as is" business processes, but these are the business processes being changed. For IT systems based on COTS or transfer systems, prime contractor staff are often the experts of "to be" business processes implicit in the base system.
- unobvious tradeoffs between cost / risk of customization vs. benefits that can be realized – Well meaning customization to support organization specific needs may have significant risks associated with cost, schedule, and other risks. On the other hand, exclusive use of functionalities "out of the box" may entail workarounds that are operationally time consuming or costly when in production use.

In the public sector, additional challenges that may be less common in the private sector include:

- competitive procurement processes - The prime contractor and the solution approach must be selected based on a fair, competitive request-for-proposal (RFP) process, because of public procurement statutes and rules. Such an RFP must be supported by requirements and a statement of work (SOW) that defines tasks, deliverables, and their acceptance criteria. Because these artifacts become exhibits of an executed contract, unclear requirements or SOW may face vendor questions or protests during the RFP process. More importantly, they may lead to contract disputes, cost escalation, or law suits later.
- lead time for contract review / approval - The RFP and the resulting contract and subsequent amendments must be approved by all participating funding partners, e.g. federal agencies in the case of state government. This review / approval process introduces significant lead time (usually many weeks) for contract amendments; typically contain changes to scope (requirements), schedule, and cost.
- Independent quality assurance and certification – The acquiring organization may need to pass independent reviews or be certified by one or more government agencies. Among Oregon state agencies, independent quality assurance by a third-party is mandated by state statute and, if applicable, federal agencies funding the work. There are also specific requirements when a system must connect to certain government systems, e.g. federal Social Security Administration, federal Internal Revenue Service, state division of motor vehicles, and federal or state law enforcement authorities. Systems for managing payments in federally supported programs, e.g. Medicaid and social services, must be approved and certified by the appropriate federal agency or agencies before production use.

Certain human factors and organizational dynamics are also of great importance, because they may negatively impact project performance in significant ways. For the prime contractor, it is driven more by profitability than the best possible solution for the acquiring organization. So, prime contractor staff would tend to do the minimum required by the contract or steer toward new tasks or contract amendments with additional costs to the acquiring organization. For the acquiring organization, it is not a given that the assigned staff want the change desired by management. They may not share the same vision of change as management or may interpret that vision differently. They may have a desire to minimize change of "as is" business processes and so resist alignment with functionalities of a base system "out of the box" (which would mean larger scope of customization). Also, they may not necessarily feel fully motivated or empowered in assigned work teams.

3. Requirements & SDLC Congruency

From a software engineering standpoint, requirements are central to high quality system delivery. However, before the acquiring organization can have a prime contractor on board, one must be procured that is capable of delivering the system – often based on a COTS or transfer system, as discussed in Section 2.

Requirements for Procurement vs. Requirements for Development. For the acquiring organization, it is very important to differentiate between the following:

1. High-level requirements necessary to procure the prime contractor including competitive RFPs;
2. Detailed requirements necessary to support design, development, and implementation.

For (1), an acquiring organization must develop high-level requirements that describe strategic intents, business objectives, general features-functions needed, conceptual relationships with other data systems and data sources, and high level workflows and user scenarios. Frequently, this work may involve consultants or contractors. If contractors are involved, these should not be the prime contractor to be procured, as it represents a conflict of interest.

For (2), detailed requirements are usually done by the prime contractor, with participation of the acquiring organization's management and staff. This work may involve development of user scenarios or use cases, analysis of features-functions needed to meet business needs, modeling and validation of business processes as needed, and documentation to support the work of cross-functional teams and third-party reviews as needed. This type of work is accompanied by solution architecture based on software products chosen to support the general solution approach selected, as described in Section 2.

Requirements That Fail to Represent All Business Needs. A "technically good" system does not necessarily guarantee that the larger mission or objectives of the enterprise can be achieved. But even SDLCs such as Agile that incorporate the users into the development process do not necessarily guarantee that the delivered systems support the overall objectives of the enterprise. This is true especially for very large projects in large enterprises in which not all users or the relevant subject matter experts can be represented when needed. Indeed, a central challenge to the acquiring organization is in assuring requirements are sound, faithfully reflect business needs, and can be expected to give rise to features-functions that have reasonable return on investment (ROI), both in financial or non-financial terms.

