New Technology IT Project Management

Dr. Russell Davis, PMP
CEO, ***Femtosecond***®

Introduction:

Most IT projects end in disappointment. Many will argue that adequate project management is sufficient to ensure any project success. This paper describes an improvement using system-wide subject matter expertise in conjunction with project management. This approach is necessary to ensure that the project constraints are achievable. In addition to the accepted cost, schedule, and quality constraints; this paper includes customer requirements, software reliability (vendor and developed), networking, and processing availability. Quickly identifying project constraints where the current scope is unachievable is critical to success.

Some Reasons IT Project Fail:

Why do IT projects have such a high failure rate? Consider the uncertainties when deploying new IT. The product likely does not have an established record. Those involved lack sufficient training and an ample understanding of the product. Then there are the uncertainties in trying to integrate or upgrade the product into the production environment. There is also the issue of vendor marketing that may not reflect actual product capabilities. How many products have been purchased based on a demonstration that never seems to work in the production setting? Can you recall a product purchased based on what was presented as a live demonstration only later to learn it was an off-line exhibition of capabilities? Have you ever experienced a vendor that made promises before purchasing the product only to claim ignorance after the fact? Have you ever used any products that should still be in Beta testing but are the latest commercial product upgrade?

Trying to get IT project success can be laborious when the complexity or size is significant. Consider the case (GAO/T-AIMD-97-91) where $80 million was sunk into an originally estimated $151 million project only to have it killed after three years. In a recent Exhibit 300 training course I recently attended, an un-cited analysis of 700 major contracts since 1977 concluded that IT projects 15% into the project that are over schedule (Estimate at Completion) do not make up for the loss. The claim is if the near-term was underestimated; there's little hope that far-term planning will be better.

Problem Space:

IT project requirements and constraints tend to increase in number and can quickly lead to unachievable or marginal results. There is a set of problems known as non-deterministic polynomial (NP Complete). For small data sets typical of lab testing environments, producing results is easy. However, with a few problem set additions typical of production environments, computing becomes unrealistic. What vendors could end up doing is to shortcut the processing thereby introducing additional ambiguity into the process. A laboratory environment may consistently successfully demonstrate all system requirements only to have the final product fail in actual use. To illustrate the complexity problem; consider the case of the traveling salesman that wants to find the shortest path for traveling between cities. Using techniques such as dynamic programming, the results are constrained exponential time ($2^n$). There are additional calculations but $2^n$ quickly becomes the overriding problem. The next figure uses a three state problem where

Indiana (with 25 locations) and Pennsylvania (with 18 locations) are separated by Ohio (with one site).


Indiana 25 locations

Ohio 1 location

Pennsylvania 18 locations

Figure 1 Three State TSP Problem

If we used dynamic programming, the dominant calculation would require $2^{(25+1+18)} \approx$ 18 trillion combinations. We could have tested in a laboratory setting with limited cities and observed excellent results. However, once applied to the real world environment, the processing requirements could culminate in random or frequent failure.

Note, however, that unlike the computer, the human observer sees that Ohio must be part of the shortest path. The problem can be divided into separate problems parts. Thus we can reduce the original problem into $2^{25} + 2^{18} \approx$34 million. Thus a simple problem division can be the difference between a successful project and one that hopelessly fails. This is a simplistic example where subject matter expertise can provide a significant return on investment. If errors such as these are missed, the project will likely fail.

Failures are not limited to technology problems but also include regulatory constraints. For example, trying to complete a certification and accreditation without having adequate time and resources could lead to project termination.

Consider the example where a Department was the first to issue the Homeland Security Presidential Directive (HSPD-12) transitional identity cards with digital certificates path validated to the common policy. These cards were issued over one year before required. A new highly capable and qualified project management team was put into place. This team decided to start fresh and the department ended up withdrawing from the previously identified line of business.

**Amortizing Costs:** Sometimes projects are deployed that include redundant or overlapping capabilities. That is, one project could meet requirements of another. Consider Homeland Security Presidential Directive 12 (HSPD-12) which institutes the requirement for a consistent government ID based on strong Identification and Authentication (I&A). Executive branch agencies are required to meet this requirement that includes deploying a Personal Identity Verification (PIV) card. As the PIV cards are required, couldn't projects such as the Transportation Security Administration's (TSA) "Registered Traveler" program use the same technology?

