Why IT Project Fail

Rasha Abdel Rahman

Department of Information Technology
Audit Bureau of Jordan,
{r.abdelrahman@ab.gov.jo}

Abstract. *Researches continually show that many information technology (IT) projects all over the world have difficulties to be completed on time or on budget or on scope, In fact many are cancelled before completion or not implemented.*

Project success is affected by many factors such as project team, suppliers, customers and stakeholders; the truth is that they can all provide a source of failure.

There are many different reasons for the failure of IT projects but the most common reasons are rooted in the project management process itself, this paper covers the key reasons of IT projects failures.

Why IT projects Fail

IT Project fails when it does not meet one or more of the following criteria for success:

- It is delivered on time.
- It is on or under budget.
- It satisfies user requirements.

Only a few projects achieve all three. So what are the key factors for IT projects failure? Organizations and individuals have studied a number of projects that have both succeeded and failed and some common factors emerged. A number of these factors are involved in any particular project failure and they are interacting with each other. Here are some of the most important reasons for failure:

Lack of Project Methodology:

Project Methodology or project lifecycle describes the approach that will be taken to carry out a project.

Lack of project methodology will force project manager to make on-the-fly decisions, based more on gut reactions than factual and objective analysis.

Project should follow a well thought-out route to avoid going in circles, getting lost, and hitting countless roadblocks. Taking unstructured approach is a risk that will lead to unstable results because things rarely fall into place by themselves.

Methodology varies greatly from project to project, taking into account environmental factors and project specifics. And, of course, methodology is relative to the size of and the complexity the project. The bigger the project, the more important it is to have this methodology. But regardless of size, every project methodology must address three core issues: planning, development, and implementation. By following a pre-defined set of guidelines and a migration path, you have something concrete to which you may refer to and measure progress against.

Poor Planning:

Planning is one of key factors that affect success of any project because "Fail to plan is a plan to fail".

Project manager should pay a lot of attention to this area and give it enough time and effort regardless of time pressure. He should be aware of bad results when project plan is none-existent, out of date, incomplete or poorly constructed.

To plan for a project is to set the foundation for project work by defining the tasks to be accomplished, and the timeframe, resources, staffing, communication, and costs involved in completing these tasks.

The competence of defining a detailed plan about how work is to be done depends on the project manager technical expertise. Lack of such expertise will lead to much generalized, in this case the project should have a technical leader whose responsibility is to cooperate with project manager to make detailed plans.

Here are some kinds of plans that should be done for any successful project are:

a. Risk Plans

Every IT project involves some degree of risks. Not doing an explicit risk assessment is one of the major problems with project planning.

Projects that do not have plan for handling risks are hit by sudden surprises and are left floundering with promised schedules and deliverables and can end up losing the client, because of that we realize the importance of an adequate risk plan .

Nowadays risk management becomes very prime issue especially when the project gets bigger, success here means creating a plan to assess the risks, the 'which', the 'what' and the 'why' of each risk identified and planned for.

b. Quality Assurance Plans

Projects must develop a QA plan as part of the overall project plan to explain the planning, implementation, and assessment procedures they will put in place to ensure that project outputs comply with business standards and best practice, as well as any specific quality assurance and quality control activities. QA plan integrates all the technical and quality aspects of the project in order to provide a "blueprint" for obtaining the type and quality of environmental data and information needed for a specific decision or use.

The project's QA plan should cover the listed issues:

- **Fitness for Purpose**

  Deliverables should be fit for purpose. For example, Projects should be internally consistent, up to standard, free of bugs, and perform well. This doesn't necessarily mean perfection, but fit for purpose consistent with the level of funding and project resources.

- **Best Practice for processes**

  Projects should follow best practice for creating their deliverables, e.g. technical design and architecture, programming, web sites, and data capture. This should include processes, workflow, tools, equipment, and methods.

- **Adherence to Specifications**

  Projects will be asked to develop their own specifications. This might involve requirements specifications, functional specifications, and/or technical specifications. Once specifications are agreed, deliverables must conform to them.

- **Adherence to Standards**

  Projects must ensure that their deliverables conform to company standards for content, metadata, interoperability, terminology, learning, linking.