Iterative Requirements. With good engineering and project management practices, subsequent work (Design, Build, Deploy, and Operate / Optimize a system) can be done on the basis of good traceability to Requirements with the participation of the end user or product owner. In the schematic SDLC representation of Figure 1, Waterfall SDLC [Jones 2012] may well have arrows from Build back to Design or from Design back to Requirements, but the general directionality is Design after Requirements and Build after Design. Agile and other iterative SDLCs (e.g. the Unified Process) would leverage the learning achieved from earlier software releases to enable refinements of Requirements and Designs in subsequent phases, iterations, or sprints.

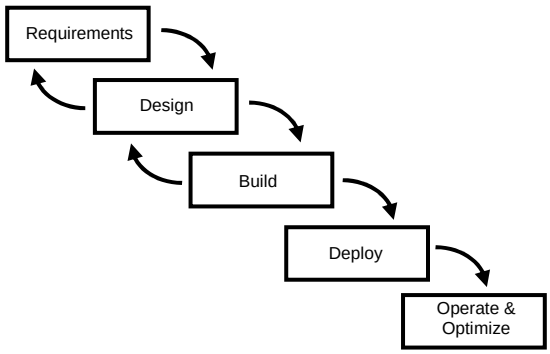

Figure 1: Generic System Development Life Cycle (SDLC) for a single phase, iteration, or sprint in a project. A project with multiple phases, iterations, or sprints has multiple instances of this.

Where there are multiple major software releases and numerous minor software releases to be delivered over many iterations or sprints, it is important for the acquiring organization to be proactive with reviews of user scenarios, use cases, detailed requirements, and testing. It is necessary to adopt a "trust but verify" mindset when working with the prime contractor throughout the project's life cycle.

With contractors that develop software using an Agile methodology, yet with a waterfall type contracting vehicle, it is critical to allow extra time for the iterations in the requirements phase where the contractor involves the acquiring organization's project management team and the end user organization. As shown in Figure 1, this can be accomplished through iterative Requirements, Design and Build. These iterations can be time boxed with reviewable artifacts in order to support a controlled process that is part of enforceable contracts.

4. Organizational Readiness

In many acquiring organizations, the capture, analysis, and documentation of detailed requirements are increasingly in scope for the prime contractor. However, even when high quality resources to support all facets of a project can be expected from the prospective prime contractor, it does *not* follow that the acquiring organization can bring a prime contractor on board at any time without substantial preparation. The extent of this preparation varies depending on policies for IT investment due diligence, competitive procurement, budget cycles, and other considerations in the acquiring organization; as well as the level of management discretion allowed and accountability expected. More often than it should, the authors have seen that insufficient work is done by management and staff of an acquiring organization in preparation of the on-boarding of the prime contractor. The important lesson learned is: regardless of the quality of the prime contractor, few projects can expect to be successful without certain preparation to assure organizational readiness. This preparation includes:

- *Project Charter & Governance*. All major IT investment should put in place a project charter that identifies purpose of a project, executive sponsors, steering committee members, a project manager, and a project management team.
- *Business Case*. A major IT project is a major capital investment and should have a carefully prepared business case that identifies business drivers and justifications for undertaking the project. The business case should contain an analysis of solution alternatives that identifies their relative benefits (both financial and non-financial), costs, and risks. It should conclude with the solution alternative selected, key success factors, and risks.
- *Detailed Planning Artifacts*. Relevant artifacts include: requirements and statement of work to support procurement, a high-level project plan, communications plan, and other supporting plans.
- *Procurement Process*. Proposals are solicited using formal RFP or other processes. Proposals received are evaluated, possibly using predefined evaluation / scoring criteria. This is followed by contract negotiation with the apparent winner of the competitive process, legal reviews, and contract execution.

Note that when working with a contractor who is using an Agile development process, communications plan and other supporting plans need particular attention. Agile requires frequent communication -- sometimes on a daily basis. As part of the planning documents, processes need to be put in place for the contractor and the acquiring organization (including end users who will ultimately be using the system) to have frequent interaction and collaboration.

5. Contractual Considerations

From the perspective of the acquiring enterprise, project success is at least partly tied to the effectiveness of the contract. In this respect, it is important for the acquiring enterprise to consider the following:

- Procurement models;
- Terms & conditions including penalty or liquidated damage clauses;
- Degree of contract modifications anticipated during the project's lifecycle.

