

# THE 10 TRUTHS OF SUCCESSFULLY MANAGING INFORMATION TECHNOLOGY

**Harold E. Kerr MBA PMP CSM**  
*Input provided by Robert Burhman PgMP*  
*Illustrations by Jay Mazar*

*“All truth, in the long run, is only common sense clarified.”*

Thomas Huxley

## TRUTH 1: INFORMATION TECHNOLOGY: NOT ROCKET SCIENCE, BUT PRETTY DARN CLOSE

---

Since the introduction of the IBM 650 computer in 1953, organizations have been using information technology (IT) to power their operations. Decades of use doesn't mean organizations have IT figured out, though. I've found in my 20 years of management and IT consulting that organizations are still perplexed with how to effectively manage IT needs, processes, projects and resources. However difficult it may be, solving this challenge is mandatory because organizations *must* effectively harness technology to successfully accomplish their missions and remain viable in the marketplace.

Chief Information Officers (CIOs), their IT departments and the business units they support struggle with the beast that is IT due to characteristics inherent to managing it:



- **Complexity:** Simply put, managing IT is tough as hell. IT has proven to be an ungovernable operational quandary for organizations for a number of reasons: IT has difficult-to-tame, kudzu-like tentacles that spread throughout all areas of the organization. The IT landscape is constantly changing. And no offense to the non-IT functions within an organization (e.g., human resources, finance, sales and marketing), but there hasn't been comparable constant and dramatic change in these areas. In large part, the underlying principles related to these functions have remained consistent throughout the years.
- **Dependence:** Caveating the statement I just made above, there *has* been one major area of change within the non-IT functions of an organization—*the increased use of IT within those functions!* In fact, almost every business unit in the organization relies on the IT department to develop systems to help them achieve their goals.
- **Expectations:** Business unit stakeholders (you know the ones—who just read a whitepaper on Big Data and now know enough about the topic to be dangerous) often consider new technology systems to be the magic bullet for what ails their operating units. In reality, technology may only partially address their ineffective operations. And because a few of them watched a “Transform Your Organization with Agile” webinar, they now expect the IT department to deliver a working prototype of the new system after a few three-week sprints.
- **Advances:** Keeping up with the latest technology is a daunting task for overworked IT departments. Determining the correct new technology to implement, and *when* to implement it, can be a crapshoot. This is especially true if the system that was just implemented has now already been deemed outdated.

- **Balance:** Maintaining (and continuing to fund) a portfolio of legacy systems while simultaneously developing their replacements requires the agility of walking a circus tightrope. One false move and the IT department spectacularly falls to the ground, taking affected business units with it.
- **Security:** Addressing all the items above while trying to stay a step ahead of nefarious threats to IT infrastructure and data is made infinitely harder due to security-adverse and trusting users. (“Click this link...” Sure, why not?)

**Face the Truth:** Faced with these challenges, how do CIOs and their IT departments consistently meet the demands of their internal and external customers? By facing the truth. Or, more specifically, they must face and address the “10 Truths of Successfully Managing IT”...continuing with Truth 2, which discusses the foundation of success—joint planning between IT and the business units...

## TRUTH 2: MAKE A NEW PLAN, STAN

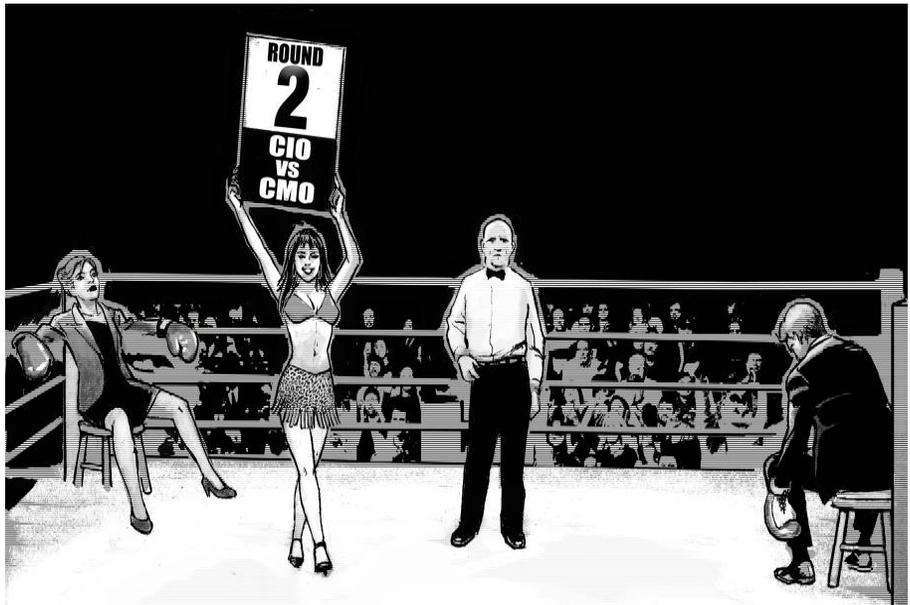
**B**usiness units and their IT department neighbors are modern day Hatfields and McCoys. They simply don't get along. (No need to be coy, Roy.) Case in point, I've been in meetings with IT and business unit representatives that should have been facilitated by Mills Lane, complete with women holding round cards!

How does this organizational discord typically represent itself? Below are some signs that suggest that the business units and the IT department might be at the point of duking it out:

- **The business units are from Mars and the IT department is from**

**Venus:** In personal relationships, reasons cited for conflict often include lack of communication/not listening, discounting the other's opinion, not following agreed-upon decisions, etc. These same issues are common in some IT and business unit relationships, with disdain and mistrust being felt by both sides. Just like in personal relationships where others are brought into the fray, business units have a way of banding together to jointly gang up on the IT department. The result? A relationship that's more sour than an apple Jolly Rancher.

- **Never the twain shall meet:** In many organizations, business units operate from a plan they developed aligned to the organization's strategic objectives, and IT operates from a separate plan related to technology and infrastructure objectives deemed important by the CIO at the time. (Or, one or both entities are not operating from a discernable plan whatsoever). In the absence of common plans, business units look at projects being worked on by IT and scratch their heads wondering why IT isn't working on the projects *they* need. Or, IT operates in what appears to be constant reaction mode, responding haphazardly to the needs du jour—and the sometimes (or oft-times) disjointed whims of the business units.
- **You're not the boss of me:** Business units want control of their own destinies and resent the fact that technology projects that impact their operations—and their performance bonuses—are at the mercy of the IT department. To reassert their authority, recalcitrant business units act out by taking matters into their own hands by conducting DIY system projects that create pockets of "shadow IT" throughout the organization. IT complains this is a waste of resources. When these rogue systems fail to deliver, business units blame IT by stating that if the IT department had done its job in the first place, the business units wouldn't have had to resort to building their own systems.



- **The IT black hole:** While most organizations' IT needs may not be rocket science, millions of working professionals have experienced the phenomenon known as the "IT black hole"—that diabolical black cloud of dust, debris and tangled CAT-5 cable where IT trouble tickets, system change requests, project prioritization lists, PC techs dispatched to fix your monitor, and budget dollars get sucked into a spiraling vortex never to be seen again. Because business units don't have adequate visibility into the IT department and its processes, they are baffled when faced with interacting with their IT department and getting answers. For example, "What's the status of our NextGen system, from like *two* generations ago?"