- **Accessibility legislation**

  It's related to provide resources that are accessible to a diverse range of users. In order to achieve it we advise that all resources meet good practice standards and guidelines pertaining to the media in which they are produced.

When deliverables are supplied, project should provide documentation describing the QA tests performed and evidence of compliance.

The more forward, future-oriented and in-detail planning the higher the chances of success. Each and every activity that is expected down the line gets due attention.

Not only is this pre-planning well-documented, but also even after the project has taken off, if things don't exactly pan out as planned , the project manager does not hesitate to re-plan, avoiding project management failure, and readily incorporates the changed circumstances in their new version, so that future events are controlled.

Project plans should consider cost, resources, and requirements to succeed. Indeed these plans should be classified according to the time schedule, means that there should be a monthly plan, weekly plan, and a daily task schedule so that you can follow the progress of the project step by step.

Poorly-Defined Project Scope (Unclear goals and Objectives):

Project manager should understand the compromise between what they want to accomplish and what they are actually able to accomplish. When goals exceed the ability to deliver timely results project will fail for sure.

Successful projects always have a well–defined scope that states realistic goals, and attainable objectives, establishes clear milestones, defines benefits and deliverables, and conducts regular technical reviews and measurements. By this you can ensure that the project will be visible for all parties such as upper managements and clients.

Scope should be clearly defined as part of the project definition. Much of the work at that time is directed at agreeing the optimum definition of the project - both in terms of its deliverables and in terms of how it will operate. This scope definition will form the baseline against which potential changes are assessed and against which the project's performance is measured.

The concept of well-defined scope is affected by many factors, for example sometimes the goal of the project maybe partially clear because of poor requirements gathering in the definition stage of project, despite defining clear requirements need time and lots of communications, goals and objectives might be unclear because project users lack the experience to describe what they really require.

The project problems start with the three most common scope mistakes:

- Overrunning initial cost estimations.
- Over- or underestimating project schedule (This is a double-edged sword: Set a large timeframe and run the risk of the project becoming obsolete by the time it's completed. Set a small timeframe in relation to the amount of work required will put a strain on personnel).
- Miscalculating work to personnel ratio.

Vague Requirements, Poor User Input, Lack of User Involvement:

Nothing kills projects faster than giving users something they didn't ask for and then pretending they did. IT teams may be given a vague and informal set of requirements, and they, in turn, may not bother to follow-up with users or ask any questions, as a result they will build what they believe is needed not what users need.

Lack of user involvement causes a great deal of resentment among the corporate user-community, projects may be seen as something forced upon them by developers who only want to test out their new toys. Don't ever forget that projects are built to support end-users, not developers.

Requirements need to be worked out on both sides because there's a symbiotic relationship between users and developers:

- Users -who know the business processes best- need to clearly express their requirements and provide feedback on each project deliverable.
- Developers -who know what technology can be used to put those business processes into place- need to ask the right questions and not make any assumptions on what they think the users mean.

Scope Creep, Objective and Requirements Changes during Project:

IT projects suffer from two classical problems in project management:

- Scope creep.
- Feature creep.

Scope creep refers to uncontrolled and unexpected changes in user expectations and requirements as a project progress, while feature creep refers to uncontrolled addition of features to a system with a wrong assumption that one small feature will add nothing to cost or schedule

