FIVE REASONS

WHY WAREHOUSE IT PROJECTS FAIL

Here are five reasons why warehouse IT projects fail and the steps to make your project succeed.

BY TOM WRIGHT,
FORTE SOFTWARE EXPERT

According to research by The Gartner Group, 74% of all Warehouse information technology projects fail.[1] Another survey conducted by the Warehouse Education Research Council (WERC) concluded, "Only 40%—4 out 10 companies were satisfied with the installation of their projects."

If you are considering a new project, chances are it will fail. Everyone will be impacted—the CEO, management, and your customers. The result is that you will waste money, time, and resources. I have seen it happen. I have witnessed the upset, frustration, lawsuits, bankruptcies and despair.

Over the past 30 years I have been involved in many system projects with a 90+ percent success rate. As I reflect on those experiences and the lessons learned, I have found that there are five primary reasons why most projects fail. My intention in this article is to share that knowledge with you so that you can apply the positives and avoid the pitfalls, thereby enabling you to contribute effectively to your organization's next project.

REASON #1: LACK OF LEADERSHIP

It is difficult to believe that an organization can commit thousands and even hundreds of thousands of dollars to a project and then fail to lead that project to success. Leadership is the one common denominator of successful projects.

What is leadership? Professor Warren Blank provides a powerful metaphor:

> "Leadership and physics share a common focus. Physics explains the energy, matter, and motion that define how the universe works. In the same way, leadership is the power that galvanizes human energy and translates it into action. So the exercise of leadership can be viewed as the practice of human physics."[2]

The leader is responsible for the vision of the future and instilling that vision in all of the people that are affected by the project. This is not easy, but it is essential to the project's success. The CEO does not necessarily have to be the leader, but if the CEO does not share the vision, the project will fail.

A few years back I was involved in a project to implement new order management software. The CEO's vision was that this software would improve customer service by giving clients visibility into the status of their orders at any time via the Web. It would also provide him with a means, at any time, to see his company's progress toward meeting commitments on any or all orders. In fact, the project included a digital dashboard that used vivid colors to indicate status: orders that were on time were displayed in green, those that had failed to meet established processing time goals were displayed in red, and those that were close to a deadline were in yellow. Unfortunately, his warehouse manager did not share his vision. In fact, the warehouse manager did not want the CEO to see the status of individual orders because his "failures" would be obvious. The CEO did not have the support of his subordinate – they did not share the same vision –and the project was doomed to failure. Ultimately, it did fail to get implemented. Professor Blank again:

> "**Natural Law 1: A Leader Has Willing Followers**
>
> No leader exists without gaining the support of others. Typically all glory and grandeur goes to the leader, and being a follower is usually thought of as a second-class or lower-status role. The first natural law of leadership changes our view of followers because it recognizes the collegial, partnering role they play. Followers are allies who represent the necessary opposite side of the leadership coin."[3]

My experience over the years has convinced me that for a project to succeed all parties impacted by it need to be a part of the project team or represented by someone on the team. And yes, this especially includes the end users! If an individual whose job is affected by the project is not provided an opportunity to participate in shaping that project, then that individual will not share the vision and will invariably sabotage the project, intentionally or unintentionally. The leader must align the team and ensure that all members share the common goal and are committed to the project's success. The leader must take care of the politics and attend to the team dynamics. George Eckles, the primary consultant for General Electric in their Six Sigma Quality Initiative agrees:

> "**Concern 4—Ignoring Team Dynamics as a Root Cause of Project Failures**
>
> By far the greatest source of team failures is poor team dynamics and poor facilitative leadership behaviors. Approximately 60 percent of teams that fail have these as their major reasons. I am reminded of an old adage spoken by some of my organizational friends. 'The hard stuff is the easy stuff; the easy stuff is the hard stuff.' The most common problem areas we have encountered are:

FORTE
THINK INSIDE THE BOX

[1] *PC Week*, December 5, 1999
[2] Blank, Warren. *The 9 Natural Laws of Leadership*, 1995, p27
[3] Blank, Warren. *The 9 Natural Laws of Leadership*, 1995, p11

FIVE REASONS

WHY WAREHOUSE IT PROJECTS FAIL

1. Meeting skills
2. Setting agendas
3. Determining the meeting's roles and responsibilities
4. Setting and keeping ground rules
5. Facilitative behaviors."[4]