**Face the Truth:** To be successful, the IT Department and the business units need to engage in joint, structured planning to address how IT will be strategically and operationally performed to meet the needs of the business. The two parties should consider developing the following types of plans:

- **Strategic and Service Management Plans:** The CIO and the Chief Technology Officer (CTO) must be included in the development of the organization's overall strategic plan. Once developed, the IT unit creates a complementary IT strategic plan describing the technology solutions that will be marshalled to achieve the overall strategic objectives. Creation of an IT service management (ITSM) plan (described more in Truth 3)—which describes the actual methods of designing, delivering and managing IT to achieve the objectives—is also advisable. As I note below when discussing the communication plan, aspects of the IT strategic plan need to be culled out, abbreviated and prominently communicated in a digestible fashion to the rest of the organization.
- **Enterprise Architecture Plan:** Tightly coupled with the strategic plan, IT should establish a framework that governs the connection between strategy, business processes and the associated information flows, technology, applications and data. This information is compiled into an enterprise architecture plan (EAP) that can include a technology roadmap identifying legacy applications and the planned applications they'll be migrated to, and the corresponding timelines.

Listen to me closely when I say this: The EAP should not be overly complex. That's the sure death knell of any EA effort. If the EAP is too elaborate, even the rest of the IT department will distance themselves from the Chief Architect. (The Chief Architect doesn't score any points for having a fundamentally correct EAP that no one understands nor adheres to.) The best advice I can give here is that the EAP should be actionable based on organizational realities and IT management maturity.

- **Customer Management and Communication Plan:** No matter how technologically advanced and strategic an IT unit becomes, yes Mr./Ms. CIO, your organization is still primarily a support function. In a troubled relationship, one side has to be the bigger person and make the first move to improve the relationship. As a support function, it's up to the IT department to swallow its pride, admit it's wrong (yes, I know the business units are just as messed up) and take this step by documenting how it will serve customers in a customer management and communication plan.

The plan should articulate how IT will engage with, listen to and communicate back to the business units. The plan also acknowledges each business unit's unique needs, and how those needs will be met. In the IT department, we know business units aren't always right, and sometimes don't have a clue what they want; but that doesn't negate our responsibility to listen, clarify, negotiate and sometimes give in.

Further customer management and communication components that may need to be developed or extracted from other plans I mentioned above include:

- *How IT obtains business units' needs and requirements.* For example, some organizations create a hybrid IT and business unit function or role that acts as the front door into the IT department. This IT "account manager" function obtains business unit needs and requirements, communicates/translates the requirements to the rest of IT, provides status updates and, most

importantly, advocates for the business unit customer—even when pushback is received from their IT unit colleagues (including the CIO).

- *How IT communicates its processes, operations and status of customer requests.* The plan needs to answer internal customer questions such as: Who do I call for X, Y or Z? Where do I send change requests? How do I get information on whether the request has been accepted and scheduled? What's the release cadence for certain systems? What happened to all that money we gave IT last year? (Well, maybe not this last question...)
- *How IT provides value to the organization.* As a function that may not directly generate revenue but instead absorbs, in their minds, a lot of the business units' money, the IT department usually has an image problem. Thus, the plan can describe how the IT department will communicate return on investment, performance measure outcomes and notable achievements.

Developing and enacting the described plans should help mend strained relationships by fostering business unit and IT interconnectedness, and increasing transparency and accountability. To increase the chances of successfully enacting the plans requires addressing IT principles and approaches described in the next truth...

### TRUTH 3: EAT YOUR CHICKEN ALPHABET SOUP—IT'S GOOD FOR YOU

---

Developing and implementing the plans described in Truth 2 doesn't mean your IT department will immediately have a clean bill of health. Without sound, proven principles and methodologies built into the plans, the IT department could still be very sick. Some symptoms of this illness include:

- **Disorganization of services:** IT services are not delivered in a consistent manner. There is no standard method for business units to interact with the IT department.
- **Project and technology indecision:** The IT department is having difficulty deciding which projects to approve and which technologies to adopt, and how the technologies will fit into the overall IT portfolio and infrastructure.
- **Low (or no) ROI:** The organization is funding systems without using a criteria-based decision-making process. Not only is there low return on investment, there's not adequate remaining funding to meet critical needs.
- **Inconsistent project and system development and approaches:** There are no effective means for prioritizing, approving, staffing and managing IT projects. System development efforts are not following a structured development methodology. Project teams recreate the wheel, or worse yet, fail to use a wheel at all. As a result, system development project performance is uneven and not replicable. It should be no surprise to anyone when the IT projects are over budget and often behind schedule. The most overused cliché of all time (so let's use it) was probably not first said by Albert Einstein, but instead by someone frustrated by their organization's delivery of IT projects: *"Insanity is doing something over and over again and expecting a different result."*
- **Lack of validation and unbiased insight:** The organization does not have a mechanism for adequately and consistently confirming that systems being developed meet requirements and perform as expected. Obtaining an accurate assessment of the true health of development projects is almost impossible. Meanwhile, no one's watching the system and software providers and vendors closely as they merrily thrive in the organization's dysfunction and exceed their service and product sales quotas on the organization's dime. (Check out Truth 7 for more on this item.)



**Face the Truth:** The IT field is full of acronym-based approaches; implementing these approaches can help a sickly IT department rebound back to health. A brief overview of some of the most effective three- to five-letter alphabet soup approaches an organization can adopt to restore the healthy functioning of its IT department include:

- **ITSM (Information Technology Service Management):** Mentioned in Truth 2 as a companion concept to an organization's IT strategic plan, ITSM is a strategic method for focusing on the results the use of IT services and systems produces for customers. ITSM methodologies include approaches for designing, implementing, measuring and improving all things IT (e.g., processes, people and technology).

The overarching objective of an ITSM approach is to deliver customer-satisfying, soup-to-nuts IT services based on best practices and supported by key performance indicators (KPIs), which are backed by service-level agreements (SLAs) that keep IT *and* their internal clients accountable. To aid in implementing and managing ITSM processes, organizations can choose from a number of software products on the market built just for this.

A listing of common ITSM-like methodologies yields yet another set of acronyms:

- ITIL (Information Technology Infrastructure Library)
- COBIT (Control Objectives for Information and Related Technology)
- ISO/IEC 20000 (International Organization for Standardization/International Electrotechnical Commission)

- MOF (Microsoft Operations Framework)
- CMMI-SVC (Capability Maturity Model Integrated for Services)

There's no correct ITSM to select over another, and an organization does not have to fully implement all aspects of an ITSM to improve its processes. For example, it might make sense for an international electronics company to implement a full-blown ITSM, whereas a 100-person organization could gain benefits by only selecting certain snippets of an ITSM.

