The Right Skills, at the Right Time, *at the Right Price*

Mitigating Project Risk

By: David Griffiths and Sergio Frank

Achieving System Quality through Strategic Test Management

By: David Griffiths and Sergio Frank

ORACLE Gold Partner

Microsoft CERTIFIED Partner

Table of Contents

Mitigating Project Risk

Achieving System Quality through Strategic Test Management

Table of Contents

Mitigating Project Risk

Abstract

Risk management is the recognition, assessment, and control of uncertainties that may result in schedule delays, cost overruns, performance problems, adverse environmental impacts, or other undesired consequences.

A risk management strategy is an integral component of a comprehensive project plan and controlled project delivery. The overall goal of a risk management strategy is to progressively reduce a project's exposure to events that threaten accomplishment of its objectives by incorporating approaches into the project plan that minimize or avoid identified risks; developing proactive, contingent risk response actions; and implementing rapid risk responses based on timely identification of risk.

A comprehensive model for risk identification will incorporate people, processes, and technologies. It will operate within the project charter, project scope, and project budget.

This whitepaper discusses how to identify, classify, and assess risks, then put together a comprehensive strategy and plan with processes to effectively manage, monitor, and respond to risks.

Risk Identification & Classification

In performing risk identification, the project team should evaluate a wide range of areas in order to comprehensively identify risk candidates. Some areas may be unique to a particular project, organization or technology; others may be more ubiquitous.

Risk identification should incorporate, but not be limited to, evaluation of business requirements, organizational factors, team capabilities, existing knowledge, technology reliability, project constraints, and project management. See Exhibit A for sample questions that may be used to identify possible risks. See Exhibit B for common project risks.

Risks may be grouped into the categories outlined below. It is possible that risks can span multiple categories. Risk categorization allows the project team to view and address risks based on what portion of the project may be affected.

Cost-based risks outline the financial goals of the project detailed in project objectives or key success factors. Typical cost risks include external contractor overspend and additional costs in changing/solving design, application project, or operational problems.

Schedule-based risks take into account the products or benefits within the specified time frame. Typical schedule-based risks arise from extensions from scope changes, resource availability, market opportunities missed, and additional schedule extensions from solving those risks outlined in 'cost-based risks' above.

Performance-based risks consider the application specifications and expected benefits.

Typical risks include functionality, system operability, integration problems with other existing systems, migration problems, performance expectations not achieved, and environment complexity.

Quality & Project management-based risks consider the impact of personnel shortfalls, unrealistic schedules and budgets, internal performance shortfalls, and performance shortfalls in externally performed or subcontracted tasks. Project management risks may also arise from inadequate performance measurement, lack of communication between teams, lack of discipline in record keeping, delivery tracking, and responding to follow-up actions pertaining to remediation of risks.

Operational-based risks focus on the peripheral organizational and business operational re-engineering changes arising from system development. Typical risks consider both the transitional and the long-term effects of the system's introduction, including the organizational and behavioral change and the human and physical resource planning and communication required to facilitate a smooth transition to the new system, business processes, and tools.

External-based risks consider the environmental factors largely outside of the control of the project management team which can directly/indirectly affect the successful delivery of the individual projects. Typically, risks arising from legislative regulations, legal requirements, communication to the market, lack of market sophistication and strategic direction, and priority conflicts of a controlling body are profiled under this category.

The Cost and Schedule risk sources are known as the risk 'indicators,' as they are often the most tangible measure of overall progress towards program/project objectives or goals. The Quality & Project Management, Technological, Operational, and External risk sources are referred to as risk 'drivers,' as these are the sources of project risks, which additionally drive the Cost and Schedule risks.

Risk Assessment

A risk response strategy is dependent on performing a qualitative and quantitative assessment of risks. This extends the value of the understanding, documenting, and reporting on project-level risks by assigning each risk to a category rating scale. It introduces a common format to the assessment of risk, based on easily understood adjective descriptions. These assist in realizing and focusing on the 'true' impact of each risk and the prioritization of the risk-reducing activities and responses identified.

The probability and impact are combined to calculate a numerical measurement of exposure for each risk.

The Probability (P) is an assessment of the percentage probability of an occurrence of the risk. This is rated on the following scale:

A **Low probability** risk has a 0-33% probability of occurrence, and is assigned a value of 1. A **Medium probability** risk has a 34-65% probability of occurrence, and is assigned a value of 2. A **High probability** risk has a 66-99% probability of occurrence, and is assigned a value of 3.

The **Impact** is an estimate of the overall consequences to the project should the risk occur.

A **Low impact** risk would have a slight effect on progress with moderate extensions to schedule and cost across short and medium term if it were to occur, and is assigned a value of 1. A **Medium impact** risk would cause significant disruption to project schedule, cost, and products over the medium and long term if it were to occur, and is assigned a value of 2. A **High impact** risk would cause significant disruption to successful delivery of project objectives, products, and benefits were it to occur, and is assigned a value of 3.

The two major variables used in classifying a risk are the Probability of the risk occurring and the Impact that the risk will have if the risk occurs. When the two variables are multiplied, the product calculates the **Risk Index**.