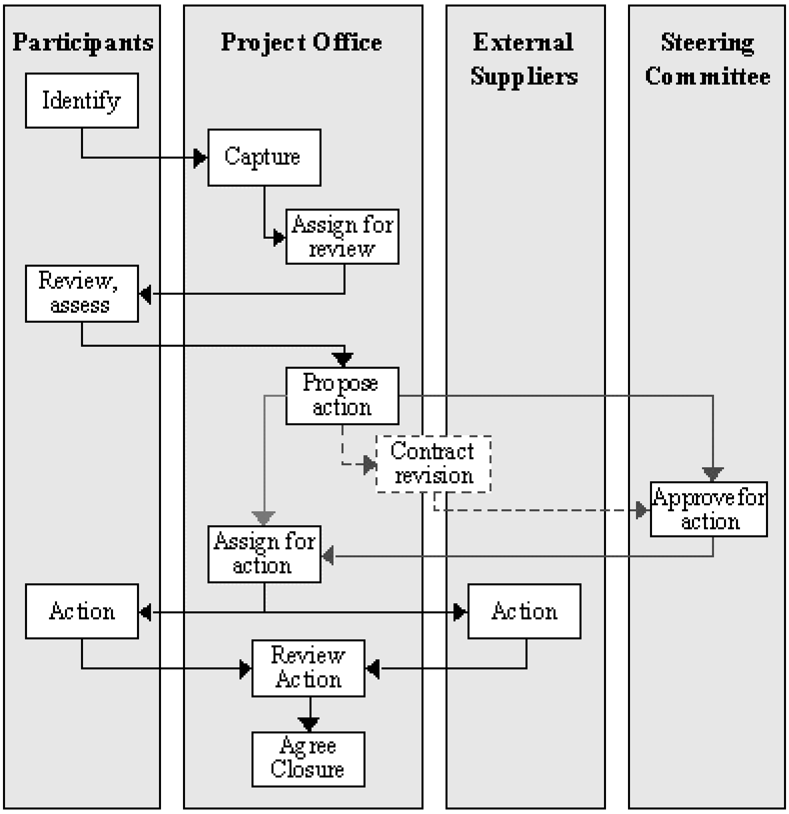
The project manager should understand project trade-offs and make the right decisions related to resources, features and time schedule even though requirements changes. He should be aware of the risks of change and the risks of not change and should have the ability to balance these risks before deciding what to do.

One obvious solution is to establish a reasonably stable requirements baseline before any other work goes forward. But even when this is done, requirements may still continue to creep. No one can design a process that assumes requirements are stable. In virtually all projects, there will be some degree of "learning what the requirements really are while building the project. Projects could be headed for trouble if architectures and processes are not change-friendly, or if there are poorly established guidelines that determine how and when requirements can be added, removed, and implemented and who will bear the cost of the changes .

On the other hand, is that if you architect a project from small, iterative phases instead of mammoth, serial deliverables, you will deliver more quickly, leaving less chance for change to overcome the work, and less risk of large projects failure.

Another key recommended solution for scope creep is change control process; the change control process will involve a combination of procedures, responsibilities and systems. The key to success is to have a well-controlled but efficient process. Define and agree:

- On what basis changes should be approved,
- Who does what,
- The membership of the change control board(s),
- The detailed procedures, forms, etc,
- Protocols for levels of authority, e.g. what types of change can be approved without reference to the project's business owners,
- Linkage to other management procedures, e.g. the issue management process, configuration management,
- Which tools will be used to support and manage the process,
- How to communicate and promote the process and its importance to all participants.



Example Scope and Change Control Process

Any participant or other concerned party may raise Change Requests. The Project Office team and Project Manager will ensure they are captured and proactively manage them to conclusion.

An initial review should be made to examine the need for change, how it could be achieved and what the consequences would be. The most appropriate member of the Project Team would normally perform this review. Based on those conclusions, the recommended action would be proposed.

In this example, there are three possible courses for the approval of the change:

- Minor changes within scope can be approved by the Project Manager.
- Any change affecting an external sub-contractor would need to be reviewed with that contractor who would agree any necessary contract revisions or payments etc.
- Changes of scope and contract revisions would require the approval of the Steering Committee (or it might have been a Change Control Board).

In making the decision, the Project Manager, Change Control Board or Steering Committee would be guided by the pre-established principles for making change decisions.

After the action is agreed the work is assigned for action by the Project Team and/or the external sub-contractor. When complete, the action would be reviewed and the Change Request closed. It is possible that the agreed action could have more than one stage. For example, it might be better to introduce a temporary solution so that the overall benefit from the project can be delivered, and then build a permanent solution after the system is live.

Poor Architecture which is Inflexible for Any Change:

Any environment usually develops, and according to this development many issues may change such as strategies aligned to this environment objective, requirement…etc.

The concept of what we are using today may be useless tomorrow is clear and understood. This concept should be considered when building any project. If the project architecture is inflexible for updates, then this project may die by time because of daily changes and rapid developments.

An unfortunate example of flexible architecture is the Patriot missile used during the Gulf War. It was not designed to intercept scud missiles, but the software was able to be reconfigured to support the new function. On the other end of the flexibility spectrum was a security program created to protect sensitive word-processing documents. Everything worked well for a few months until the operating system was updated. The word-processing programs still worked, but the security program became useless and unfixable because much of its code was tied to operating system features that were dropped in the new system.