- **ITIM (Information Technology Investment Management):** ITIM and another commonly used term, CPIC (Capital Planning and Investment Control), are terms coined primarily by the U.S. government to represent a process that all types of organizations can perform for IT investment decision making. ITIM typically includes an investment review board (IRB) function (comprised of business unit, IT, CFO and COO representation) that reviews and prioritizes proposed IT projects against items such as the organization's strategic plan, the enterprise architecture plan or the technology roadmap. The IRB is also tasked with analyzing each project for risks and returns before approving a significant investment. For example, projects closely aligned to strategic goals will receive a higher prioritization than those that aren't. This decision-making activity is not only related to proposed projects, but also addresses funding for existing investments, as ITIM governs the ongoing performance of projects to ensure they are on target to achieve their defined benefits. Projects not faring well are less likely to receive continued funding than well-performing projects.
- **PMO (Program/Project Management Office):** The PMO is a unit, typically comprised of business unit and IT representation, that performs varying degrees of the functions related to how the organization standardizes, prioritizes, initiates, staffs, manages, performs and monitors programs/projects (that's a mouthful). The PMO can be centralized (one PMO addresses all programs/projects), distributed (each major program/project has a PMO) or a mix of both models. Further, the PMO can act primarily in a standards/reporting/governance role (typically referred to as an administrative PMO), or also be involved in project execution (e.g., a managerial PMO). But look, forget the labels...the reality is most PMOs are a blend of the different types based on the needs of the organization, and the PMO role can even vary by specific program/project or by how much control a certain program sponsor wants.

When integrated with the ITIM process, the PMO can act in concert with the IRB. For example, as input into the ITIM process, the PMO works with the business units to document business cases and to define requirements to present to the IRB. After ITIM decisions are made, the PMO receives approved *and funded* projects to plan, staff and schedule. Project managers can reside within the PMO, or be sourced from elsewhere in the organization. The same holds true for project staff such as business analysts, architects, programmers, etc. However, in many cases, these individuals work in the IT development and delivery units, and are matrixed to the PMO.

Determining if a PMO is needed within an organization depends on organization size and the complexity and number of concurrent IT projects.

- **A Guide to the Project Management Body of Knowledge (PMBOK® Guide):** It's probably a fact that the largest percentage of projects performed within an organization is IT related. Thus, even if a PMO isn't warranted, an organization should still take advantage of implementing formal project management practices and ensure that ownership for maintaining these practices is identified. These standards can be based on program/project management practices defined within *PMBOK Guide*, the worldwide standard of methodologies and practices for project management. The guide provides project-related methodologies and resources for the management of integration, scope, time, cost, quality, human resources, communication risk, procurement and stakeholders. The creator of *PMBOK Guide*, the Project Management Institute (PMI), provides Project Management Professional (PMP) certifications for individuals that meet certain criteria and pass a comprehensive test. Increasingly, organizations are requiring that their project managers possess a PMP certification.
- **SDLC (Systems Development Lifecycle):** A SDLC methodology is a repeatable process for managing IT design and development projects that defines various project phases and identifies standard deliverables required by each phase, as well as phase entry and exit requirements. Using a well-defined SDLC increases the chance of a system development project's success. SDLC methodologies can be generally categorized as "waterfall" or "agile." An organization's PMO would be involved in determining which SDLC is suitable for a given project. And regardless of what contemporary literature suggests, one SDLC type is not inherently better than the other. (To all the agile fanatics out there, I've seen both

waterfall and agile system projects be performed poorly.) The SDLC chosen should best fit the organization and specific project characteristics.

- **EVM (Earned Value Management):** EVM is a project management technique that measures the progress of a project objectively by using the following categories of measurements: (1) Accomplishment of planned work, (2) schedule performance and (3) cost performance. While EVM is a great tool for assessing project performance, it requires a lot of planning and effort to implement accurately and consistently across an organization. I'll spare the readers a lengthy discussion on the additional list of acronyms associated with EVM (like EAC, BAC, VAC, etc.) and instead focus on what an organization needs in place to obtain the most benefit from EVM. For EVM measurements to be accurate, in addition to using formal project management and schedule development processes, an organization needs to employ the following elements on its projects (at a minimum):
  - Have fully developed, and baselined, work breakdown structures (WBS) with cost and time estimates at a work package level
  - Have a means of assigning work to project staff at the work package level
  - Have a logical means of identifying how progress will be measured (not, "Bob, what percent done are you?")
  - Have a means of accurately capturing and integrating all actuals and progress data
  - Have a means of performing baseline revisions

I've found the implementation of EVM spotty because organizations don't have all these necessary components and practices in place. For large (meaning having the wherewithal to fund the enablers for formal project management) disciplined organizations, EVM can be part of normal project management; however, less able organizations should strive to walk before they run. They should first focus on developing better project WBS and schedule artifacts, and then work on doing a better job of monitoring progress. Finally, they can slowly graduate to EVM if warranted based on organizational needs.

- **V&V/IV&V (Verification & Validation/Independent Verification and Validation):** According to the Institute of Electrical and Electronics Engineers (IEEE) standard on the subject (IEEE 1012 - 2016), "Verification and validation (V&V) processes are used to determine whether the development products of a given activity conform to the requirements of that activity and whether the product satisfies its intended use and user needs."<sup>1</sup> Note: V&V is not synonymous with system testing!

While EVM measures if a project is on track in terms of schedule and cost, V&V tasks are performed on system development projects to continually assess and ensure a project is on track from a functionality and quality perspective. V&V tasks include assessing all areas of the system development project including system concept, design, requirements, testing, implementation, etc. These areas can be assessed using the IEEE 1012 – 2016 standard and other related IEEE standards. Even CMMI or *PMBOK Guide* checklists can be used to perform the assessment.

IV&V is a flavor of V&V, where an independent third party (not development project resources) performs the verification and validation tasks, even including non-technical assessments such as assessing the capability of project management staff, project impact, organizational readiness and change management, financial aspects of the project and vendor contract gap analysis.

IV&V is typically used for large projects spread out over a lengthy period of time, or projects being performed by a systems contractor when there are not adequate internal resources to sufficiently assess the vendor's performance. One pitfall of IV&Vs is when, based on the oversight of the IV&V effort, the IV&V is not truly independent. For example, if the IV&V function reports to a CIO or PMO that is responsible for managing IT projects, independence can be affected. To increase the independence of IV&Vs, organizations should consider placing IV&V oversight in an operating unit outside of the IT department; for example, under the COO, the CFO, or its oversight could be performed by the IRB function.

Once the organization has developed the necessary methodologies to use, it can begin trying them out when performing system development projects. But, before it embarks on that next system development project, the organization should read...

---

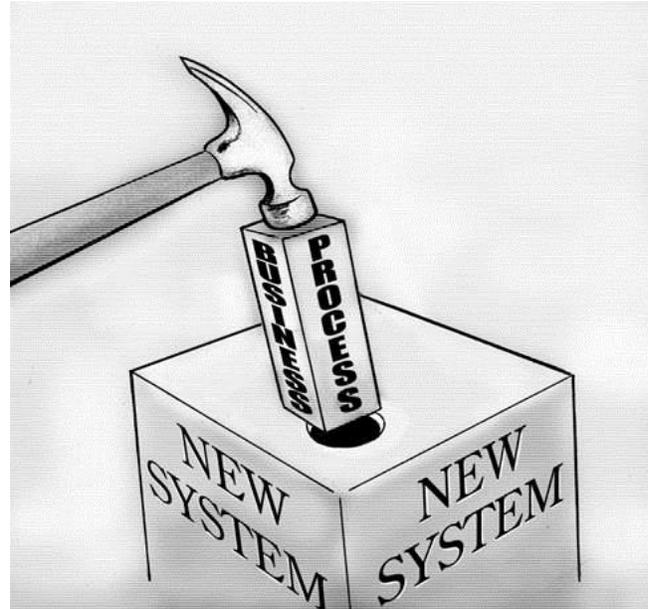
<sup>1</sup> 1012-2016 - IEEE Standard for System and Software Verification and Validation. <https://standards.ieee.org/findstds/standard/1012-2016.html>

## TRUTH 4: PROCESS OR SYSTEM—THE NEW CHICKEN OR EGG

There's a debate raging (okay, maybe not *raging*) within the system development community—which comes first: documenting the organization's business process, or selecting the new system software? For example, when undertaking a new system project, should the organization first define the desired “to-be” business processes and select a system that can satisfy the process requirements; or, should the organization buy the system/software first [e.g., a commercial off-the-shelf system (COTS)] and design the processes to match how the system functions?