Risk Index					
Risk Index (RI)			Probability (P)		
			Low	Medium	High
			1	2	3
			0-33%	34-65%	66-99%
Impact (I)	High	3	3	6	9
	Medium	2	2	4	6
	Low	1	1	2	3

Risk Index (RI) = Impact (I) * Probability (P)

Figure 1. Risk Index

The table in Figure 1 allows the project management team to take an objective view on how to address each risk being tracked once the Risk Index has been calculated.

If a risk falls in the green area (RI = 1 or 2), it is categorized as low risk and requires monitoring. If a risk falls in the yellow area (RI = 3 or 4), it is categorized as medium risk and requires a plan to mitigate the risk be prepared in case the risk increases in probability or impact. If a risk falls in the red area (RI = 6 or 9), it is categorized as high risk and requires that active steps must be taken to prevent its occurrence.

As risks are identified, assessed, and measured, the project management team determines which risks are important enough to manage. As the qualitative risk analysis evaluates each risk and designates it as high, medium, or low, the project team determines the correct response strategy for each risk.

Risk Response Strategies

The goal of creating a response strategy for each identified risk is to take action that will allow the project to achieve its goals even if an identified risk occurs. The response strategy effectively makes the impact of the realized risk “less severe or painful.” Risk response strategies can include proactive measures intent on lowering either the probability or the impact of the risks, fully reactive measures that are set into motion after a risk is realized, or a combination of the two. Standard responses and approaches are outlined below.

The initial steps in the risk response strategy consider those risks that may occur the earliest in the development lifecycle, irrespective of probability. This is intended to cover any short-term exposure first, before considering overall project risk reduction.

In situations where there are no feasible preventative strategies, project management may choose to plan and implement a solution that will be deployed should the risk occur. As with preventative actions, this is a proactive approach that requires the commitment of resources prior to the realization of the risk. Contingency activities do nothing to reduce the probability of the risk but craft a pre-built fallback solution that can be rolled out on short notice, thus reducing the impact that the risk may have on the project.

Overall, project risk response strategies cover the following characteristic responses. These are all valid risk response strategies. Depending upon the nature of the risk, importance, and impact, the appropriate risk response may be a contingency plan, avoidance, acceptance, control, transfer, or investigation.

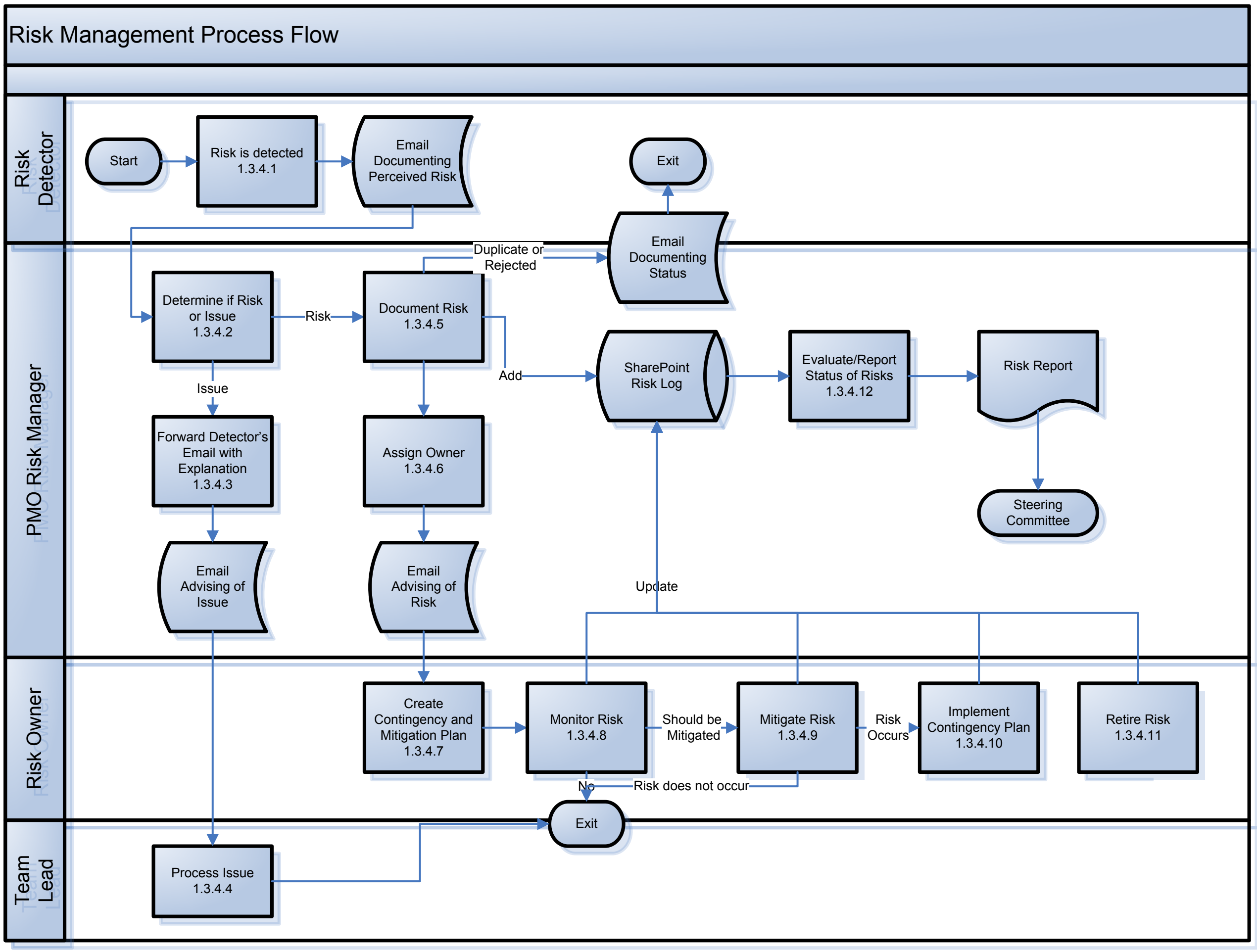
Often risks are seen as neither having enough impact nor being likely enough to justify committing significant resources to active mitigation or prevention activities, but are important enough to warrant foresight and planning. In these cases, the project management team should determine ahead of time the actions to take should the risk become reality.