Procurement Models. Procurement models primarily refer to whether the contract will be deliverables based or time-and-materials based. If the requirements are well understood, stable, and can be well documented, a deliverables based contract is most appropriate; but this is not usually the case. Therefore, many enterprises (especially in the private sector) may feel that time-and-materials contracts are more easily set up and administered. This model is more supportive for Agile methods, because there is no need to have detailed requirements completely specified in the beginning of the project, allowing for discovery along the way. However, the acquiring enterprise must be willing to accept the risk associated with contractor non-performance and potentially flawed time and cost estimates.

On the other hand, a deliverables based contract does not give complete protection against cost overrun. This is because requirements may not be sufficiently detailed, and enhancements of requirements as the project progresses frequently entail additional costs. Also, the prime contractor would have included a price premium or slack in its cost proposal to deal with uncertainty or error in estimation.

This situation contrasts with a time-and-materials contract in which there is no specific upfront assurance of the cost and time required to deliver a specific set of features-functions. In short, there are positive and negative for either approach. What we have found that works is a combination of the two approaches where a time boxed iterative approach is used, but with specific deliverables after multiple iterations. The iterative approach in the beginning of the project allows for requirements discovery while both sides develop a better understanding of

everything from the requirements to the technical difficulties involved. This is followed by specific milestone based deliverables that enable work in progress to be reviewed iteratively.

Terms & Conditions. Terms and conditions in the contract that awards performance (such as early completion) could be a useful incentive for the contractor; as are clauses that delay, retain, or otherwise reduce payment due to non-performance (such as late completion). Especially important would be payment terms (e.g. Net 30 upon invoice), provision for progress payment for deliverables work in progress, retainage (e.g. 15% holdback on all accepted deliverables invoiced until after User Acceptance Test (UAT) and formal acceptance of the system). This makes the UAT a critical part of any contract. More on UAT will be discussed in the next section.

Contract Statement of Work (SOW). Considerations for developing a SOW include the following:

- Deliverables need to have well defined acceptance criteria. In other words, the contract should specify what is included and not included in each deliverable.
- There is a need for contract language that balances specificity with flexibility to enable dynamic response to certain project unknowns, e.g. approved scope expansion or schedule changes.
- For requirements that may not be stable (e.g. requirements that have not been thoroughly vetted or rapidly changing business conditions), it may be necessary for the acquiring organization to budget extra funds for on-demand work by the contractor, through change requests, task orders, or contract amendments.
- The SOW needs to be consistent with the SDLC anticipated. If the contractor and acquiring organization agree to time boxes or an iterative process, the SOW and contract should reflect this directly.

The last point requires further discussion. Unfortunately, many organizations use SOW templates that assume a sort of “big bang” Waterfall SDLC, with major deliverables assumed to be a single instance of the detailed requirements, architecture, design, or testable software release. Referring to Figure 1, the contract / SOW template may implicitly assume a single unidirectional pass through the Generic SDLC model. From the perspective of the acquiring organization, this approach may be the result of historical practices, or it may seem “simpler” to have fewer contractual deliverables. When the contractor is using Agile methods, this practice is extremely problematic in two ways:

- First, it may encourage the prime contractor and the staff of the acquiring organization to actually adopt a “big bang” Waterfall SDLC; with all the schedule, budget, and scope / quality risks associated with delayed identification of defects in requirements and designs.
- Second, it may create mismatch of expectations around key deliverables (e.g. detailed requirements, architecture, design, and testable software releases); as far as what specific work products can be reviewed by the acquiring organization by what time. This is especially problematic in federally or state government funded projects in which independent quality assurance or independent verification & validation (IV&V) by a third party may be mandatory.

The overall result is a mismatch of development methodology with deliverable and expectations. This is a recipe for trouble. In summary, you cannot expect to have fixed timelines and deliverables without detailed requirements, and you cannot expect requirements not to change or at least be elaborated. Make sure the contract structure and deliverables are in alignment with expectations in SDLC and development methodology.

6. Business Aware Testing

Owing to the real-world challenges described in Section 2, major IT projects in large enterprises, including those that use iterative or Agile SDLCs, may discover defects in functional requirements late in the project’s life cycle -- perhaps as late as User Acceptance Testing (UAT). This situation is not necessarily the fault of the prime contractor and the specific choice of SDLC. These types of defects are costly to fix and represent major risks to quality, schedule, and cost. To reduce defects at the time of system launch, we introduce the term Business Aware Testing. Business Aware Testing emphasizes testing from the business point of view, not only from the end users, but also from the acquiring organization’s business needs.