The answer is different depending on who's answering the question. For a systems or software vendor, it's a no-brainer: Buy their system first because it captures “best-of-breed” processes, and *presto* the organization will be able to very easily implement those processes. (Excuse me, I just gagged on my sandwich as I wrote the preceding sentence...) If you're a business process junkie like I am, the answer in the majority of cases will be to document the desired processes first, extract the resulting functional requirements and either build or buy a system that satisfies the most requirements at the best price.

I've seen both approaches used. The result? Defining the processes before buying or building a system takes longer up front. However, the technology selection process is improved and has a higher chance of working properly (“Measure twice, cut once” comes to mind.) On the other hand, buying a COTS system first and trying to force existing business processes into it is akin to trying to force a square peg into a round hole and ends up being downright painful for the organization.



For example, I witnessed a COTS built for one type of function be re-purposed, and the client had to try and make its processes fit with it, which resulted in scores of issues. As a result, the multi-million dollar contract was cancelled before the system was ever implemented—*after* the client had spent big bucks for all the user licenses and the development. Also common in this type of scenario, huge chunks of system functionality go unused, and user-developed workarounds proliferate. The system/software vendor is the only beneficiary, as the cash register rings with each change order they get to retrofit the system to match the organization's processes.

**Face the Truth:** Almost always define business processes and extract functional requirements before selecting an IT solution. This approach helps improve the organization's chances of obtaining a system that works. A high-level approach for performing this involves:

- **Define processes:** Systems typically automate a business process (e.g., sales, customer service, booking travel, managing human resource procedures). The target processes should be defined at a level sufficient to capture what needs to be automated. The organization should document details on what steps take place, who performs the steps, what system interaction takes place, what data is input and what data passes through to each step. A cross-functional process map with “swim lanes” does this nicely, and I always add a system swim lane to show the process to system interaction.
- **Document requirements:** If the process maps are thorough, the functional requirements and use cases can be derived directly from the process maps. Here's a little trick—every process step that has system interaction represents a set of functional requirements, and one or more process steps and system interactions make up a use case.

The approach above works for both waterfall and agile approaches. There's a misconception that an agile methodology doesn't require documented business processes and detailed requirements. That's bullpucky. A full set of end-to-end functional requirements (or user stories) form the product backlog, which is carved into sprints (e.g., if using a Scrum approach). Remember in Truth 3 when I stated that I've seen poorly performed

agile projects? The primary reason was a lack of detailed, end-to-end requirements before the team started their sprints.

Having a full set of defined processes and requirements doesn't mean the system team is ready to start development. Using the requirements derived from the business processes as a guide, the systems development team should now perform the systems selection analyses described in...

## TRUTH 5: THERE'S MORE THAN ONE WAY TO SKIN A CAT—JUST DON'T GET CUT!

Consider this scenario: An organization's mission-critical legacy system is nearing the end of its useful life. The IT department is asked for a new technology solution that improves the way the organization handles the function supported by the system. Because CIOs are constantly barraged with information from vendors and trade literature on the latest and greatest technology, programming methods and software platforms, the good news (and the not-so-good news) is that they have a multitude of choices to select for the solution. The tricky part for the CIO is determining the correct choice and the best time to pull the trigger on introducing a new technology within the organization.



Another common scenario is when a system project team has developed a set of detailed requirements (because they abided by Truth 4) and the IT department has to decide the best option for meeting the requirements: Can an existing in-house system be enhanced to meet the need? What about a COTS? Or, are the requirements so unique that the organization needs to build a custom system?

I've found that many IT departments do not have a formal method of performing the analysis to answer this question. Or, the IT department may make the mistake of relying on a system/software vendor to provide an answer. In these cases, magically, the alternative analysis scoring ends up favoring the software suite the vendor is most able to develop (if the vendor only sells a hammer, everything looks like a nail), or corresponds favorably to skillsets of staff the vendor has sitting on the bench unbillable.

Under the presented scenarios, by the time it becomes apparent that the new technology or system option selected was wrong, it may be too costly to choose a different path. Considering the risks related to the sizable cost, technology fit and to its reputation, how does the IT department help improve the technology selection decisions it makes?

**Face the Truth:** To avoid heading down a path to technology perdition, IT departments should utilize a structured, unbiased method to identify new technology or system selection options. Components of such an approach include:

- **Ensure EA compatibility:** When determining if a new technology should be introduced into the organization, an analysis should be performed to ensure that selected technology is compatible with the organization's enterprise architecture plan. (You developed this in Truth 1, remember?) It's important to make sure the technologies play well together and fit logically into the IT infrastructure.
- **Conduct analyses:** When presented with a set of requirements, a "buy versus build" analysis should be performed to determine if the system requirements can best be met with an existing COTS product (buy); or, if the requirements are so unique, it's better to code the system from scratch (build). (And there's also the "buy then build" option, where a COTS is selected, but customization will be performed.) A fit-gap assessment is performed against any suitable COTS products, and if a sizable majority of the requirements—especially the critical requirements—are met, then COTS should be the smart selection. Once this choice is made between buy versus build, the second level of analysis is an alternative analysis, which involves selecting which COTS product or which coding language best fits the bill (e.g., it was determined that there were a number of COTS systems that could meet the need, now the organization needs to determine which one).
- **Decide to adopt or not:** The organization may be presented with new or cutting-edge COTS products or new coding or development technologies. However, most organizations don't need to be an early adopter of new technology. (It's called bleeding edge for a reason, and remember, we're trying to not get cut.) To determine what's best for an organization, an analysis should be conducted on the advantages of the new technology versus other choices; and other factors should be considered, such as skillsets needed, training needs, support needs, cost comparisons (purchase and maintenance), etc.

When there's a strong business case for a new technology, a best practice is using it first on a small, or pilot, basis.

So, your organization has now used a structured method and has decided the appropriate system to select. But note, a system is only as good as its data. Which brings us to...

## TRUTH 6: DATA—THE DEVIL’S IN THE DETAIL

Most systems exist to process and store data. However, the process of actually *managing* an organization’s data is often an afterthought. For example, the organization determines what data the system needs to contain. However, detailed planning may not be performed related to: where the data physically should reside; what other systems/processes will use/share the data; how the data will updated and maintained; and what’s the best method of reporting the data.

Data are one of the most difficult IT elements to effectively manage. Some data truisms, if proper data management practices are not in place, include:

- **The same data in different systems is never the same data:** Take, for instance, a customer’s address—this data should be the same in every system that holds it. Consider this actual example: An organization offers membership and publishes magazines. An individual that was a member and subscribed to a magazine had records in both the member database and the magazine database—with no connection between the two systems once the membership data was pushed to the magazine database. If a customer called the magazine department to change the address, the address did not get updated in the membership system.