> **Avoidance-based responses** are employed at any point in the development lifecycle where future-planning work is performed. Typically, most risk avoidance occurs during the project definition and planning phases where objectives, scope, key success factors, work breakdown, and project outputs or deliverables are defined. However, risk avoidance solutions may limit the ability to achieve high-level project objectives by unnecessarily constraining a desirable solution.
>
> **Control-based responses** occur at all points throughout the project lifecycle and are typically the most common response. They identify an action or product that becomes part of the project work plan and are monitored/reported as part of the regular performance analysis and progress reporting of the project.
>
> **Transfer-based responses** target the party who is best placed to analyze and implement the response to the risk, based on their expertise, experience, and suitability. Typical transfer responses include sub-contracting to specialist suppliers who are able to reduce the overall risk exposure.

Investigation-based responses do not define any approach for reducing an individual risk. They are responses to risks where no clear solution is identified and further research is required. However, investigative responses will not be ignored, as they immediately and directly lead to a greater aggregated project risk.

Risk Management Processes

The steps in the risk management process are outlined below.



Risk is Detected

A risk is detected by a Risk Detector (project team member or stakeholder) and brought to the attention of the Risk Manager (part- or full-time project role) through an email. Risks are initially captured during the start-up of the project. Additional risks are captured on an ongoing basis as the project progresses.

Determine if Risk or Issue

The Risk Manager should work with the Risk Detector to determine if the item is an issue or a risk. If it is an issue, the risk process is exited. The Risk Detector has the further responsibility of documenting the issue and passing it to the appropriate Team Lead designated by the Risk Manager. If it is determined that the item is truly a risk, the Risk Manager will work with the Risk Detector to assure that the risk is adequately documented.

Forward Detector's Email with Explanation

The Risk Manager will notify the Risk Detector that the perceived risk was actually an issue. A detailed explanation will be provided.

Process Issue

Upon receipt of the email informing him/her that an issue has been detected, the appropriate Team Lead should process the issue in the same fashion as normal.

Document Risk

The risk is documented in the Risk Log. The Team Lead should use the Risk Assessment to determine Impact and Probability and calculate the Risk Index.

Assign Owner

Risks received are reviewed by the Risk Manager to determine if they are truly “new” risks and not a repeat of a current one. In addition, the Risk Manager determines if the risk is worthy of logging. The Risk Manager assigns a Risk Owner, which will typically be a Team Lead. The assigned person is responsible for monitoring the risk and developing a contingency and mitigation plan. The Risk Manager sends an email to notify the Risk Owner.

Create Mitigation and Contingency Plans

The Risk Owner is responsible for researching the risk and developing a contingency plan and a mitigation plan. Mitigation strategies focus on ways of reducing the impact and the probability of the risk. Contingency plans outline what the project will do should the risk occur. The effort and cost to create the mitigation and contingency plans are commensurate with the risk exposure and are created by the owner in consultation with the project management team.

Monitor the Risk

The Risk Owner is responsible for monitoring the risk for any changes in probability, impact, or timeframe in order to determine if the risk will have to be mitigated. Any changes to impact or probability should be noted in the risk log.

Mitigate Risk

If it is determined that the risk should be mitigated, the Risk Owner will implement the mitigation plan. The factors that would indicate the need for mitigation would be that the timeframe is getting closer, the probability of occurrence has increased, the impact of its occurrence has increased, or the project management team directs the elimination of the risk. If the decision is made to not implement the mitigation plan at this time, the risk is returned to the monitor step. Any changes to impact or probability should be noted in the risk log.

Implement Contingency Plan

If the risk occurs despite the mitigation plan, the project team should implement the contingency plan to recover. If the risk does not occur and the effective time frame has passed, the risk should be retired. The risk log should be updated to reflect the latest status.

Retire Risk

In collaboration with the Risk Manager, the risk should be retired when it has not occurred and all signs of danger have passed or when the risk has occurred and a contingency plan has been successfully put into place. The risk log will be updated to reflect the latest status.

Evaluate/Report Status of Risks

In the (weekly) update provided to the steering committee and the (monthly) steering committee meetings, the risk log should be summarized and risks with the highest risk indexes discussed.

Risk Monitoring and Summary Reporting

Systematic risk monitoring will provide the project with assurance that established controls are functioning properly.