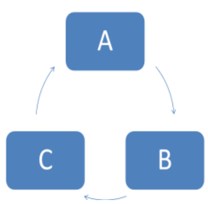
Consider the situation where a TSA screener would be confronted with two travelers. One has a current background investigation as required by HSPD-12, works for the Government, and this can readily be proven. Should the PIV card holder be granted the same fast access as the Registered Traveler? If the same fast access is granted to PIV card holders, then both projects would realize benefits. This is the argument that ***Femtosecond*** made to the TSA during the Registered Traveler comment period. Similarly, the First Responder Access Credential (FRAC) and Transportation Worker Identity Credential (TWIC) cards follow the same logic. Interestingly, the Federal Enterprise Architecture has a goal to reduce redundancy in the Federal Government. We sincerely hope the Office of Management and Budget (OMB) does not miss this prospect. As new technology continues to become available, the ability to come up with optimal solutions will continue to require human observation to identify opportunities not otherwise obvious.

Project Constraints:

As projects evolve there is a tendency to increase the requirement count and thereby the scope. Reasons vary for adding requirements including new regulations, features not originally considered, and stakeholder change requests. Sometimes project management becomes the art of negotiation where needed resources are only available for exchange. Some of these constraints are explored further in this paper.

**Customer Requirements:** The customer may request changes that look simple but are not achievable. Many project managers would not believe this to be the case and take the path to failure. Moreover, the available subject matter experts (if available) may not be knowledgeable in algorithm complexity. Consider the partition problem where a set of integers is divided into two halves with the same value. On the surface, this seems to be an easy problem and lab testing would work quickly for smaller data sets. However, this problem is NP Completed and when using more numbers there will be times the system seems to hang. Have you ever experienced an application where it works most of the time and yet occasionally hangs? And if the inputs are obtained from random sources, the problem might never be reproducible.

Consider a request to add another data feed to an application. Further, assume there is a compelling business need for this addition. But what are the overall impacts? Are there timing, network bandwidth, data conversion, or security issues? Might there be other constraints not immediately obvious? Are there other dependencies that could lead to a deadlock? To understand deadlock, consider resource **A** requesting information from resource **B**. If resource **B** then needs information from resource **C** to satisfy the original request, the complexity is increasing. Now consider that resource **C** in turn needs information from resource **A** to satisfy the request made from **B**. This deadlock is depicted in the figure to the left. Each resource waits for the input of another resource that can never be produced.




By careful analysis of the data sources to include timing, race, and deadlocks, it is possible to determine upfront that the customer request is unfeasible. Once a problem has been determined to be unfeasible, options include either alter or eliminate the request. For example, the information that resource **A** requests from **B** could be limited to what is available within **B** so that external requests do not take place. This, in effect, breaks the chain of dependencies. To achieve this, a system analysis needs to take place.

**Software Reliability:** How many new commercial software upgrades have problems? In today's marketplace, some companies rush to market with the expectation that once a product is in place, the cost to change products will be too great for the customer to change products. Thus the software is deployed with problems. If a project upgrades to a new software version, the true cost may not be known for some time, typically when the patches start becoming available. By continuously monitoring reports on the proposed product, in conjunction with adequate lab testing, a better risk determination is possible.

Software developed in-house provides better control but at a higher cost. The challenge for software development is to ensure the algorithms are sound. Software developed to flawed algorithms results in risk to developers.

**Available Processing:** In many cases, there are system limitations that prevent processing completion. For example, one agency developed an application using a commercial cryptographic application programming interface (CAPI). The programmers had a single threaded program running on a Windows server. The time required to complete the initiation context was 5 seconds; and the Internet Protocol (IP) timeout for trying to connect was 60 seconds. In lab testing with three test subjects it always worked. However, once the application was deployed, it operated for less than 10 minutes before the IP connection attempts exceeded the capabilities of the server forcing a hard shutdown. What appeared to be happening was that once the number of transactions was longer than 60 seconds, multiple attempts to re-establish a connection quickly resulted in tens of thousands of attempts. After this problem analysis, the developers used a cached context that required approximately 5 milliseconds to perform the action. This was adequate to process the entire agency without risk of exceeding the IP connection timeout. However, if the same application were to be deployed at a large Department, it would be possible for the same scenario to develop.