REASON #2: LACK OF CLARITY

As you will see, all of the reasons why projects fail are related. Good leaders provide a clear and compelling reason why a project or initiative needs to happen. However, clarity also deserves special attention because projects fail to deliver the anticipated results when the results have never been clearly defined. James Lewis says this step is where many projects go wrong:

> "I often say that projects don't fail at the end, they fail at the beginning—right here in step 2. The reason is that we take for granted that we all understand the problem perfectly, when this is not the case at all."[5]

Several years ago I met with an electronics wholesale distributor who wanted to implement a bar code system. The project team included representation from the warehouse, shipping, receiving, accounting, and information services departments and included an outside consultant who was an expert in their enterprise resource planning (ERP) system. The Vice President of Operations was the team leader. Every group that was impacted by the project was to be represented at each of the meetings.

A conceptual design was developed that integrated bar codes with the ERP system. The system was designed in such a way that when products were received they were bar coded and entered into the ERP inventory system; when they were shipped out they were scanned and removed from inventory. After the design was approved, a detailed functional specification was written and signed off by each member of the team. The system was designed and went into production. It was tested, accepted, and the users were trained. The project came in on time and within budget and everything seemed perfect. However, soon after implementation, the president of the company called and asked me to meet with her. She was very upset: the bar code system was not fulfilling her expectations and was not working as she had envisioned. As such, she felt it was "a complete waste of money".

What went wrong? The problems were many. First, she had not been present at any team meetings. Consequently, her vision of the project's goal was not communicated and the software was not programmed to do what she wanted it to do.

A survey conducted by the London School of Economics reports that:

> "CEOs felt that only 33% of their IT spending resulted in an improvement to the bottom line."[6]

In order for projects to be effective, it is vital that its expected result be clearly defined. Dr. E. Rechtin, Professor of Systems Architecture and Engineering at the University of Southern California, elaborates:

> "The purpose of systems is to produce added value, system functions, that none of the elements alone could produce. Indeed, the reason that systems are built at all is to achieve those results. As examples, the principal system function of an automobile is transportation; the principal system function of the human body is life.
>
> Now, although each of the elements (heart, lungs, brain and so on) has its value, none of them produce the system function and, in many cases, none of them can operate independently. Each must be interconnected to others in order to function at all. The added value of a system, therefore, is not only in providing system functions but in also providing the infrastructure necessary for its elements to operate."

REASON #3: LACK OF DUE DILIGENCE

Sometimes we need to have a project implemented yesterday and we don't have the time or resources to first do our homework. While it is certainly possible to fast track a project, many projects fail because someone did not do the necessary research and gather the information required to make informed decisions.

Several months ago I received a telephone call from an operations manager of a fulfillment company who needed to have a system implemented within 30 days in order to secure a new client's business. He informed me of his plan to implement a software package that would produce a compliant bar coded label for each package that was to be shipped. I asked him why he had chosen that particular software package and he said because it was certified and had been recommended by one of his vendors. As I asked him additional questions, I learned that he planned to fulfill batches of 50,000 packages at a time. I suggested that the software that he was considering had been designed to process packages one at a time and asked if that was the process he really wanted. As a result of our conversation he realized that it would have cost him a fortune in labor had he implemented the certified software package but that a system which could produce mass quantities of labels at one time would be much better suited to his needs.

In every project there is the risk of failure. One way to mitigate risk is to perform research and gather as much information as possible relating to the project. Often we have neither the required expertise nor the time to acquire it ourselves and need to engage an expert or consultant we trust to provide unbiased information about aspects of a particular project.

Another advantage of using an expert is that they can focus all of their attention on the project. Most of the time, our project is not an

[4] Eckles, George. *The Six Sigma Revolution*, John Wiley & Sons, 2000, p24
[5] Lewis, James. *Master Project Management*, McGraw Hill Professional Group, 1998, p63
[6] *Computing Canada*, v24, June 8, 1998, p9

FIVE REASONS

WHY WAREHOUSE IT PROJECTS FAIL

employee's sole assignment. It is easy to get caught up in what I call the "tyranny of the present". Everyday emergencies prevent getting done those tasks that are essential for the project to be successful. It is what Stephen Covey refers to as the "urgent but not important":

I. Urgent — Important • Crises • Pressing problems • Deadline-driven projects, meetings, preparations	**II. Not Urgent — Important** • Preparation • Prevention • Values clarification • Planning • Relationship building • True re-creation • Empowerment
III. Urgent — Not Important • Interruptions, some phone calls • Some mail, some reports • Some meetings • Many proximate, pressing matters • Many popular activities	**IV. Not Urgent — Not Important** • Trivia, busywork • Junk mail • Some phone calls • Time wasters • "Escape" activities

A project's success depends on getting the important tasks done within the time they deserve.