A useful model for visualizing testing is the “V-model” that relates different types of testing to different aspects of the system, as depicted in Figure 2.

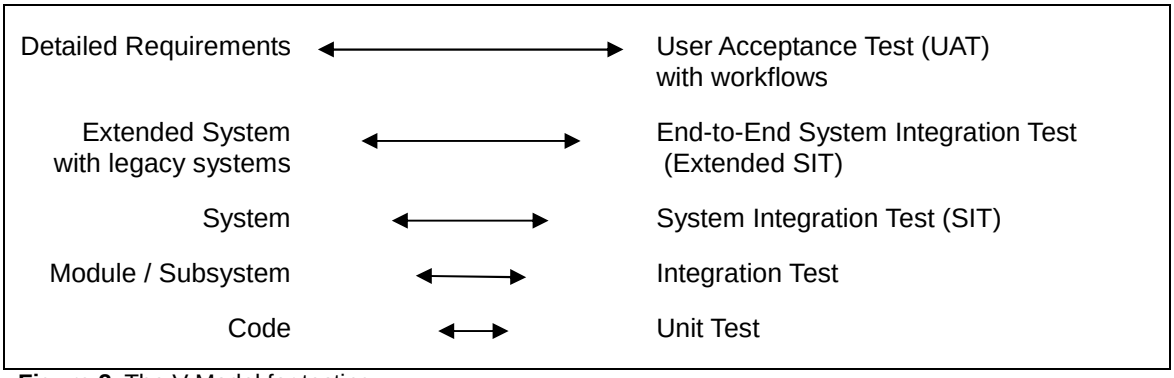

Figure 2. The V-Model for testing.

A good prime contractor would have planned for ample testing activities; especially in unit test, integration test, and system integration test. In the spirit of "trust but verify", it is important for the acquiring organization to organize a testing effort of its own, separate and independent from the prime contractor and their development and testing teams. When doing this, the following considerations are important.

Testing by Type. For the acquiring organization, there are three types of testing that are especially important conceptually:

- Functional Testing: features-functions, user scenarios, use cases, workflows, etc.
- Non-Functional Testing: performance, load, stress, etc.
- Regulatory Compliance Related Testing: connectivity to government systems, certification, attestation, etc.

In the context of functional testing, there are more specialized types of testing that tend to be a mixture of the three types of testing identified above:

- Security Testing
- Data Conversion Testing
- Accessibility Testing
- Extended System integration Testing (with interfaces / integration with legacy systems)
- User Acceptance Testing

Iterative / Agile SDLC with Waterfall Overlay. To assure accountability and in the spirit of "trust but verify," it is important for the acquiring organization to insist on testable code releases from the prime contractor that the acquiring organization can independently develop test scripts and run tests that can be traced to known detailed requirements. The authors believe a waterfall like overlay on the prime contractor's iterative or Agile SDLC, called "IterFall", would be advisable for accountability. See Figure 3 below.

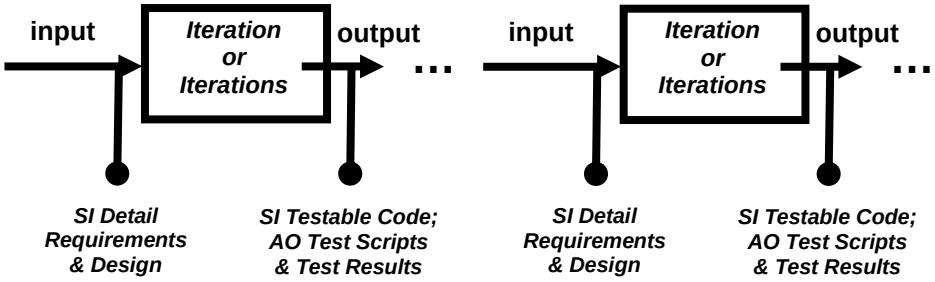

Figure 3. Iterative SDLC with Waterfall Overlay, called "IterFall" by the authors. In this figure, one or more iterations (sprints) would produce a software release that the acquiring organization or a third-party can test. Here, "SI" denotes system integrator (prime contractor) and "AO" denotes acquiring organization.

Entry & Exit Criteria. For the acquiring organization, User Acceptance Tests (UAT) and Pilot of a major system involve coordination between different parts of the organization. Having well defined entry and exit criteria are not only important for mutual understanding between the acquiring organization and the prime contractor, they are important for the acquiring organization to effectively communicate with internal stakeholders and to manage the participation of staff and contractors resources in UAT and related activities. Examples of typical entry and exit criteria from a major IT project of the state of Oregon are given in Figure 4.