Typically, an organization will have a system of record that contains the set of official data. Theoretically, there should be only *one* system of record within a company for a specific data set—not one for finance, one for marketing, and one for customer service, etc. On a large scale, multiple systems of record occur when there is a lack of integration between major systems within business units. On a smaller scale, business units sometimes request data extracts from the official system of record. These extracts are dumped into homegrown systems, and for *that* business unit, their system becomes the system of record for their needs.

- **Data will stay clean about as long as a 5-year-old’s bedroom:** Company data changes constantly. Because of the dynamic nature of data, once an organization cleans up dirty data (if there are not strict data rules in place or a system design that enforces data standards and policy), the data become dirty all over again. Case in point: A company that has to regularly merge and purge its system data of duplicate customer records because users are not checking the system for existing customer records before a “new” customer record is added.
- **Data migration is always going to take longer than you think:** During system development, it can be difficult to derive the initial estimate for the data migration effort because the scope of the system project isn’t yet clear. For example, an organization was replacing its policy administration system with a COTS product. It was impossible to understand what data needed to be migrated to the new system because it was difficult to determine where the legacy system (which consisted of thousands of COBOL files) began and where it ended, and what functionality would be subsumed by the new COTS system and what would remain outside the system. During system development with projects involving large legacy system replacements, an organization should be very wary of any vendor that promises an automated data migration tool that makes migrating legacy data to the new system a breeze. (Like the vendor engaged in the presented example above—even before it knew the scope of the data!)



- **Addressing reporting requirements late in a system development project is a mistake:** System-generated reports can only report on data stored in the system. Sounds elementary; however, schedule-pressured system development teams—in an attempt to simply get core functionality implemented—may decide to address reporting requirements near the end of the project, or after a system's initial implementation. Once the reporting requirements are defined, all too often it becomes apparent that not all necessary data elements were built into the system to generate needed reports, resulting in unanticipated backtracking to add new data fields or new functionality to the system.

**Face the Truth:** Data has to be whipped into shape and managed like the unwieldy troublemaker that it is. To get a firm handle on an organization's data, the IT department should:

- **Develop a master data management (MDM) plan:** A MDM plan identifies an organization's most critical information and identifies a single source of its origin. Designed properly, a MDM plan facilitates data sharing between departments and different platforms and systems. This plan should be derived based on the data component of the EAP, which is developed with businesses unit input (because it's their data!). The plan should also address items such as: organization data needs, data ownership, what areas of the business share the same data, data format standards, data currency, data retention, archival, etc.
- **Centralize data:** The terms "data warehouse" or "big data" have been used to convey massive amounts of centralized data the organization is able to do a bunch of neat things with. Don't be fooled by the smoke and mirrors. All too frequently, it means access from one interface to a collection of data that is actually still stored in various places. Instead, IT departments should strive to achieve a "store once, share many" goal (as defined in the MDM plan). Meaning, a customer's address really does only reside in one place, but can appear in any number of systems. Making a change in any of those systems changes the master data. Moving to centralized data takes time and should be done in steps versus all at once. Each system replacement and development project should have a data component and requirements related to achieving a data centralization goal.
- **Build systems with strict data rules:** Most data get into a system via a user interface; thus, this is the ideal place to ensure data is input exactly to specification. This can be accomplished by having data field and input standards (defined in the MDM plan) that are incorporated into each new system development. Additionally, logic or validation rules should be programmed into the system for ensuring the accuracy or cleanliness of the data being input (e.g., checking for duplicate customer records).
- **Scrub data before it's moved:** If a legacy system being replaced has a lot of dirty data, the data should be cleaned in that system *before* it's moved—or during the extract, transform and load (ETL) process. Because nine times out of 10 (my based-on-real-life statistic, not empirically derived), we know that when the organization states that it will clean the data once it's in the new system, it ain't gonna happen. The data cleaning in the legacy system can begin as soon as the data model of the new system is known, and some obvious existing data impurities can be cleaned even before that.

Sometimes cleaning the data involves making programming changes to the legacy system so dirty data doesn't keep getting introduced. While the organization may not want to pay to fix a system that's going to be replaced soon, keep in mind that "soon" may end up taking a long time. Thus, it may make sense to go ahead and fix the legacy system.

Yes, managing data is hard, but what's harder than managing data? Find out in...



In addition to having worked with scores of system/software vendors, I had a brief stint on the dark side as a technology salesperson, so I say this with confidence: Your technology vendor is more focused on his/her sales commission than the success of your organization or system project. Sure, there are some good apples out there, but to be safe, just assume that a vendor with a sales quota is a wolf in sheep's clothing until proven otherwise—and then still be wary!

Vendors can be very charming and convincing (better yet, beguiling). As a result, here are some of the biggest vendor-related sins IT departments commit:

- **Believing the sales team:** Organizations fall prey to making decisions based on what the vendor's sales team tells them, because the sales team makes solving the problem sound easy ("Sure, our tool can do that!"). In reality, salespeople typically don't take time to dig deep into the nitty-gritty of clients' needs or requirements. They have a knack for telling prospective clients just enough to get the sale. Then the vendor's implementation team may or may not be able to live up to what its ambitious sales folks promised. Not only does the client get stuck in the middle, the client foots the bill for the gap in communication and capabilities.
- **Believing vendors sell solutions:** Consider this actual situation: A large software vendor sold four of its products as a bundled CRM and

knowledgebase solution to a customer, but couldn't merge its own products into an integrated solution as promised. The project manager from the vendor admitted, in effect, that the company was organized around—and salespeople were compensated for—products, not integrated solutions. The result? After the once-helpful sales team sold its respective product software licenses, there was no financial incentive for it to stick around and help the project team figure out product integration issues. The multi-million dollar project got cancelled after the client had already paid for all the licenses.

- **Having a vendor crush:** Some CIOs and IT managers select a vendor based on name recognition or what sporting events it sponsors. (I attended a meeting once where the organization staff was more interested in the picture of the vendor's famous golfer spokesperson on the cover of its presentation than what was in the actual presentation.) Then there's the case of vendors bearing gifts to remain in a client's good favor. Picture this scene: It's a couple of weeks before Christmas and a queue of IT vendors make their annual pilgrimage to the CIO's office, bearing gifts like the three wise men, bringing the CIO incense, myrrh and Virginia Honey Baked Ham.
- **Giving the vendor the keys to the castle:** Organizations can make the mistake of giving their system/software vendor too much control over a system project by letting the vendor make many of the major decisions, or allowing it to unilaterally determine project approach. This can lead to vendors making self-serving decisions. Organizations fall into this trap because they don't have the requisite technological expertise or don't have a structure in place to adequately manage large system projects and its associated vendors.
- **Believing a vendor can do it better:** Some organizations look for IT functions to outsource to a vendor/service provider either to save money, or because some functions are deemed not critical to the organization's function. The results on outsourcing are mixed. Clients do not always achieve the benefits they expected or were promised.

**Face the Truth:** When it comes to system/software vendors, don't put them on a pedestal. Treat them like the hired help they are (This also holds true for IT consultants like me and my ilk). To avoid making vendor-related mistakes, IT departments should employ the following safeguards:

- **Develop strong contracts:** The vendor relationship begins and ends with the contract. Components of a good contract include detailed customer and system requirements, a fixed-price structure when possible (but only if there are *detailed* requirements), payment based on completed and accepted deliverables, user licenses not purchased until *after* (please notice the emphasis) the system has been developed and user-accepted, incentives and penalties based on defined performance metrics, and definition of who owns the system and source code. Organizations like government agencies and large corporations have more ability to dictate contract terms than other organizations; this won't change until more mid-size to small organizations insist on the items listed above.