REASON #4: LACK OF ACCOUNTABILITY

Most vendors, consultants, or project managers will promise to produce systems that fulfill a functional specification but are reluctant to promise the system will achieve specific business results. By accountability I mean a commitment to produce the business results, meaning the project is complete only when the results are visible. This means that the business results also have to be measurable. Can you see why clarity is so important?

A number of years ago, I became involved in a situation with the owner of an automobile parts distribution business who was angry because his expensive new system was not working. The IT vendor who provided the system claimed that it was performing in strict compliance with the functional specification that the CEO had signed. Unfortunately, however, it was not printing invoices, something the owner believed that it should do. The IT vendor said it was the client's responsibility to provide a required printer driver and the owner had no idea what a printer driver was, let alone how to provide it.

It is not uncommon for a CEO to feel helpless and frustrated when confronted with technology s/he doesn't understand. In fact, the Wall Street Journal reports that:

> "Eighty percent of CEO's recently surveyed in The Wall Street Journal say they cannot adequately evaluate their chief information officer's performance because they do not understand what the CIO does."

Most vendors do not take responsibility for the result – and they get away with it. What if the vendor was willing to really put his/her neck on the line and guarantee that the system solution would produce the business results the CEO envisioned? What if that company took responsibility for the entire system (i.e. software, processes, people and environment)? Would the project succeed? Would such a commitment allow the CEO to make a decision faster and sleep better at night? You get the idea. Someone has to be accountable for the results.

REASON #5: LACK OF A PROVEN PROCESS

In our fast-food, instant-answer society, we tend to skip over processes. We want the answers now, and we are not very patient. Recently, I came down with a cold. After trying to ignore it for about two weeks, I finally went to see my doctor. Of course, I would rather have had him simply call a prescription in to my local pharmacy. Instead, he had a nurse take my temperature and blood pressure. He listened to my chest, asked me several dozen questions, swabbed my throat, ran a culture and x-rayed my sinuses. As it turned out, my "cold" was really a sinus infection. He gave me a prescription for the proper medication and, in a few days, I had recovered. I say the reason his "project" succeeded was because he had a proven process. He used this process to accurately diagnose my illness and prescribe the proper medication.

In my business, we use a system called the "systems engineering process". I do not claim that this is the only process or even the best process in the world. Nor do I say that you need to implement this particular process for your project to be a success. However, some process must be in place if you are to maximize your results. Major technical research groups recommend:

> "Institutionalize the processes, metrics and tools. This is an integral part of successful IT project management. It improves sponsorship, minimizes risks and improves the probability of success."[7]

I have always had a preference for systems engineering because it looks at the whole picture. Benjamin Blanchard, Chairman of Systems Engineering at Virginia Polytechnic Institute, provides this definition of systems engineering:

> "Broadly defined, systems engineering is the effective application of scientific and engineering efforts to transform an operational need into a defined system configuration through top-down iterative process of requirements definition, functional analysis and allocation, synthesis, design optimization, test and evaluation."[8]

Systems engineering is a process that is evolutionary in nature. It considers the whole system and involves integration of the hardware, software, people, processes, and environment to address a specific objective or result.

Systems engineering consists of the following seven phases:

1. Requirements and Specifications (What)
2. Preliminary Conceptual Design (How)

[7] The Gartner Group, META Group and Standish Group
[8] Blanchard, Benjamin. *The Journal of the National Council on Systems Engineering*, July/September

FIVE REASONS

WHY WAREHOUSE IT PROJECTS FAIL

3. Design and Architecting (Details of How)
4. Production and Testing (Preparation)
5. Operational Implementation (Delivery and Integration)
6. Evaluation and Modification (Change Orders)
7. Deployment and Maintenance (Production and Support)

Each phase contains four components: activities, deliverables, milestones, and a review. Let's look at each of these phases individually and see what specific actions can be taken to maximize the chances of a successful project.