Project Time Frame	Typical Entry Criteria	Typical Exit Criteria
System integration Test (SIT) or Extended SIT	• As determined by prime contractor.	• Prime contractor completed all SIT test cases • Level 1 Defect = 0 and Level 2 Defect = 0
User Acceptance Test (UAT)	• SIT Exit Criteria met • Acquiring Organization's test cases ready	• Level 1 Defect = 0 • Level 2 Defect = 0 or as agreed with prime contractor • Shape Metric requirement met (see below)
Pilot	• UAT Exit Criteria met and Federal approval to start Pilot received • Data conversion completed & checked • Server recovery exercise complete at hosting facility • Business / people ready, especially training • Rollback strategy complete	• Level 1 Defect = 0 • Level 2 Defect = 0 or as agreed with prime contractor • Shape Metric continues to trend downward (see below)

Figure 4. Sample entry and exit criteria for SIT / Extended SIT, UAT, and Pilot in a state of Oregon major IT project. "Level" in this table refers to severity level of a defect. (See text below.)

In Figure 4, "Level" refers to severity level, which classifies defects in terms of their impact on overall system operations. This table assumes four levels of severity: Level 1 = critical software defects / total application failure, Level 2 = major software defects, Level 3 = minor software defects, and Level 4 = cosmetic defects that do not affect operation [Jones 2012]. In a contract, severity levels may need more precise operational definitions, in order to avoid disputes. It turns out dispute around assignment of defect severity level is common during UAT, with the prime contractor pushing for lower severity level than that assigned by the acquiring organization's testers. This is understandable as earlier acceptance means earlier payment is possible from the prime contractor's perspective. Within the acquiring organization, there may be enormous pressure to accept the system so it can be launched and the project closed – sometime before system quality has stabilized, unfortunately.

Empirical Shape Metric for Defect Counts. In order to understand if system quality has stabilized, it is important to track the temporal trend of total defects discovered by testing against some specific set of test scripts or test cases. Figure 5 depicts schematic defect trend in time that would need to be observed before a system can be considered production worthy for applications that are considered mission critical and/or must conform to strict regulatory compliance.

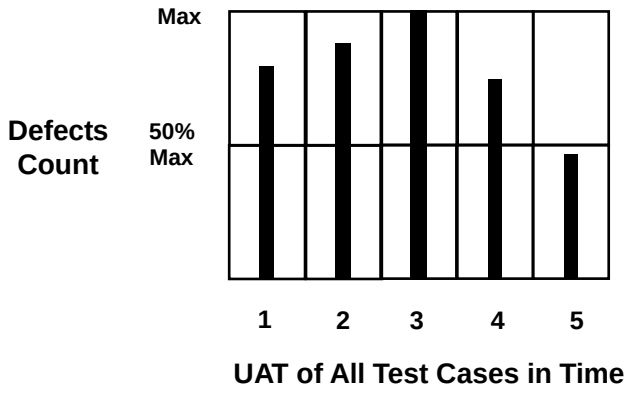

Figure 5. Sample Defects Shape Metric, which is a time series of Defects Count when a fixed number of test scripts are run multiple times.

It is possible to develop contract language to formalize understanding around defects shape metric between the acquiring organization and the prime contractor. Below is sample language from an executed contract between the state of Oregon and a major system integrator:

1. There will be at least five data points gathered during UAT. Agency will designate data points in its UAT Test Plan;
2. Defects discovered at the final data point must be less than or equal to 50% of the Time Series Maximum. "Time Series Maximum" is the data point representing the highest number of Defects discovered during the execution of the UAT test cases designated as data points;
3. There must be at least two data points following the Time Series Maximum; and
4. Data points must show that the total number of Defects is declining.

Lessons Learned From Testing. Some of these lessons learned may seem like common sense. However, there is a pattern of history repeating itself that the authors have observed. One reason is organizational dynamics and politics that sometime motivate premature system launch. Another reason may be general lack of awareness of certain types of risks among staff and management of the acquiring organization or the prime contractor.