Additional suggestions for controlling vendor contract costs include incorporating technology and software performance bonds in contracts (surety companies offer these), or implementing invoice retainage (a percentage of each vendor's invoice is retained and paid only when the product is delivered according to the conditions of the contract). If contract terms or conditions are not met, the vendor forfeits the retained amounts.

- **Watch the vendors like a hawk:** IT departments should have staff with the experience and ability to monitor system/software vendors. If the organization doesn't have this skillset, hire it—even if it's a contractor—because this person will have the organization's best interests in mind. Additionally, organizations can use an IV&V (mentioned in Truth 3) contract as a means of monitoring not only the health of an IT project, but also the vendor's performance.
- **Outsource only what's necessary:** Organizations should be cautious about what systems and IT support they outsource, especially if the outsourced function is not going to be performed on the organization's premises. Administrative systems (e.g., payroll, email, data storage) are typically safe areas to outsource. However, mission-critical and customer-facing systems should be carefully considered because the organization has to make sure that a vendor doesn't interfere with the organization's ability to serve its customers. A good partial outsourcing approach involves outsourced functions being performed onsite. The vendor's staff perform the functions, but they're physically located where the organization can manage them and provide oversight real-time.

Successfully managing system/software vendors is definitely a big challenge, but manageable when utilizing Truth 7 tips. Speaking of big, the next section will help you understand large IT projects and their accompanying pitfalls.

## TRUTH 8: IF IT'S TOO BIG TO FAIL, THEN IT MOST LIKELY WILL

During the financial crisis that became fully clear in 2008, a number of companies previously perceived to be “too big to fail” had to be bailed out by the U.S. government. This is analogous to what happens within some organizations—a system project so big that the success of the organization supposedly hinges on it. The organization feels it has no choice but to continue to pour money and resources into the project even though it keeps getting worse. For some of these projects, the plug eventually gets pulled; but more often than not, organization employees trudge dutifully along to implement at least something. The result: a partial system limps into production (and is often called a success) that accomplishes just a fraction of the original intent.

Large systems projects fail to live up to their objectives due to some of the following contributing factors:

- **Ignoring the gorilla in the room system for too long:** Staff can typically quickly name the albatross, Edsel (or other appropriate metaphor) systems in their organization. It may be the COBOL mainframe application that processes the majority of the organization’s transactions that only two people in the organization know how to program. (Wait. Susan just retired? Make that one person.) Perhaps it’s the workflow system that neither works nor flows. Making the mistake of ignoring a system’s failings—or planning for a major system replacement until too late—throws the situation into crisis mode, where planning time is sacrificed in favor of the need and speed to implement.



- **The “big bang” theory:** IT departments sometimes decide to undertake a massive IT project all at once. Occasionally an organization has no choice but to take this approach due to new legislation or other external factors. Or, organizations simply bite off more than they can chew. To complicate matters, an “all-in” approach may be employed that doesn’t provide options for smaller, more manageable implementations of functionality.
- **Inability to mimic production in test:** Large-scale systems span the entire enterprise and may need to integrate with numerous other production systems. When a system is so far-reaching that the IT department is not able to replicate the system’s full production functionality in the test environment, this is a huge risk. Case in point: A large telecommunications company had an order and billing system that integrated with a 30-year-old provisioning system. The order and billing system needed replacing, and it wasn’t possible to test integration with the live legacy provisioning system. The development team tried its best to simulate those interfaces. But absolutely no one should have been surprised when the system went live and scores of customers who placed service change orders lost their dial tone based on a choppy handoff between the new order system and the legacy provisioning system
- **The users will be fine:** Underfunded or under-considered organization change management initiatives are another significant contributor to a large IT system failure. (See Truth 10 for more information on this.)

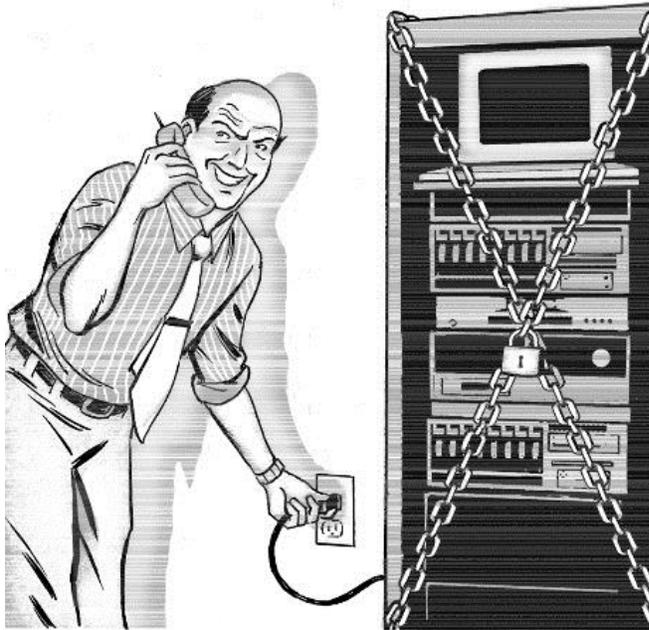
**Face the Truth:** Because there’s no bucket of bailout funds for failed large system implementations, organizations need to proactively get in front of bulky system initiatives to avoid them becoming unwieldy. The best method for achieving this is to mitigate the risk associated with large projects:

- **Plan and budget it:** When developing and updating the enterprise architecture plan and technology roadmap, the organization should identify the legacy or enterprise-wide systems it will need to either replace or implement within the next few years. The organization should document the planning time

and budget needed for these efforts. This provides the organization an orderly and measured approach to address large projects versus tackling them under circumstance-imposed duress.

- **Chunk it:** Large system development projects should be broken into logical chunks of functionality that are developed, implemented and demonstrated early and often. This helps identify critical missteps early, and if for some reason the organization is not able to fully implement the planned system, the organization may still have some working functionality.
- **Pilot test it:** If a new system must integrate with a legacy system and there's not a way to adequately replicate full production in the test environment, the IT department should conduct a small production pilot test with a limited number of users/customers that represent every user and transaction type. Everyone else continues to use the old system, and the two systems should operate in parallel until the organization is certain the new system works.

Big projects translate into big bucks. But so does the sum of all the other functions the IT department has to perform to keep the organization humming along. Thus, no article on IT would be complete without addressing the bottom line as discussed in...



*"If you want to keep your darling system alive,  
pay \$10 million in unrestricted funds!"*

Regardless of how open or commoditized IT becomes, for the foreseeable future it will continue to be one of the largest cost centers in any organization. Effective management of IT requires masterful and often creative planning, budgeting and spending of *very* limited dollars. (The formal phrase making the rounds in the U.S. government IT community that normally elicits sarcastic comments is: *Do more with less.*)

Newsflash: Staff in the IT department didn't get into IT to be bean counters. Against the backdrop of this inherent skillset mismatch, below are examples of difficulties most IT organizations encounter in managing the financial side of their work:

- **Inequitable or non-transparent funding methods:** An organization's IT department can be funded in a number of different ways, the most common being: (a) *Pay as you go*, where internal units pay by the IT system or project, and (b) *Put your money in the shared bucket*, where internal units are assessed an amount by the IT department to pay for infrastructure and a number of shared services like email and other support

systems. Well-funded business units typically don't mind *pay as you go* because money talks, and no IT department worth its salt is going to pass up a paying customer. Less-funded units don't necessarily like this approach because their projects seem to get bumped for more lucrative ones. Universally, all units have nothing but disdain for the "one bucket of money" approach—mostly because IT departments have a hard time telling their internal customers what they're getting for the money they've plopped into the bucket. Further, business units feel that the IT department diverts resources they have "paid for" to respond to other projects or high-priority issues that crop up, and well-funded units are suspicious that other units are getting a free ride.