SYSTEMS ENGINEERING PROCESS PHASE 1: DEFINE SYSTEM REQUIREMENTS AND SPECIFICATIONS

This, the most critical phase of any project, is also known as the project's definition or discovery phase. You must understand the problem in order to solve it. An elegant solution to the wrong problem is useless. During this phase, you must define precisely what the system will do in terms of the expected results, the cost required, and the time needed for the project. One of the requirements should be the means of measuring the achieved results to determine if the project's goals have been met.

Sometimes we are tempted to take shortcuts here, and that is a huge mistake. Changing the requirements later during the production or implementation phases will almost certainly have a significant impact on cost, schedule, and ultimate outcome.

You must also gain agreement from all stakeholders on the requirements. A stakeholder is anyone who will be affected by the system and on whom the system must rely for its success. This includes end users and the owner. Remember my bar code system experience?

One method that is often helpful in getting everyone on the same page is to create a project charter. A charter contains the problem statement, a goal statement, identifies constraints, lists team members, and outlines a preliminary plan.

So, how do you do it? There are seven steps to the completion of this first phase:

1. Identify and meet with the people that are impacted by the problem that you want to solve. Clearly define the problem and identify its cause.
2. Define the goal of the new system. Create the big picture. Develop a shared vision. Clarify the specific results that you need to produce and define how to measure their success.
3. Identify all of the project's constraints. These may include budget, operating system, interfaces with other systems, the operating environment, and the time frame or schedule.
4. Prepare a business case or return on investment to assess the system's economic benefits.
5. Determine if the project makes economic sense and is possible to execute within the identified constraints. If it does...
6. Put it in writing. Prepare a document that contains the results of the previous steps and a plan that describes the major functions necessary for the system to achieve the goal within the required time frame and budget. This document is sometimes referred to as the "system requirements document" or the "statement of work".
7. Submit the document to your team to gain agreement, and validate that it is accurate and complete. Revise the document as necessary and repeat this step.

Sometimes a customer doesn't know what s/he wants or needs. I have found that one of the best ways to determine project requirements is to conduct a site survey, interviewing the potential users in the warehouse. I have always found that they were able to provide me with information that simply isn't available elsewhere within the organization and that information is crucial to the project's success. Remember due diligence?

This phase's deliverable is the signed and approved system requirements document.

SYSTEMS ENGINEERING PROCESS PHASE 2: DEVELOP THE CONCEPTUAL SYSTEM DESIGN

The conceptual system design determines how the system will work in order to accomplish the requirements and specifications identified and documented during Phase 1. The objective of this phase is to identify all of the system's functionality and determine the best means of accomplishing it.

This phase's activities include:

- White board discussions with team members on different design approaches
- Block diagrams or functional flow diagrams of the components and sub-assemblies, including their interfaces, inputs and outputs
- Early prototypes of the system
- Pilots
- Evaluation of design alternatives
- Trade-off analyses and selection of design.

Before this phase can be considered complete, each of the following questions needs to be answered affirmatively:

FIVE REASONS

WHY WAREHOUSE IT PROJECTS FAIL

1. Does this design fulfill all of the requirements?
2. Can it be accomplished within the project's constraints (i.e. budget and schedule)?
3. Have all of the risks been identified and sufficiently mitigated?
4. Will the system as designed be supportable?

This phase's deliverable is an agreed-upon conceptual design of how the system will work.

SYSTEMS ENGINEERING PROCESS PHASE 3: DEVELOP THE DETAILED SYSTEM DESIGN

This is where the details of how of the system will operate are worked out. The deliverable for this phase is the approved and signed functional specification document. The functional specification should fully describe the system architecture, all of the system's components, and how they interface. All of the hardware, peripherals and software are described. All deliverables should be identified. This is the blueprint for building the system. A well-written functional specification allows the builders of the system to focus on the task of building without having to make design decisions or ask questions of the designers.

The detailed system design document should include:

- Overview – a reiteration of the project's charter, the problems to be solved and the goal of the system.
- Environmental description – a description of the environment in which the system must operate.
- Design diagrams – showing each subsystem and component and the inputs, outputs and error conditions of each.
- List of standards to be used – any applicable standards that must be adhered to when the system is built and installed.
- Acceptance test plan – a plan that will test all the functionality of the system. The purpose of the acceptance test is to verify that the system operates as intended. The steps included in this plan describe how the operation of each system feature is to be verified. Each step contained in the plan should describe the purpose of the step, the specific function tested, the required inputs and the expected outputs.
- Implementation plan – a plan that details the roles and responsibilities of each member of the implementation team, the time frame or schedule for each implementation activity, and the sequence in which the activities will be performed.