- Systems should not be placed into production when there are critical defects that require complex workarounds or have no workaround.
- Empirical defects trends are important for go-live decisions.
- Interfaces to legacy systems and integration with other existing systems may be complex and high risk. Time and time again, interfaces with existing / legacy systems have delayed a project's ability to move to pilot or to go-live enterprise wide. Interfaces and integration with existing systems must be tested.
- Data conversion (with shadow data repositories that often lack good inventory and documentations) may be complex and high risk. Time and time again, data conversion has delayed a project's ability to move to pilot or to go-live statewide. Data that need to be converted must be identified early, and converted data must be checked.
- Security testing and reviews need to be done by trained specialists that apply appropriate test protocols, methods, and tools. Applicable standards may be industry specific or mandated by relevant government entities [End Note 1]. Also relevant are standards, policies, and procedures of the acquiring organization.
- Disaster recovery plan needs to be in place, tested, and validated.
- Regression testing needs to be done after bug fixes. Lack of test automation is a challenge against sufficient regression test or even "smoke test" after bug fixes or software upgrades; which include upgrades or patches of underlying COTS products / platforms, middleware, and operating systems.

7. Organizational Dynamics

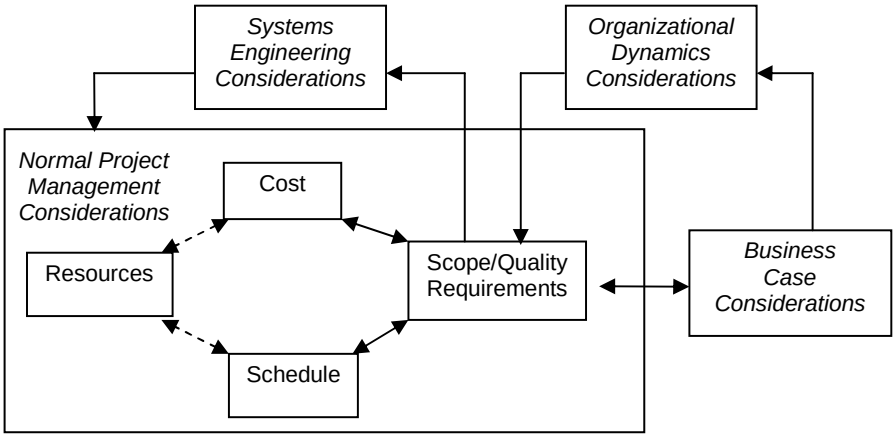

Figure 6. Project management considerations are not simply the management of the triple constraints of scope, cost, and schedule, based on system engineering considerations. Scope / quality requirements are affected by changes in business case considerations and are susceptible to the influence of organizational dynamics.

Referring to Figure 6, "normal project management" is predicated on the management of a baseline project plan, which is the science and art of balancing the triple constraints of scope, cost, and schedule. When the business case and "to be" system requirements are stable, the challenge of balancing the triple constraints is mainly influenced by system engineering considerations. However, a stable business case with corresponding stable requirements is not a valid assumption in many projects, especially those that aim to transform operational paradigms or other important aspects of the enterprise. In these situations, requirements are susceptible to the influence of organizational dynamics. This means human factors of individuals and groups -- operating in the context of the organizational culture of the enterprise -- play an important role in shaping and morphing the requirements. Some lessons learned regarding organizational dynamics are:

- Projects need to deal with enterprise-wide paradigm change early, especially with respect to business process change, staffing change, and other changes that are in conflict with prevailing organizational practices or culture.
- Projects need to adapt and adjust scope / quality requirements with evolving changes in the business and external environment that may affect the business case; not ignore or resist them on the ground of scope creep or change control.

There is a fundamental tension between change management from the perspective of project management (which emphasizes change control) and organizational change management (which emphasizes necessary and beneficial change to the enterprise). The acquiring organization must balance change to project scope, schedule and budget with change needed by an enterprise, but the latter frequently requires time and organizational learning to develop and to refine [Crawford 2014]. In Agile, as in other SDLCs, a major IT project in a large acquiring organization must [Sills 2017]:

- develop project roadmaps that integrate relevant business functions and IT;
- put in place project governance that integrates all work teams and relevant business functions;
- coordinate cross-team / cross-function dependencies methodically and carefully;
- assure end-to-end functionality of work products across all work teams and relevant business functions.

Given organizational culture, it is not a given that all enterprises can do these things well to properly support Agile or, for that matter, another SDLC chosen to deliver a major IT project. In other word, SDLC is just one facet of success in delivering a major IT project in a large complex enterprise.