People must think ahead about what is likely to change. If you do architecture right, you will not have to restart from zero again and rebuild the project from the beginning as nothing is existed because you are able to add and modify features that caused by any change any time, but if you do it wrong, you will suffer death by a thousand cuts. Bad choices show up as long-term limitations, aggravation, and costs.

Stakeholders' Conflicts:

All the stakeholders of the project should share similar business interests.

For example, assume that a project is being built, after while the developers need some clarifications, i.e., with input A, does the system choose X, Y, or Z? If stakeholders could not agree on answers this will force to acknowledge deep incompatibilities among their business interests, then the system will be canceled in an expensive failure for the entire enterprise.

It’s a problem when the stakeholders work under the illusion that everyone is going to get everything that they want. They will contradict each other by their differences rather than going through conflict resolution in the early stages. The developers will expose the stakeholders' irreconcilable differences because programmers cannot create an ambiguous system.

Stakeholder conflicts can play many different roles in project failures. Often, stakeholders have personal reasons for not being able to work together. When ego and pride get in the way of any project, it will almost always end in some disaster.

Other projects, especially smaller projects within larger projects, never go anywhere because the internal stakeholders never agree on priorities. This is called "pretend projects," meaning a few developers work on them half time or quarter time, and nothing is ever delivered, but what ever the case is, you should always think like this if you start any fixed-fee project you should end it according to a specific deadline, because it is important to allocate budget ahead of time.

Lack of Top Management Support and Involvement:

Few projects have the chance of getting off the ground without the support of those high up in the corporate food chain.

Without executive support the project managers in the organization will find a difficulty in aligning business with their projects.

No one can deny that it's a problem when developers do not know who the "real" sponsors are, and keep progressing without sponsor involvement. For the best true sponsors need to be shown up and communicate with the team, follow the project step by step, hear good and bad news in "small pieces" rather than in "one chunk", this way you will avoid losing their support if any surprise comes on the way.

Non-sponsored projects are taken less seriously and may sometimes be viewed upon as someone's self-interest pipe dream. Without the backing of senior management to lend credibility to these projects, originators will have a difficult time recruiting employees to participate in development and testing, reason for that is because teams are usually made up of people from different departments who all have their own set of priorities and of course, they all have their own bosses, it's natural that those involved in any project will have tendency to keep the best interests of their own department in mind, and there's nothing wrong with that. In fact, that's why they are on the team, to represent the needs of their department. However the risk is in having a selfish person or group who may control project, ignoring requirements of others.

Insufficient Budget and Bad Resources Allocations:

Financial threats are the result of poor budget forecasting and tracking, lack of inter-department charge backs, and ineffective tracking of resource and cost allocations

Insufficient budget is still a major reason for missing goals and objectives of projects within the quality framework that is required. The concept for any project is like that project Y always need to be delivered tomorrow within X budget.

When we talk about budget we should be aware of what may happen if there is no enough funding, so a resource assessment should be made carefully by conducting complete and accurate financial analysis.

A resource assessment describes the people, skills, hardware, software, and network resources needed to complete a project. Resource assessment is sometimes a practical first step to making staffing decisions for a project.

The project manager is typically responsible to assess resource needs and to decide whether a formal, documented assessment is necessary.

What kinds of projects need a Resource Assessment?

Although every project undergoes some kind of resource assessment, they are frequently informal and undocumented. Large, complex projects, and those working with new technology, will benefit most from formal assessment of resource needs.

A resource assessment needs to consider and document each of the following items:

- Project Name.
- Staffing & Skills Inventory (What staff are already assigned to the project? What skills do they have?).
- Roles/Skills Needs (What roles and skills are needed that aren't covered by project staffing?).
- Staffing Needs (What staff is needed to address roles and/or skills not covered by staff already assigned to the project?).
- Training Needs (What training is needed to cover skill gaps?).
- Hardware & Network Needs (What hardware and network resources does the project require?).
- Software Needs (Does the project require any specialized software?).
- Support Needs (What kinds of support are needed from other C&C units to address needs for skills and/or roles not covered by project staffing?).