- **Inability to accurately estimate:** Estimating is an art, not a science. Most IT pros don't like the ambiguity involved in estimating, and typical IT organizations don't invest time and effort in establishing estimating frameworks. In addition, IT staff typically prefer to wait until the investment details are fleshed out before committing to a number. Then there's the fact that many estimates include the cost of the IT product or service but fail to capture the corresponding costs of the ongoing operation and maintenance (O&M). This can lead to an IT budget shortfall that impacts the entire organization, year over year.
- **Not developing or updating the business case:** Every dollar is precious, and the best way to fund an IT investment is by demonstrating the value to the business and mission. The investment's business case is the vehicle used to articulate its potential value, timeframe for value realization and risk assessment impact. The business case should be the most important artifact of the investment's lifecycle, but I've found that it is often not developed, or considered an unnecessary paper exercise that is performed with the least amount of effort possible. Even with little or no business case documentation, many investments get funded and then end up not achieving the value that was expected.
- **Waiting too long for infrastructure refresh:** Business units care about having their applications and tools available and performing effectively when they need them. But viscerally, they don't understand or care about underlying IT infrastructure. In fact, most organizations fail to understand the value of adequate funding for updating and refreshing the infrastructure at the end of its service life. If this process is under-funded or ignored, it leads to out-of-warranty or end-of service-life equipment and software that costs more to maintain and introduces greater risk to the organization.

**Face the Truth:** Because technology is based on ones and zeros (that have dollars signs in front of them), IT departments need to become more adept at the financial aspects of IT to effectively manage IT. Some sound approaches related to departmental and project funding include:

- **Show me the money:** First and foremost, the organization has to determine if business units receive IT funds as part of their annual budget, or if all IT funding is included in the budget the IT department receives. Sometimes the funding is split—system development dollars are included in business unit budgets, and after a system is in production, the O&M dollars become part of the IT department's budget. IT departments need to be prepared to show their internal customers how their money was spent by first identifying the IT shared services components, and calculating and communicating the cost of them (e.g., the cost of email, LAN, service desk support, etc. per person). For the services that cannot realistically be identified on a per-user basis, IT should develop, *with input from the business units*, the method for allocating these costs proportionately across all units. In-year additions or new costs should also be calculated, allocated and documented.

Lastly, and very important, the organization needs to have rules of road for those cases where a business unit has a crucial IT need but not the funds to cover it (and robbing Peter to pay Paul should not be the most viable solution).

- **Don't guesstimate, estimate:** IT departments should identify and implement an estimating framework that leverages several methods to achieve the most precise estimate. No purchase of new technology or service should be allowed without an estimate for the operations and maintenance cost. Further, the cost of completed investments needs to be captured for future investment estimating. This helps ensure future funding needed to support IT.

Many of the IT infrastructure estimates should include the cost to refresh. In addition, the value of the refresh should be tied to the value defined in the investment's business case. Industry best practices suggest developing a rolling schedule that has 20% of the infrastructure refreshed per year. This keeps capital cost of refresh predictably tied to the annual budget process. The organization may also explore other methods of keeping the cost of infrastructure steady, such as commercially managed infrastructure (outsourcing) or infrastructure as a service (IAAS). Keep in mind that although outsourcing or IAAS can provide ease of management and a more steady cost structure, they can also end up costing as much or more than managing it yourself.

- **The case for change:** Investment should be accompanied by a business case that includes an analysis of alternatives with costs, benefits and risks compared to the status quo. The business case should be a living document and be updated as more information becomes available. It should also serve as a tool to assess the success of the investment in achieving the stated benefits and providing defined value.

Okay, I've got to come clean: In an article about truth, I was untruthful. The title for Truth 9 is, at best, a half-truth (at worst, a lie). While we can all acknowledge money is important, it pales in comparison to something much more important. It's not all about the money. Instead, for your consideration, I present...

## TRUTH 10: NOPE, IT'S ALL ABOUT THE PEOPLE

An organization could implement all of the preceding recommendations, but if it doesn't understand that the IT department—like all other organizational units—really functions through *people*, and not just servers and systems, then all is for naught. Conventional wisdom states that a unit's problems can be traced to its leaders/managers. More so with IT departments due to the complex juxtaposition of leaders with very technical skillsets managing an extremely technical area, with the management of the most non-technical element of an organization—people. (As much as it may want it, the IT department doesn't get a pass from managing the softer aspects inherent in running an organization.) The result is more opportunity for personnel-related missteps that can deeply wound organizational performance.

Some common IT-related people problems that hamper the successful management of IT departments and staff include:

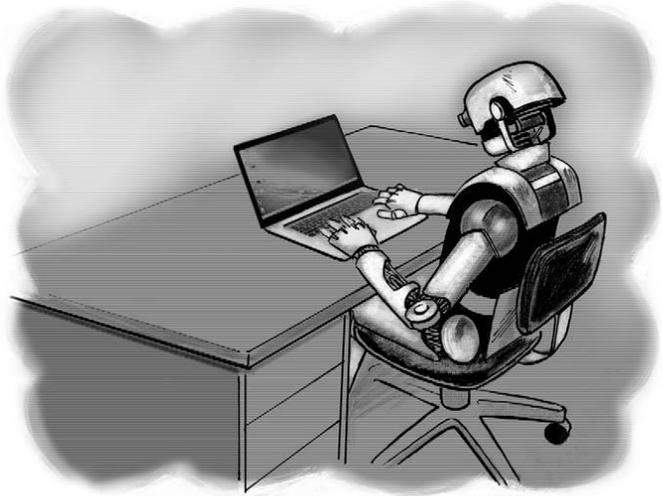
- **Not all skills are transferrable:** Many IT leaders and managers rose to their position based on their technical acumen. Unfortunately, being able to code in multiple languages, or being able to assemble a server rack with the best of them, doesn't translate into being a good people manager. When faced with technical or programming problems, the technical side of these individual's brains kick in and they use brute logic to fix the problem (If: X, Then: Y). When faced with people problems, the same type of thinking backfires. Employees typically don't respond well to the same logic-based solutions. Frankly, because of how the human brain is wired, the skillsets inherent in being technically adroit may not *naturally* coexist well with the communication, interpersonal and interaction skillsets required to effectively manage staff. (Again, my observation, not pulled from some study I found and can cite.)

Based on my experience observing leadership groups comprised of business unit and IT executives, this shortcoming may exhibit itself in IT executives' ability to effectively work with their non-IT department peers, especially in navigating amicably through differences of opinion. (But in fairness to IT leaders, my primary point of reference for over 15 years has been IT managers and CIOs, and possibly their behavior could be in response to the longstanding tension between the IT department and business units discussed in Truth 2.)