SYSTEMS ENGINEERING PROCESS PHASE 4: BUILD AND TEST THE SYSTEM

The purpose of this phase is to build the system described by the functional specification. The activities of this phase include acquiring, building, constructing, coding and assembling the components of the system (i.e. hardware, peripherals and software). Each component and subsystem should be unit tested before being tested again as a part of the system. Quality should be controlled to ensure that all components meet specifications. The entire system should be tested to ensure completeness and that all functional and performance criteria are met. Testing is the only way to tell whether a component, subsystem or system can accomplish its intended functions and has all of the desired performance features. During this phase, any deficiencies found in components or processes must be corrected or resolved. The acceptance test plan must be modified to reflect any changes to the system's structure or functionality and any other components and/or functions affected by the changes must be modified and re-tested.

This phase's deliverable is a functional system containing the features and functionality described in the functional specification document.

SYSTEMS ENGINEERING PROCESS PHASE 5: PERFORM OPERATIONAL IMPLEMENTATION OF THE SYSTEM

The purpose of this phase is to deliver and implement the system in the environment in which it is to be operated. Since all aspects of a system cannot be tested in a laboratory environment, this phase is the crucial test of how well it will work in the actual working environment. I spoke with a vice president of a large food distributor who had recently implemented a bar code scanning system in a freezer environment. While the specification for the bar code scanner/terminal required it to scan barcodes in a -20° environment, it did not require that the display not fog up.

This phase is when the acceptance test plan is executed. At this point, all of the functionality described in the detailed design is demonstrated to the acceptance test team. It is vital that this review is completed successfully before the system goes into production. The last thing you want is to put a system into full-scale production and then have to pull it back out because it was not thoroughly tested. When this happens, confidence and trust in the system is severely damaged and regaining momentum can require a monumental effort.

This phase's deliverable is the test team's signature on the acceptance test plan.

SYSTEMS ENGINEERING PROCESS PHASE 6: EVALUATE THE SYSTEM AND MAKE ANY NECESSARY MODIFICATIONS

No matter how thorough the planning and the testing are, nothing ever goes absolutely perfectly. There is always something that someone, after seeing the system operate, wants tweaked or modified. Usually, it is a detail that no one thought of until that very

FIVE REASONS

WHY WAREHOUSE IT PROJECTS FAIL

moment. There is a mountain climbing adage that says you can't see what is on the other side of the mountain you are climbing until you get to the top. Now we are at the top of the mountain. At this point, we have tested every function and are ready to go into production, and then someone asks if we can make a "just a small change". Now a decision must be made as to how critical the benefit of the change is in relation to the time it will take to implement and test it. If it is not practical to implement the change at this point, it is added to the (hopefully short) list of desired future enhancements.

During this phase, the system's operational process is documented and operating instructions are generated so that they are available during the end-user training process.

This phase's deliverable is final system approval and the operating documentation.

SYSTEMS ENGINEERING PROCESS PHASE 7: COMMISSION THE SYSTEM

The system is placed into production and the end-users trained on how to operate it.

Most of the project's work has been concluded at this point, and it is easy to overlook the importance of user training. You can build the best car in the world, but an unskilled or poorly trained driver can reduce it to a crumpled wreck in an instant. I have found that hands-on training is the most effective way of performing user training. After all, a lecture isn't a very effective way to learn how to dance. Users seem to learn best by doing and having a trainer available to coach them through the process. The instructor can use the newness of the system and its inherent complexity to coach users in every aspect of its operation. When many users need to be trained, it generally works best to "train the trainer", directly training a few lead personnel and letting them train the remaining users.

The majority of support issues in the days and weeks following the commissioning of a system are attributable to inadequate or incomplete user training.

CONCLUSION

In summary, the secret to successfully completing any project is to use a proven process that ensures that you get the desired results. I have presented the five most common reasons that projects fail and have described on such proven process that you can use to successfully implement your project.

One last piece of advice: after you have successfully implemented your project – celebrate! You have succeeded where many have failed. Congratulations!

Tom Wright is a senior engineer of software improvement at FORTE, a distribution operations improvement firm. The firm provides planning, design, implementation and optimization services, enabling optimal levels of Distribution On Demand. FORTE can be accessed at www.forte-industries.com or by calling 513.398.2800.