**Obsolesce:** There is another dilemma when the solution deployed is inadequate to meet the needs of the agency. Consider the case of the Reserve Component Automation System (RCAS) that had procured a large number of hardware Data Encryption Standard (DES) encryption devices. Analysis confirmed that the devices were not expandable and the encryption key was fixed (could not be changed). Moreover, these devices were comparatively slow for the network connections and could not handle increased bandwidth. It took several months and a number of high level meetings. The US Army concurred with the assessment and realized cost avoidance exceeded $10 million (obligated funds minus sunk costs). This did not include the operation and maintenance costs. There would have been a cost for disposal as well.

Quickly Identifying Defects: Important to reducing impacts is early recognition or removal. Unfortunately, there is no application or process that can fully capture years of experience. It is this system-wide experience that ***Femtosecond*** offers. The more subject matter expert experience, the greater the ability to detect system-wide problems. In the case of deadlock detection, systems often lack the documentation that fully describes all interdependencies. Consider the previous deadlock example where the configuration settings use the cached information available within resource **B**. Now consider that the administrator maintaining resource **B** makes a change (configuration, patch, upgrade, or security setting) that now forces an information refresh before processing. All of a sudden, a seemingly insignificant change produces an environment that results in deadlock. The problem becomes more arduous when the data is pulled at irregular intervals.

Defect Analysis:

Defects typically have the unwelcome effect of increasing cost and delaying implementation. If defects are not prevented or identified early enough, each can have a schedule impact. The schedule then includes the sum of delays. Each delay is expressed as a non-overlapping contribution to the overall schedule. Similarly, the cost is the sum of non-overlapping costs. As the number of defects increases, the risk of project failure increases. The next figure illustrates schedule and cost with defects[1].

$$C = C_b + \sum_{i=0}^{n} CD_i$$

Where
***C*** = Cost
***n*** = number of defects (problems)
$C_b$ = Base cost without defects
$CD_i$ = The Cost per defect ***i***

$$S = S_b + \sum_{i=0}^{n} SD_j$$

Where
***S*** = Schedule
$S_b$ = Base schedule without defects
***n*** = number of defects
$SD_j$ = The schedule impact (additional time) for defect ***j***

Success Stories: ***Femtosecond*** was supporting a large Department in assessing a vendor HSPD-12 implementation. We were attempting to construct digital certificates path validated to the Federal Public Key Infrastructure (PKI) Common Policy (PKICP). Within the certificate is a mandatory value called the Federal Agency Smart Credential Number (FASC-N). In a multiparty Government and contractor meeting, just prior to the schedule critical path deadline for issuing these certificates, it was discovered that there was no implementation available. None of the contractors could provide a FASC-N capability in time to meet the deadline. This meeting concluded on a Friday. The following Monday, a source code implementation for generating FASC-N values was posted to the ***Femtosecond*** webpage (http://www.femto-second.com/Code_Samples.htm). Accordingly, the pilot that would otherwise have failed instead made the milestone. The ***Femtosecond*** developed code was used to generate the initial FASC-N values for the remaining digital certificates during the pilot. The code was made available for free with no intellectual property stipulation, so we have no way to ascertain what other implementations are based on this realization.

Closing Notes: In 2005 I resigned from the Bureau of Land Management (BLM) where I was the alternate Bureau IT Security Manager, a GS-2210-15 grade. One project I managed was the logical access project (using smart cards to log into Active Director) where we went from 200 users to over 5,500 within a four month period. I am a firm believer in the PIV card approach. The main drawback for Public Key Infrastructure (PKI) deployment has been in the end entity Identification and Authentication (I&A) which is addressed by the PIV card. Years ago, OMB A-130 required risk assessments. There were a number of companies offering risk assessments that had inconsistent costs and quality that destroyed the initiative. Recently, risk assessment based on consistent approaches has returned with the new National Institute of Standards and Technology (NIST) guidance. My fear was the same thing would happen to the PIV card. My Doctorate was earned at the George Washington University in Computer Science (with a minor in Artificial Intelligence). Additionally, I have been involved in all aspects of IT security.

[1] $SD_0$ and $CD_0$ are separate sums of cost ensuing from defect consequences (for instance new requirements).

# Femtosecond®

9747 Water Oak Drive
Fairfax, VA 2203-1029

Phone: (703) 282-1837
Fax: (703) 242-8142
email: rdavis@femto-second.com
**www.femto-second.com**