The risk log should be updated in real time by either the project management team or a particular Risk Owner. The Risk Manager should review and, if necessary, update the risk log weekly and address risks proactively during team status meetings. The project will continuously adjust its response strategy based on changing events, impacts, risks, and the success/failure of the response strategies implemented.

Every month, the Risk Manager should review each newly added risk with the project team to determine if there has been a change in the probability or impact of that risk's occurrence.

Once per quarter, the project leadership team should review the risk log and highlight risks that were identified but not realized. These risks should be specifically reviewed in the weekly status meetings with project leadership to determine if the risk should continue to be tracked or if the risk can be retired.

As a part of the project governance process, key risk areas should be measured and reported in project scorecards on a monthly basis.

As a part of the weekly review of risks, a risk summary report should be prepared and reviewed. This report tracks assessed overall risk, total risk exposure, and average risk exposure across all risks open for the project. Trends are also presented to allow the management teams to determine if the relative probability or impact of risks is increasing and needs further attention, or decreasing, indicating a more favorable risk profile.

Risk Escalation

Risks may be escalated as required during the risk management process. They may be escalated to inform other levels of management or other teams of risks and response plans or in order to consult others to assist with the development of appropriate risk response strategies. They may also be escalated when the profile of the risk changes (becomes higher impact or the probability of the risk occurring increases) or previous risk responses have not had the desired effect.

Risk escalation should follow a hierarchy through the project team up to the Steering Committee, such as:

- Delivery project team members to their project management
- Project management to the PMO (if applicable)
- PMO to the steering committee

Any risk owner may make the recommendation to escalate a risk. When escalated, the risk owner will not change.

Exhibit A – Sample Risk Identification Questions

Requirements

- Are the requirements consistent with strategic objectives and goals?
- Are the requirements technically feasible?
- Is there consensus among the stakeholders about the requirements?
- Are the requirements subject to significant change based on external developments?

Organization

- Have all stakeholders been identified?
- Is the work that must be done high profile?
- Can the project goals be met within the desired timeframe?
- Are employees hearing the messages sent by project management?
- Are employees buying into the decisions that the project is supporting?

Capabilities

- Will sufficiently skilled and capable personnel be available?
- Are management and all resource-providing organizations committed to making resources available?
- Will the organization develop skills that can be maintained once each project is complete?
- Are the end user groups willing and able to adapt to change?

Knowledge

- Is information about the external environment (e.g. current policy, pending regulatory changes) available, accurate, and timely?

Reliability

- Are the expected techniques and technologies to be used “new and untested” or “tried and true?”
- Is there a track record of similar projects?

Constraints

- Is there latitude for deviation from the expected costs and timeline?
- How stringent are the project scope requirements?

Management

- Is management willing to grant project management a level of authority commensurate with its level of responsibility?

- Does project management have the authority to enforce the assumptions it wishes to make?
- Does the culture facilitate building effective teams?
- Does the culture encourage collaboration and cooperation?

Exhibit B – Common Project Risks

Enterprise

- Mission and Goals - Project does not support or relate to any corporate missions or goals. Project will indirectly impact an organizational goal or mission.
- Work Methods - Project will directly alter the work methods of one or more departments. Project will alter parts or have a slight effect on work methods.
- Clients - Project provides main support of delivery of services to primary clients. Project will alter some service delivery to clients.

Oversight

- Project plan - Project plan is outdated and/or plan is not followed by most of project team. Plan is approved, complete and used by most of project team.
- Customer Service Quality - No improvements in customer service are made with the implementation of the project. Minor improvements to customer service are made.
- Monitoring Process - No process is established or process is ignored. Process is established, but not well followed or is ineffective.
- Project Size and Scope - Rapidly changing size or scope, requirements are not defined and not signed off by users. Requirements are defined and users signed off but changes to the baseline are expected.
- Quality Assurance - No quality assurance process or procedures are established. Procedures are established but not well followed or effective.
- Management Requirements - None are in place or defined, or they are ineffective. Requirements are defined, some inconsistencies remain, and requirements may not have been distributed to all employees.

Budgetary and Cost Factors

- Funding Sources and Constraints - Budget allocation is in doubt or subject to change without notice. There are some questionable allocations or doubts about availability.
- Cost/Schedule Review - No review process is established or it is totally ineffective. Controls are established but not all complete or in place.
- Cost Controls - Cost control system is lacking or nonexistent. Cost control system is in place but weak in some areas.

- Budget Size - Insufficient budget is available to complete project as defined. Questions remain concerning budget.
- Incremental Payments Based on Deliverables – There are no set, agreed-upon incremental payments based on product deliverables, i.e., per module, per feature, per product. There is some set of agreed-upon incremental payments based on product deliverables.

User Factors

- User Training Requirements - Requirements have not been defined or have not been addressed. User training needs have been considered but training or a training plan is in development.
- User Acceptance - Users have not accepted any of the concepts or design details of the system. Users have accepted most of the concepts and details of the system and a process is in place for user approvals.
- Involvement of Users - Minimal or no user involvement on the development team or little user input into process. Users on the project team play minor roles or have only moderate impact on the system.
- Achievable Benefits - Benefits are not defined, no baseline is established, or benefits are unattainable. Some questions remain about benefits or baseline changing and measurements are doubtful.
- Deliverable Requirements Defined - No requirements are defined for deliverables or they are unreasonable requirements. Some deliverable requirements remain to be defined or the existing ones are vague and immeasurable.
- User Requirements Defined - User requirements have not been defined or are insufficient for a successful project. User requirements have been defined but changes are anticipated.

