FEATURE

# Plan for Successful System Implementations

**Do you have something to say about this article?**

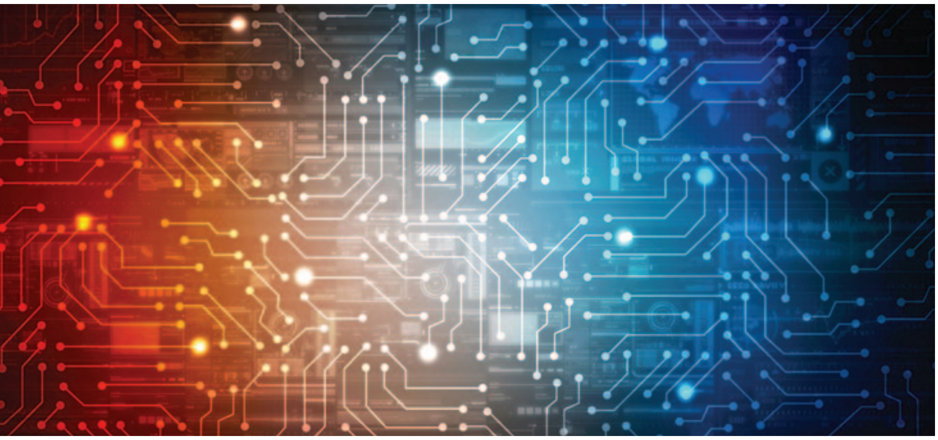
Visit the *Journal* pages of the ISACA® website *(www.isaca.org/journal)*, find the article and click on the Comments link to share your thoughts.

***https://bit.ly/2RSUWQ2***

Organizations need to constantly assess their systems landscape and identify projects that meet business needs and add value to the enterprise. Information technology departments play a crucial role when implementing systems, whether the systems are developed and implemented in-house or outsourced. The key decisions made during system implementation have an impact on IT events that occur long after a project is completed. These events include but are not limited to the following:

- The inability to implement certain types of technology in the future due to incompatibility with the system being implemented now
- Financial impacts due to the project team's failure to consider the future costs of maintaining the system
- Security and legal implications as a result of the project team's failure to evaluate and implement IT best practices and general controls at the time of implementation
- Business continuity implications because disaster recovery plans and procedures were not adequately considered at the time of implementation

All projects are unique, but there are basic principles that project teams can apply to ensure success. The discussion herein outlines factors to consider when implementing systems based on real-world project experience.



**Hashim Seedat,** CISA

Is currently the head of information technology at National Bioproducts Institute NPC (NBI), based in Durban, South Africa. Prior to joining NBI, he was a senior member of the PricewaterhouseCoopers (PwC) Durban office, leading IT audits and providing assurance to clients on major project implementations. Seedat had been seconded and served various PwC offices globally including the Boston (Massachusetts, USA) and Atlanta (Georgia, USA) PwC offices, and he has also performed internal audit quality reviews on behalf of PwC member firms.

## Project Governance

On a recent Software-Defined Wide Area Network (SD-WAN) implementation project at a pharmaceutical manufacturing enterprise in South Africa, project team members found that their expectations and deliverables were different from those of their systems implementation partner. The project leader constantly had to refer the systems implementation partner to various agreements and documentation that had been approved by both parties at the beginning of the project. Had some of this basic project documentation not been approved, the effort and cost to the enterprise would have been significantly higher than expected. Based on this project, the project team learned the following key lessons:

- The objective of the project needs to be clearly defined, and the project sponsor needs to be involved in defining the key deliverables before the project starts.
- The project team may be unable to fully document the technical specifications for the new system at the beginning of the project. In these cases, it may be useful to consider going to market with a request for information and consulting with experts prior to releasing a quote or a proposal request.
- Even though the project may involve a new system implementation or an upgrade, the project team should focus on understanding the current system's setup and processes (as is).
- Enterprises generally budget for projects a year in advance because they need to go through a budget approval process. If a length of time has

passed since the budget approval and the project start date, it is beneficial to reassess the project's feasibility and expected return on investment (ROI), as other recent projects and changes in the business may have affected project outcomes.