8. Conclusion

In this paper, we have examined various aspects of assuring good quality & risk management for large organizations when acquiring enterprise systems, with emphasis on the use of Agile methods. In this context, an acquiring organization needs to pay particular attention to:

- how requirements are to be discovered and refined, given the general solution approach chosen;
- organizational readiness before the prime contractor should come on board;
- alignment between contract and SDLC, especially in relation to ongoing verification & validation of prime contractor work products;
- business aware testing, especially in relation to end-to-end functional testing and defects management during UAT;
- human factors and organizational dynamics that may cause real-world circumstances to deviate significantly from those assumed by life cycle reference models and processes, especially those that assume stable requirements and business case as given.

When working with a prime contractor, it would be important to adopt a “trust but verify” mindset during all aspects of a project. A project that aims to bring about major business process changes and operational paradigm shifts [End Note 2] is especially challenging and need to review business case and requirements frequently with those participating in project governance. Real-world complexity – including but not limited to availability of subject matter experts, contracting models, resources for end-to-end functional testing, and organizational dynamics / culture -- may render Agile (or other SDLCs for that matter) not a good fit for the acquiring organization.

Large IT projects in large enterprise are among the most complex human endeavors. A viable solution that meets business needs requires good cross-team / cross-functional coordination that takes into account human factors and organizational dynamics. There is no "silver bullet" for successful delivery of major systems. The choice of SDLC, including Agile if chosen, is one facet of good management and systems engineering / systems integration.

References

[Beck 2015] K. Beck, M. Beedle, A. van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, R. C. Martin, S. Mellor, K. Schwaber, J. Sutherland and D. Thomas, *Manifesto for agile software development*, June 2015. Available: http://www.agilemanifesto.org.

[Crawford 2014] L. Crawford, A. Aitken, and A. Hassner-Nahmias, *Project Management and Organizational Change*, Newtown Square, PA: Project Management Institute, 2014, ISBN: 978-1-62825-043-5, page 1-3.

[Jones 2012] C. Jones and O. Bonsignour, *The Economics of Software Quality*, Boston: Addison-Wesley, 2012, ISBN: 978-0-13-258220-9. See page 246 for relevant paragraphs on Waterfall SDLC and page 281-282 for relevant paragraphs on defect severity levels.

[Nuottila 2016] J. Nuottila, K. Aaltonen, and J. Kujala, "Challenges of adopting agile methods in a public organization," International Journal of Information Systems and Project Management, Vol. 4, No. 3, 2016, page 65-85; and references cited therein. Available: http://www.sciencesphere.org/ijispm/archive/ijispm-040304.pdf.

[Sills 2017] D. Sills, K. Tunks, and J. O'Leary, "Agile in Government – Scaling Agile for Government," *Agile in Government: A playbook from the Deloitte Center for Government Insights*, Deloitte Development LLC, 2017, page 8-13. Available: https://www2.deloitte.com/insights/us/en/industry/public-sector/agile-at-scale-in-government.html.

[SEI 2016] Software Engineering Institute, "RFP patterns and techniques for successful agile contracting," Report No. CMU/SEI-2016-SR-025, November 2016. Available: https://resources.sei.cmu.edu/asset_files/SpecialReport/2016_003_001_484063.pdf

[USGAO 2014] United State Government Accountability Office, "Healthcare.gov – ineffective planning and oversight practices underscore the need for improved contract management," Report No. GAO-14-694, July 2014. Available: https://www.gao.gov/assets/670/665179.pdf.

End Notes

[End Note 1] For examples, see: https://www.pcisecuritystandards.org/pci_security/standards_overview for an overview of payment cards industry security standards; http://owasp.org for Open Web Application Security Project (OWASP) which provides a Testing Guide; https://www.hhs.gov/hipaa/for-professionals/security/index.html for HIPAA Security Rule that applies to the U.S. healthcare industry; and https://fedramp.gov for security standards of cloud services used by U.S. federal government agencies.

[End Note 2] The term "paradigm shift" originated in philosophy of science. See T. Kuhn, *The Structure of Scientific Revolutions, Second Edition,* Chicago: The University of Chicago Press, 1970. Besides the physical sciences, this term has been widely used in different fields (e.g. technology, marketing, and social sciences) to mean a major change in thought pattern or approach of individuals or organizations.

Acknowledgement

The authors acknowledge helpful discussions with (in alphabetical order of last names): Edward Arabas, Nicholas Betsacon, Bob Cummings, Capers Jones, Patricia McQuaid, Alex Pettit, and Wallace Rogers.