Project Factors

- Staff Productivity - Staff productivity is low, milestones are not met, there are excessive delays in meeting deliverable requirements. Most milestones are met, there are some delays in deliverables, staff output is acceptable.
- Management Approach - Project management approach is weak or ineffective. Project management approach is good but needs development.
- Manager Authority - Project manager is manager in name only. Project manager has support of most of staff with some reservations.

Technology Factors

- Analysis of Alternatives - Analysis of alternatives is not completed, not all alternatives are considered and/or assumptions are faulty. Analysis of alternatives is completed with some assumptions questionable and some alternatives not fully considered.
- Complexity of Requirements - Project is very complex with multiple requirements from many different users; requirements are complex and hard to define. Project is fairly

complex with some requirements more easily defined; several user groups will be aiding in the design.

- System Integration/Interfaces - Extensive integration of systems or exchange of information or interfaces are a major part of project. Some integration or interface is required and/or of some importance to the project.
- Fit with Existing Environment - Introduces new technologies to the environment. Limited use of new technologies.
- Quality Control - Time line is likely to adversely affect quality and completeness. Has critical time line but little or no impact on quality.
- Open Systems - Proprietary system with little or no communication with other technologies possible. System is capable of communicating with other technologies on a limited basis.
- Vendor History - Vendor has a poor history or has little experience dealing with deliverables. Vendor has limited history dealing with deliverables or has some successful projects.
- Vendor Support - Vendor provides little or no support for hardware/software and only at high cost and with poor response times. Vendor provides adequate support for hardware/software at a contracted price with reasonable response times.
- Maturity of Solution - Leading edge (in operation less than one year) or aged technology (over 5 years old). State-of-the-art (in operation from 1-3 years).
- Security - No security measures in place, backup of data and hardware lacking, disaster recovery not considered. Some security measures in place, backups of data and hardware being done, disaster recovery considered but procedures lacking or not followed.
- Multiple Vendors/Major Contractors - No clear delineation between vendor responsibility, contractors are in conflict with one another and there is no clear prime contractor. Prime contractor is delineated, vendor responsibilities are defined, but there is a conflict between vendors/contractor.

Project Management Factors

- Elapsed Time - Project has major schedule delays that threaten the success of the project. Project is within schedule with minor delays on some parts or deliverables.
- Change Control Management - No change control process is being used. Change control process in place but it is not being followed completely or it is ineffective.

Project Team Factors

- Experience of Staff - Staff has little or no experience with projects of this type and lacks experience with hardware or software. Project staff has some experience with projects of this type but lacks experience with hardware or software.
- Availability of and Experience with Productivity Tools - Productivity tools are not being used or considered. Productivity tools are available but not being used to their full

potential or are in the process of being implemented and training is needed.

- Consultant/Personnel Mix - Complete reliance on consultant staff with no corporate staff being trained in the new system. A small percentage of staff or some personnel is being trained on the new system.
- Available Personnel Resources - Project staff has a high turnover rate, little or no experience, or is not available for the project. Project staff is available but not all in place; a training plan is established.
- Expertise with Hardware - New hardware, little experience, different technology. Technology is similar to existing systems, there is some in-house experience.
- Expertise with Software - New software and no experience with software or similar products. There is some experience with software or a similar product.
- Technical Training of Staff - Training is not readily available or no training plan is in place. Training for some disciplines is not available but training is planned and is available.
- Expertise with Methodology - Project staff has little or no experience with the project methodology or a similar methodology. Project staff has some experience with the methodology or a similar methodology.

Achieving System Quality through Strategic Test Management

Abstract

This whitepaper will discuss how to effectively assess and provide objective evidence regarding overall system quality in support of cutover decision-making. It will identify how to ensure that Oracle software is properly installed; the system is configured to meet business needs; applications, databases, servers, interfaces, and networks interoperate properly; the system is reasonably free from defects that could negatively impact business operations; and business transactions are processed accurately and reliably.

Introduction

TQM, Crosby, Juran, and Kaizen are all highly successful and useful quality programs aimed at achieving total quality management. There are probably some principals from each that can apply to ERP, however achieving a “quality” strategic test plan does not mean you have to be a Quality Engineer. When most speak of testing in the ERP space, many think of the traditional “big 3” boiler plate approach: unit testing, integration, and user acceptance testing. Creating an effective strategic test management plan goes much deeper than that, as the generic approach to testing and delivering a quality ERP product is not one size fits all. The primary purpose of creating an established test plan is to ensure system quality by reducing or eliminating project risk at go-live time. Digging into the weeds of the "big 3" should help you establish a quality product. Testing begins when the project does, establishing your master test plan and the details that go into it.

Creating the Goals

All projects have a goal. Establishing a test plan should be treated as a project, with goals defined to ensure the system meets users requirements and will present minimum to no risk at go-live. Establishing what goals your test plan should achieve helps provide the framework for the who, what, when, and where parameters to be tested during your testing phases. A key point to remember is that in order to ensure a quality product, testing throughout the project lifecycle is necessary. Testing at the end of a phase or at user readiness could be too late or not sufficient enough.