- A project charter should be drawn up, and all responsible parties should sign off on it at the beginning of the project. This document should include details of or references to other relevant documents. At a minimum, it should include the following:
  - Objectives and deliverables of the project
  - In-scope items, exclusions and assumptions
  - High-level timelines and responsible parties. Detailed timelines should be documented in the project plan.
- The system blueprint is the single most important document in the project. It needs to be reviewed in detail, as the blueprint document is the foundation for successful implementation.

> “THE SYSTEM BLUEPRINT IS THE SINGLE MOST IMPORTANT DOCUMENT IN THE PROJECT.”

- At project inception, the following logs should be saved in the project management repository and maintained throughout the project:
  - Decision log. The minutes of steering committee meetings can be used to ascertain key project decisions or to compile the decision log. It is a matter of personal preference, but often it is simpler to use one central document or log rather than tracing key decisions back to various steering committee meetings.
  - Change log
  - Risk and issues log
  - Lessons learned log. This type of log is almost never maintained, which can result in enterprises repeating costly mistakes. Some enterprises document project lessons as part of the post-implementation review. However, lessons can be learned at every stage of the project from inception to completion. Hence, it is useful to start a lessons learned log at project inception and add to it throughout the project.
- Regular project meetings (e.g., weekly status update meetings) should be scheduled, and stage gate requirements should be defined. Stage gates are used to describe a point in a project or plan at which development can be examined and any important changes or decisions relating to costs, resources, profits, etc., can be made.[1]

## Hardware

When an enterprise decides to implement a system, it generally defines the hardware specifications at a very broad level in the request for proposal (RFP) or user requirements specifications (URS) document, which is part of the procurement tender process. If the hardware specifications are not clearly defined, the bid evaluation team may find itself in an uncomfortable situation when comparing proposals from various vendors. For example, Vendor A may propose superior hardware, but Vendor B obtains the same number of evaluation points because it meets the requirements stated in the RFP or URS document. Therefore, project teams must critically evaluate their hardware requirements. It is a good idea to consider the following, at a minimum:

- Enterprises should define hardware standards in terms of minimum specifications and preferred brands.
- In some environments, such as the pharmaceutical manufacturing industry, using virtual environments and virtual machines can make IT systems management simpler and more efficient. If the systems are suited to virtual environments, it is much easier for the IT team to add new machines to the existing backup and replication process. Other advantages include the security benefits that come with virtual environments.

- If the project involves the replacement of hardware that contains data (e.g., hard drives), the hardware needs to be disposed of in a secure manner in accordance with the enterprise's policies and procedures for the disposal of sensitive data.

## Operating System

In one case, a project team implemented a system without taking into account operating system requirements. After some time had passed, IT general controls were evaluated, and the reviewers noted various security risk factors at the operating system level. The project team was then tasked with remediating the problems, and it found that to do so, the live production system had to be taken offline. As a result, the remediation efforts took double the time of the initial setup. Based on the lessons learned from this project, project teams should consider the following:

- Make sure that the latest stable version of the operating system that is compatible and supported is implemented. Patch the system to the latest patch levels before adding applications and configuring services on the system.
- Disable default accounts and do not use generic accounts on the system. Rename any default accounts that are required. It is much easier to configure accounts before services and applications are installed.
- Consider who will require administrator/privileged accounts and the levels of access required. Local accounts should have the same password requirements (i.e., length, complexity, history, lockout) as those set out in the corporate security policy.
- Evaluate audit trail requirements. The default event log on Microsoft Windows has known limitations (e.g., ease of use and retention period). Consider using third-party tools to meet needs.

## Industrial Control Systems

An industrial control system (ICS) is defined by the US National Institute of Standards and Technology (NIST) as:

> *An information system used to control industrial processes such as manufacturing, product handling, production, and distribution. Industrial control systems include supervisory control and data acquisition systems used to control geographically dispersed assets, as well as distributed control systems and smaller control systems using programmable logic controllers to control localized processes.*[2]

> **“PRACTITIONERS CAN BE FIXATED ON THE FUNCTIONALITY WITHIN THE APPLICATION AND IGNORE…OTHER KEY ELEMENTS RELATED TO THE APPLICATION ITSELF.”**

Therefore, inadequate IT and security controls on the ICS will result in unsafe and unreliable operations. At a minimum, enterprises should consider the following controls related to the ICS:

- Most important, segregate or segment networks to prevent the spread of malicious activity.
- Restrict physical and logical access to the ICS to the extent possible.
- Limit interconnections between the ICS and other networks through demilitarized zones (DMZs), firewalls, proxy servers or specialized network filtering devices.
- Implement measures and controls to reduce the risk of vulnerabilities (e.g., through patch management and technical configuration).
- Add ICSs to the existing incident response and disaster recovery plans.