- **Not enough CIO trust in the managers:** Instead of relying on their staff, some CIOs—due to their actual or perceived superior technical knowledge—want to be seen as the smartest person in the room and can't resist constantly being the first to propose technical solutions. Or worse, they second-guess the approaches proposed by their IT managers. These actions send the message that the CIO alone will decide the best technological approach, without input from line managers or staff that are highly skilled in a particular IT area.
- **Not enough diversity in IT:** Dearth of diversity—not from a race, gender and age perspective (that's all a separate article unto itself), but from a skill set and business unit perspective—is a reflection of how IT departments may not have the correct mix of staff to perform some of the less technical work of IT. They tend to rely on fairly homogenous skillsets to conduct a wide range of functions.

Without a full complement of diverse talents, staff backgrounds and business unit representation, IT organizations fall into unfortunate scenarios such as:

- IT leaders and managers make what are organization-wide, business-related decisions because they “know best” about the technological aspects of a business need. The tail wags the dog and the technology solution ends up driving the business decision.



A CIO's dream: Robot Programmers

- Technical IT staff write documentation and expect system users to understand it. For example, these techies gather requirements from business users, then ask users to validate complex depictions (can someone say UML diagrams?) versus presenting the requirements in a manner business unit staff understand.
  - Development managers, who assume quality assurance is synonymous with testing, have developers test and ensure the adequacy of their code, versus individuals versed in quality management that ensure quality of not just code, but of development practices and outputs.
  - Multiple skillsets are grouped into one job position. I recently saw a job posting to hire a single person to perform the following roles: project manager, business analyst, and data migration subject matter expert. Performing two out of three of these roles may be doable (e.g., project manager and business analyst), but having one person perform all three roles is not advisable.
- **IT doesn't understand the users' jobs:** Sometimes IT is far removed from the work an organization's people perform. In large organizations, an IT department may be located in its own building or even on a different continent. This can make it difficult for the IT staff to understand how the system's users operate. For example, at a company I worked at years ago, I was involved with a large-scale implementation of a new, custom-built order and billing system. The problems were too numerous to count. I led a tiger team (yeah I know, that's what we called it), which included business process analysts and system programmers who were dispatched to multiple call centers to troubleshoot issues and identify the causes of the problems. After a few days of collecting pages of notes, one of the programmers made the following statement on a call with the CIO: "It would have been good to have visited the call centers to observe how customer service performs its job *before* we developed the system." (Unfortunately I can't repeat the CIO's NSFW response in this article.)
  - **The people aspect of IT projects is underemphasized:** Change management was mentioned in Truth 8 related to large-scale IT projects, but is not solely a large project concern. We in IT are naive to think that a system we're developing will be welcomed with open arms. We are wrong to assume users will automatically recognize that the new system is better than the old one and that the change is good (ever try to convince your elderly parent the benefits of an iPhone and texting?). We fail to understand "fantasies of punishment," a concept a graduate professor of mine taught our class: *People don't automatically fear change' instead, they fear how the change will adversely affect them.* When it comes to systems people use at work, even with a legacy system they despise, there's a level of comfort in proficiency. They've learned years of tricks and workarounds to get their jobs done. Conversely, staff also like the crutch aspect of legacy systems and the ability to blame their productivity issues on a crappy system.
  - **People-related cost-saving decisions can have adverse effects:** People are like any other resource involved in producing a product. You get what you pay for. Sure, offshore development resources are cheaper, but as more organizations are realizing, they come with hidden costs. Additionally, organizations that release request for proposals (RFPs) and select the low-cost IT service provider risk subpar project and service quality—and vendor employees that may not be paid enough to be highly skilled.

**Face the Truth: Information Technology is just as people-intensive as any other function within an organization.** To get in touch with its softer side, IT organizations should perform the following:

- **Assess people skills:** An organization should assess its IT department staff's people skills, beginning with the CIO. Because of the CIO's increased role at the organization's executive table, the CIO needs to be more leader, less techie. In large organizations with multiple business units vying for resources and IT services, the CIO position is like being the head of the United Nations. The job requires deft people, negotiating and diplomatic skills. Further, an organization needs to hold the CIO accountable to the same people-related management principles the other organizational leaders are held to.

Some large organizations, understanding the business and people-related skills needed by the CIO, have two senior IT positions: a CIO and a Chief Technical Officer. The CIO may be the more polished and customer-oriented of the two; this allows the CTO to focus on the nitty-gritty technical aspects of IT operations without worrying about managing the SVP of sales' delicate ego each week in the senior leadership meeting.

After evaluating the CIO position, the people-related competencies of the IT managers should be assessed. A 360-degree evaluation survey can help pinpoint those managers that have room for

improvement in managing staff. And finally, an inventory of frontline IT staff skills can be conducted that compares technical and non-technical skillsets needed to address all the elements necessary to assure quality IT processes. (This includes assessing skills needed to introduce and manage new technologies based on the IT roadmap.) Once an organization gathers this information, it will have a portrait of its IT department's technical and required non-technical skills, including collective strengths and weaknesses.

- **Ensure a breadth of skill sets in IT:** To be successful, IT departments need skillsets related to the entire lifecycle of IT management. In addition to technical staff, IT needs people capable of working with business units to gather business needs (requirements), understand business processes, facilitate information-gathering sessions, communicate IT activities, train staff on use of systems and conduct organizational change management. Some of these skill sets are possessed by business-unit staff, which strengthens the case for ensuring PMOs and IT projects are staffed with staff from multiple areas of the organization.
- **Understand and integrate change management:** IT needs to better understand that successful implementation of systems boils down to two facts: (1) end users determine if a system is successful (regardless of what leadership says about the success of an implementation), and (2) the users are either with you or against you. If adequate change management isn't employed on a system project, the IT department will be 0 for 2. IT departments need to plan and budget ongoing change management efforts—performed by people who understand the unpredictable and hard-to-please nature of system users and recipients of IT services.

Additionally, an organization needs to understand that change management does not end when a system is implemented. Ongoing effort is needed to ensure users can effectively use the system with the businesses' processes, and to make sure legacy system habits are not replicated in the new system. Further, the other type of change management (i.e., system change requests) takes on increased importance. No system is perfect when implemented. The IT department has to have a means of gathering users' system-related complaints and suggestions to address them as expeditiously as possible. And that data cleanup that was promised to the users (mentioned in Truth 6)? Don't put it off, just get it done. In the mind of the users, crappy data equals a crappy system.

- **Locate IT within the overall organization:** Ideally, an organization physically locates the IT unit onsite. When that's not possible, the organization needs to ensure that it trains and acclimates its IT staff so they understand the business, its mission and its operations. The IT department should strategically place staff liaisons throughout the organization that are part intelligence gatherers, part customer service/trouble shooting ambassadors.

## THE TRUTH BE TOLD

---

**P**resident James Garfield is credited with saying, "*The truth will set you free, but first it will make you miserable.*" Such is the case with the 10 truths of successfully managing IT. Implementing all the concepts may seem overwhelming. But, if you understand the truths and apply the suggestions to address them, the payoff is huge. You'll give your organization the poise and balance to make it across that technological and organizational tightrope in a way that would make Barnum and Bailey proud.



---

**About the Author:** Harold Kerr is the owner of [The Kerr Company](#), which provides services that range from traditional management consulting to information technology consulting.