When establishing your goals, you are not just testing whether you can process journal entries or enter a PO. The entire system as a whole needs to be accounted for through proper planning.   While establishing the goals, start at the 10,000 foot level and, as you develop your test plan, dig down into the weeds to ensure that you will have covered your goals once execution of the test plan is complete.

The following list represents goals from a recent project:

- Software (3rd party and Oracle applications) are properly installed
- System is configured properly to meet business operations needs
- Applications, databases, clients/servers, interfaces, convertors, and networks interoperate properly
- System is reasonably free from defects which could negatively impact business operations
- Business transactions are processed accurately and reliably
- Prescribed requirements are satisfied (e.g., functional, technical, interface, conversion, infrastructure, security)
- System is effectively integrated within the business environment -- i.e., interoperates properly with legacy systems and integrates effectively with business functions in accordance with business policies and rules.

Reading through the list, you can see the goals are clear and all point to a common theme: having a quality system to operate a business.

Test Scope

Characterization of a project's test item inclusions can be made from several perspectives. As a baseline, the following are a good start.

- Primary applications

- Primary processes and system components
- Requirements classes
- Sub-processes

These characterizations are intended to be illustrative and holistic in nature. Given the dynamics of project requirements and design/configuration solution engineering, your scope could change as the project progresses.

Class	Subclass	Repository
Functional Requirements	• Manufacturing • Financials • Procurement • BI reporting	Requirements Traceability Modules -- RTMs spreadsheets have enumerated functional requirement statements, prioritized and traced to business processes and sub-processes, solution elements, and (eventually) test case mappings in these repositories.
Interface Requirements	• Inventory • Core Financials and Procurement • Parent Company Systems	System Interface Requirements and Strategy describes interface requirements
Technical Requirements	• Usability • Performance • Security • Operational • Development	Technical Requirements describes (non-functional) technical requirements
Conversion Requirements	• General Ledger • Human Resources • Inventory • Purchasing • Accounts Payable • Fixed Assets • Accounts Receivable • Cash Management • Bills of Material	Data Conversion Strategy and Overview describes conversion requirements
Architecture (Infrastructure) Requirements	• Availability • Clients/Servers • Database • Environment • Recovery • Data Retention • Back-Up • Scheduling • Printing • Security • Networks • Capacity • Performance	Technical Architecture Overview overlaps with above document and describes architecture (infrastructure) requirements

Primary Applications

A good test plan should clearly identify what the primary applications are. Although you are implementing the E-Business suite, what modules are included? Not everyone is a member of the project team, so the audience of your test plan and results may include others, such as a parent company or Board of Directors. Both of these groups have a vested interest, but may not be involved with day-to-day activities of the project.

E-Business Suite Applications: Purchasing; Inventory; Accounts Payable; General Ledger; Fixed Assets; Cash Management; Accounts Receivable; Work in Process, Bills of Material; Engineering

Primary Business Processes and System Components

In order to identify the decomposition of the business processes that are included in the project, they should be matched to an application. Examples could include general accounting functions to the General Ledger module and procure-to-pay activities matched to the Purchasing, Inventory, and Accounts payable modules. Keep in mind an application module can be linked to many processes and vice versa; it is not always mutually exclusive.

Requirements Classes

Creating test classes and sub classes can help you manage results and ensure you have adequate coverage for your project. Requirements classes applicable to the system are documented in class-specific repositories as shown in the following table.

Key Processes (and sub-processes)

The following table details sub-processes associated with each business operation "track" (e.g., Procure to Pay, P2P; Financials, FIN; etc.) that could be included in a project test plan. As you identify and detail each process, the relating sub-process enables you to further delineate the process. This will ensure that you have thorough coverage when you develop your detailed test cases.

TRK	PROCESS	SUBPROCESS
P2P	Purchasing	Purchasing
P2P	Purchasing	Receiving
P2P	Purchasing	Requisitions
P2P	Purchasing	Shipping
P2P	Purchasing	Subcontracts
P2P	Purchasing	Suppliers - Perf Mgmnt
P2P	Inventory	Inventory
P2P	Inventory	Item Maintenance
P2P	Accounts Payable	Exception Processing
P2P	Accounts Payable	Perform Acctg, Reconn, & Reprtg
P2P	Accounts Payable	Process Invoices
P2P	Accounts Payable	Process Payments
P2P	Accounts Payable	Record Retention
P2P	Accounts Payable	Reporting
P2P	Accounts Payable	Supplier Maintenance

FIN	General	Ledger Acct Analysis and Reconciliation
FIN	General	Ledger Financial Reporting
FIN	General	Ledger General Accounting Maintenance
FIN	General	Ledger Journal Processing
FIN	General	Ledger Period End Close
FIN	General	Ledger Reporting
FIN	Assets	Create Assets
FIN	Assets	Dispose/Retire Assets
FIN	Assets	Maintain/Track Assets
FIN	Assets	Period End Accounting
FIN	Assets	System Maintenance
FIN	Cash Management	Account Analysis
FIN	Cash Management	Bank Account Maintenance
FIN	Cash Management	Bank Account Maintenance/Conversion
FIN	Cash Management	Bank Reconciliation
FIN	Cash Management	Cash Position
FIN	Cash Management	Develop Cash Forecasts
FIN	Cash Management	Interfaces
FIN	Cash Management	Security

Test Cycles

This section characterizes the common levels of a master test plan.