## Software

The benefits, added functionality and value expected from new software and application packages drive IT teams to embark on system implementation projects. However, practitioners can be fixated on the functionality within the application and ignore the integration, security, configuration model and other key elements related to the application itself. Some of these elements project teams should consider include:

- Clarifying the software licensing agreement (i.e., perpetual vs. subscription) and recording license keys in a central place
- Adding any new software to the internal list of authorized software
- Performing penetration and vulnerability testing on applications
- Configuring most software systems with a server and client interface that communicates. It is not advisable to allow users to log directly on to servers to access application interfaces.
- Aligning the password parameters on applications to mirror those set out in the corporate security policy. Default accounts should be disabled, and generic accounts on the system should not be used. Any default accounts that are required should be renamed.
- Evaluating audit trail requirements and ensuring that the audit trail is complete and accurate and cannot be edited or deleted in any way

## Backups

Imagine a project team successfully implementing a new system within the allotted time, budget and quality parameters, only to lose all those benefits due to data corruption, cybercrime, negligence or malicious behavior. Designing system backups during the project may cost time, but it will prove to be a wise investment if there is ever a need to recover from a disaster. At a minimum, project teams should consider the following:

- Updating the existing backup schedules to include the new system being implemented and configuring alerts for backup events (i.e., successful and failed backups)

> “DESIGNING SYSTEM BACKUPS DURING THE PROJECT MAY COST TIME, BUT IT WILL PROVE TO BE A WISE INVESTMENT IF THERE IS EVER A NEED TO RECOVER FROM A DISASTER.”

- Considering the impact of malicious code being replicated and its effect on the team’s ability to restore data (i.e., offline backups to prevent ransomware from damaging the enterprise too severely) if the organization is replicating between data centers
- Not overlooking the implications of sensitive data being accessed by unauthorized personnel through the restoration of unencrypted backups. Most organizations consider security controls when implementing systems, but some stop there.
- Considering the recovery point objective (RPO) and recovery time objective (RTO) for the specific system being implemented. The RPO is determined based on the acceptable data loss in case of a disruption of operations. It indicates the earliest point in time that is acceptable to recover the data. The RPO effectively quantifies the permissible amount of data loss in case of interruption.[3] The RTO is defined as “the amount of time allowed for the recovery of a business function or resource after a disaster occurs.”[4]

## Periodic Reviews

Consider the example of a very technologically savvy parent who purchased a top-of-the-line home Internet router and used it to implement parental controls on their children’s tablet devices. A few weeks later, the parent noticed one of his children using a tablet outside of acceptable hours, as defined on the router. After reviewing the router’s activity logs, the parent correlated the times when the tablet was connected to the router with the times he saw the child using the tablet. The parent was shocked (but impressed) to discover that a child could easily bypass the parental controls by connecting directly to the WiFi extender instead of the router. Reviewing logs on a periodic basis ensures that the system is being operated by authorized individuals as intended. At a minimum, consider the following review controls for systems that are being implemented:

- At the operating system level, the following reviews should be undertaken periodically:
  - Successful and failed log-in attempts
  - Log-in patterns (e.g., log-in attempts outside of business hours)
  - Time elapsed since passwords were last changed, especially for administrator accounts with a valid business reason for having no maximum password age
  - File and/or folder changes on the operating system
- At the application level, business owners should be defined, and they should consider reviewing the audit trail periodically.