- Unit testing -- involves the validation of the functionality of a specific code unit (e.g., procedure, function, trigger, etc.).
- Integration testing -- verifies that individual code units designed to function together actually work as a whole.
- System testing -- performed at the highest level and includes all the modules and code units that are part of the applications.

User acceptance testing -- the ultimate testing step performed by representatives of the user community to establish system readiness for deployment.

Unit Testing

The objective of unit testing is to determine that each developed component works individually and correctly as designed. Defects rectified at this level of testing are least costly, so thoroughness of verifications is extremely important. Unit tests will be exercised by project team personnel--typically developers--within the development environment. Ideally, the developer's own tests of his/her unit should be supplemented by one or more peer developers to enhance objectivity.

Unit tests verify lower level application components (module configurations and any applicable report generation, internal and/or external interfaces, conversions, and extensions). Unit testing is conducted during the Build phase. It is crucial that units to be integrated are as free of local defects as possible.

Unit testing measures the quality of these components. Unit tests should be written before the coding or configuration and preferably be automated. Ideally, consideration should be given to unit tests which are automated and even integrated with check-in procedures.

Some of the tasks performed during unit testing are as follows:

- Ensure that business rules are implemented correctly and in their entirety
- Evaluate the functionality of menus and graphical user interface items
- Assess the navigation characteristics of data input forms
- Check data validation at the field and record level
- Assess screen layouts and functionality
- Evaluate error messaging capabilities
- Test reporting functionality over a wide range of parameters
- Test batch procedure error trapping and recovery capabilities
- Ensure that application modules communicate properly
- Ensure that interface systems import or export the right information to and from the application database.

Integration Testing

Integration testing involves the sequencing of putting individual units together into increasingly larger entities until the entire system is built, i.e., units have been fully integrated. Interim testing steps may involve verifying individual applications or partial threads of system functionality through one or two applications. The integration tests are conducted within a controlled test configuration environment.

These tests largely address communication between components. Four main mechanisms for communication exist and must be tested:

- Direct call from one function to the other
- Communication through shared data / databases
- Communication through files (batch)
- Communication through distributed methods (gateways, proxies, FTP, etc.)

Integration testing can be applied at more than one level (e.g., unit integration, application integration, system integration).

System Testing

System testing determines that the software components work together as designed to support the business processes. These tests check externally visible features of the complete system. Every function should be addressed, as well as object lifecycles, performance, security, and other characteristics. ‘End-to-end’ processes within and between the ERP modules and external entities are fully tested in system testing. The tests are conducted within a controlled test configuration environment.

Performance Testing

Performance testing evaluates various scenarios that include users or simulated user transactions representing business use cases. Scenarios are generally of two types: those that have significant online activity and those that are primarily or totally batch processing in nature. An example scenario may simulate peak daytime workload and include a mixture of online transactions and some limited batch activity. Another more batch-oriented scenario might simulate a nightly batch process and contain very limited or no online transactions.

Security Testing

- Security testing verifies security requirements including:
- User authentication
- Responsibility/role-based authorization
- Application data security
- Auditing procedures
- Encryption of data in transit
- File system security
- Secure File Transfer Protocol (FTP)
- User provisioning

Operations and Maintenance Testing

- A variety of requirements govern the operations and maintenance features which are tested including:
- Start-up/shutdown
- Back-ups
- Logging
- Failover/recovery
- System monitoring
- Status and error handling
- Reporting

User Acceptance Testing

Acceptance testing is where business processes are used to validate the system fit to the business environment. This test is initiated by end users, but supported by technical assistance from the development team. Technically speaking, it resembles the system test, but the focus is operational evaluation and validation of the solution. The acceptance tests are conducted within a controlled production configuration environment.

User acceptance testing pulls designated end users into the testing process. The testers will create test scripts based on the requirements traceability matrix and execute those scripts. This testing should mimic real-world business processes and situations. This ensures that the system is satisfying the requirements of those who will be using the system long-term.

User acceptance testing is the litmus test of overall project success. A successful result from this testing step usually provides the green light for deployment activities to begin. A panel of end users and a user committee coordinator should be appointed to participate in the testing process. Ideally, the coordinator of this group should participate in all phases of the project lifecycle and provide feedback about the design and effectiveness of the application.

Regression Testing

Regression testing evaluates system response to changes in components. Most often these changes are related to applying defect corrections, but could also be triggered by other system changes such as updates to commercial components. Regression testing actually encompasses all the types of testing described above. Any tests may be re-executed as necessary to assess the impact of the change. Expect that there will be multiple "test cycles" within most testing phases leading to success in the final cycle.

Test Readiness Review (TRR)

A Test Readiness Review (TRR) will be held in advance of the formal testing during the Test phase of a project. The TRR is a product and process assessment to ensure that the system and all subsystems have stabilized in configuration and are ready to proceed into final testing. The TRR assesses system integration; performance and user acceptance plans; test objectives; test methods and procedures; scope of tests; and determines if required test resources have been properly identified and coordinated to support planned tests. The TRR verifies the traceability of planned tests to project requirements. It also determines the completeness of test procedures and their compliance with test plans and descriptions. The TRR assesses the impact of known discrepancies to determine if testing is appropriate prior to implementation of the corrective fix. The TRR documents will include a final plan for testing, training, and system deployment.