## System Testing

During the testing phase, the blueprint document should be compared with the as-built document and live production system. Business users need to be included in the testing phase, and testing should be done on a system that simulates the live environment. Other testing considerations include:

- Conducting end-to-end, interface, compatibility, stress/volume and performance testing
- Testing the application on different operating systems, smartphones, tablets, desktops, thin clients and so forth
- Testing Lightweight Directory Access Protocol (LDAP) integration and the impact on the network
- Performing full integration testing to ensure that there are no adverse effects on individual modules
- Performing system failure testing to evaluate and identify error handling and system recovery procedures
- Masking sensitive data, and protecting or restricting access to sensitive data during the testing phase

## Post Implementation

During the project, access rights are generally relaxed. Securely removing or destroying copies of business information once testing is complete, and revoking access to third parties and the project team is essential. Tracking all post-implementation issues and updating the lessons learned log accordingly are also important post-implementation actions. Finally, the disaster recovery plan should be updated at the end of the project to include details for the new system.

## Conclusion

A number of project management methodologies, tools and checklists can be used when implementing IT systems. The recommendations herein are not meant to replace any of those methodologies. They are intended to provide project teams with supplementary items to consider and are based on experience with major system implementation projects. The lessons shared in this article are summarized in **figure 1** to remind project teams of key considerations at each phase of the system implementation project.

## Endnotes

1 Cambridge Dictionary, “stage gate,” *https://dictionary.cambridge.org/dictionary/english/stage-gate*

**Figure 1—System Implementation Considerations**

| Project Management |
|---|
| • Set clear objectives.<br>• Consider a request for information (RFI) before a quote or proposal request.<br>• Document as-is processes.<br>• Define ROI.<br>• Sign off on the project charter, including objectives, deliverables, scope, exclusions, assumptions, timelines, responsibilities.<br>• Create and maintain logs (decision log, change log, risk and issues log, lessons learned log).<br>• Schedule regular meetings. |

**Figure 1—System Implementation Considerations *(cont.)***

| |
|---|
| **Hardware Considerations** |
| • Define standards and preferred brands.<br>• Consider virtual environments if supported.<br>• Dispose of data in a secure manner. |
| **Operating System Considerations** |
| • Patch to latest stable version.<br>• Disable default and generic accounts; rename required default accounts.<br>• Consider necessity of privileged access and set passwords per corporate policy.<br>• Evaluate audit trail requirements. |
| **ICS** |
| • Segregate network and restrict physical and logical access.<br>• Limit interconnections through DMZs, firewalls, proxy servers or filtering devices.<br>• Implement patch management.<br>• Add ICS to incident response and disaster recovery plans. |
| **Software** |
| • Clarify licensing agreement (i.e., perpetual vs. subscription) and record license key in central repository.<br>• Add software to white list.<br>• Consider penetration and vulnerability testing.<br>• Design software architecture (e.g., server-client interface model).<br>• Consider necessity of privileged access and set passwords per corporate policy.<br>• Evaluate audit trail requirements. |
| **Backups** |
| • Add new system to backup schedule and configure backup alerts.<br>• Consider offline backups.<br>• Encrypt and secure backups.<br>• Define RPO and RTO. |
| **Periodic Reviews** |
| • At operating system level, monitor successful and failed log-in attempts, log-in patterns, password age, file and folder changes.<br>• At application level, identify business owners who will review the audit trail. |
| **System Testing** |
| • Perform end-to-end, interface, compatibility, stress/volume and performance testing.<br>• Test on different operating systems, smartphones, tablets, desktops, thin clients, etc.<br>• Test Lightweight Directory Access Protocol (LDAP) integration and perform full integration testing.<br>• Perform system failure testing to identify error handling and system recovery procedures.<br>• Mask sensitive data and protect/restrict access to sensitive data during testing. |
| **Post Implementation** |
| • Clean up (i.e., remove access required during the project, destroy business data not required).<br>• Update records (i.e., track post-implementation issues, update the lessons learned log and disaster recovery plans). |

[2] National Institute of Standards and Technology, “Assessing Security and Privacy Controls in Federal Information Systems and Organizations,” Special Publication (SP) 800-53A, USA, December 2014, *https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53Ar4.pdf*

[3] ISACA®, Glossary, “Recovery point objective,” *https://www.isaca.org/Pages/Glossary.aspx?tid=1751&char=R*

[4] ISACA, Glossary, “Recovery time objective, ”*https://www.isaca.org/Pages/Glossary.aspx?tid=1754&char=R*