Test Data

Testing consumes and produces large amounts of data, and its integrity plays a huge role in testing success. Data describes the initial conditions for a test, forms the input, and is the medium through which the tester influences the software. It is manipulated, extrapolated, summarized, and referenced by the functionality under test, which finally spews forth yet more data to be checked against expectations. It is a crucial part of most functional testing. A system is programmed by its data. Functional testing can suffer if data is poor, and good data can help improve functional testing. When properly structured, test data can improve understanding and testability, reduce maintenance effort, and allow flexibility. Careful data preparation can even help to focus the business where requirements are vague.

Requirements

This section establishes test program requirements including:

- Testing policy (e.g., entry/exit criterion, iteration, deviations/approvals, controls/metrics)
- Test documentation plans (e.g., purpose, format, content, review/validation and approval)

Policies

General test policies can include the following:

All documented requirements shall be traced to a test case (i.e., must be verified within at least one test activity)

A stand-alone test plan shall be developed for each major test level (i.e., unit testing, integration testing, system testing, and system acceptance testing)

Test procedure documentation shall be developed for all test activities

Verification Methods

Most requirements should be verified by test or supported by quantitative test data. The testing can take place at many levels, including unit-, subsystem-, and system-level tests.

Verification by demonstration is applied to some qualitative requirements that cannot be tested. For example, a requirement such as a failure of X shall not preclude operation of Y could be verified by demonstration. The example might, in addition, require some analysis. Demonstration is also used for requirements that can be tested but that cannot be tested over a full range of relevant scenarios. Demonstration is almost always used for requirements containing phrases such as shall support or shall not preclude, since it is impractical or impossible to prove that these requirements are met under all reasonable circumstances.

Because of cost or physical limitations, some requirements cannot be verified by test alone. In such cases, analysis is performed to ensure that the system will meet its requirements. A common example may be expandability requirements. It may be impractical to verify by testing or demonstration that a system can be expanded with X maximum additional server clusters, so analysis of vendor technical data may suffice. “Analysis of lower-level test results” may also support certain verifications at higher levels. The analysis performed to verify the requirement shall be presented in a formally released document referenced in the compliance matrix. To the extent practicable, analysis should be validated by correlation to test data.

Defect Priorities

Anomalies experienced during test execution are referred to as “defects.” Defects shall be logged in a tracking system along with salient information referencing the circumstances of discovery (e.g., tester, date/time, test case, procedure step, free text description). In addition, a standardized prioritization of defects shall be made. Take, for instance, the following example:

Level 1 Defect:
A failure that causes a complete loss of service or would cause severe operational difficulties and/or financial loss. Work in a functional area, the operation of which is critical to the business, cannot reasonably continue. A Level 1 defect may have one or more of the following characteristics:

- Data is corrupted and impacts a business function;
- a critical function is not available;
- the system hangs indefinitely or causes unacceptable or indefinite delays for resources or response times; and/or
- the system crashes.

Level 2 Defect:
A failure of the system, or part thereof, which would affect an APL business function such that it is:

- Degraded and impacts multiple users; or
- is unusable by end users.

Level 3 Defect:
A failure of the system, or part thereof, that would have a minor impact on a business process and can be resolved on a non-immediate basis. Examples include: user requests (e.g., system enhancement); peripheral problems (e.g., network printer).

Metrics

Metrics appropriate to the level of testing shall be implemented to facilitate effective management, communications, and process improvement. The following metrics are to be used at a minimum through each major test activity:

1. Defect density – measures the rate of defect discovery during test cycles.
2. Defect closure – may be combined with above; measures the rate of resolving defects over time.
3. Test case execution – measures the planned vs. actual rate of test case completions over time within a given test phase.
4. Test case outcome – measures the passed versus failed test cases over time.
5. Defect source – tracking and trending measure for assessing where a defect was introduced and caused (e.g., requirement error, design error, development error, etc.)

Documentation

Documentation is inherent to all projects, and creating a master test plan can add to that mountain of paperwork. However, as cumbersome as it may be, thorough documentation of your master test plan enables you to get one step closer to a successful implementation. This section will discuss the various types of documentation.

Test Plans

Test plans prescribe the scope, approach, resources, and schedule of the testing activities. They identify the items being tested (and not tested, if any), the features to be tested, the testing tasks to be performed, the personnel responsible for each task, and the risks associated with the plan. They include:

- Test plan identifier
- Introduction
- Test items
- Features to be tested
- Features not to be tested
- Approach
- Item pass/fail criteria
- Suspension criteria and resumption requirements
- Test deliverables
- Testing tasks
- Environmental needs
- Responsibilities
- Staffing and training needs
- Schedule
- Risks and contingencies

Test Procedures

Test procedures incorporate the details of each level of testing (unit, integration, system, and user acceptance). The specific organization of test procedure documentation shall be defined in approved level test plans.

Test Logs

Test logs shall be used to record day-to-day testing activities and results. Specific content and management of these official test records shall be as specified in approved test procedure documents.

Test Reports

Test reports summarize the results of the designated testing activities and provide evaluations based on these results. The specific organization of test report documentation shall be defined in approved level test plans.