Poor Schedule Estimation, Unrealistic or Long Timescales:

Scheduling project work is an essential element of project management. A project schedule makes clear to all participants when work is expected to be completed. It also shows the time-related dependencies between different project tasks.

In a complex project, several schedules may be necessary, covering different levels of detail or different parts of the project

Bad time estimation causes project related problems. One common problem during the creation of the Work Breakdown Structure is assuming that the time on task equals duration. The time on task is the time the task will take to complete without interruptions, whereas duration is the time the task actually take to complete including interruptions. Using the time on task to estimate schedule is one of the common mistakes made by project managers**.**

Another common problem is using linear approximation when estimating schedule .For example, if you doubled the cows in a farm, you double your production of milk. The IT projects are beyond the scope of such approximations. Assume we have a large IT project using a team with a staff of one hundred people. Linear thinking would support the conclusion that increasing the people by 100 percent would decrease the schedule and increase the cost to approximately the same degree. In reality, doubling the staff produce a non-linear result.

In general, every project has a minimum achievable schedule.Many managers are well aware of the need for fast delivery, leading to other problem of unrealistic timescales. These are set without considering the volume of work that needs to be done to ensure delivery. As a result these projects are either delivered late or only have a fraction of the facilities that were asked for or they are bug-filled, because of that every project manager should consider volume of work, number of staff, number of working hours, and the duration of each task in parallel to avoid any kind of pressure, its is true that working under pressure can increase the quantity of results one receives, but, after a point, dramatically reduces the quality of those results. In fact pressure sometimes produces the opposite of its intended effect.

On the other hand if the project manager sets long time scales, the project may be useless as a result of changes in requirements.

Normally requirements change from time to time due to changes within the project users' environment. If the project objective is to serve certain society; it should be parallel to their requirements. The key recommendation is that the project time scales should be short, which means that larger projects should be split into separate projects.

Who does Project Scheduling?

Setting overall completion dates must be done by the project sponsor and stakeholders. The project manager assists in this by digesting information about scope, deliverables, and resources, and estimating times for completion of project tasks.

Once an overall schedule is set, the project manager is responsible for monitoring the progress of the project and revising the schedule if needed. This must be done in consultation with project team members who are doing the work. Working with team members to produce accurate time estimates is one of the high mysteries of the art of project management. The project manager must balance the needs for honesty and realism with appropriate motivation to keep the project on track despite inevitable surprises.

There will typically be give-and-take as a project proceeds among budget, features, and schedule. It is essential for the project manager to keep all participants informed as to current schedule status.

Time schedules should be reviewed to see if they are realistic and participants should express their reservations on it

Communication breakdowns, Failure to communicate and act as a team:

Projects sometimes fail because of improper communication between teamwork; in such cases they lack the ability to work as a cohesive unit and are in constant disagreement. The arguments and infighting cause everyone to move in opposite directions, lowered morale, and spawn an "us versus them".

Another common problem is the size of project team; there is a direct relationship between size of project team and difficulty in keeping all members of that team up to date on changes, progress, tools, and issues. Such problems are common on large projects, especially if people are working at different sites. In many troubled projects there isn't one person who has an overview of the whole project. Each project member needs to know how his or her one piece fits into the entire architecture.

The key recommendation here is that rarely to form a team of more than five members, instead opting to form multiple teams working on individual objectives. Furthermore, each of these smaller teams has a manager, who is himself part of a management team. In extreme cases multiple management teams exist and an executive team is formed. The focus of each team is strictly enforced and rigorous in definition.

In general Communications problems can be avoided by adopting a communication plan in the planning phase.

Communication plan identifies people with an interest in the project (stakeholders), communication needs, and methods of communication. Communication planning helps to ensure that everyone who needs to be informed about project activities and results gets the needed information.

The project manager is responsible to identify communication needs and to decide whether a formal communication plan is needed.

Although every project undergoes some kind of communication planning, it is frequently informal - determining who needs to attend which meetings, receive which reports, etc. Projects of long duration will benefit from formal planning because the project stakeholders are likely to change over time. Projects that affect a large number of people or organizations may also benefit from formal planning to ensure full identification of both stakeholders and of communication needs.

A communication plan needs to consider and document each of the following items:

- Project Name
- List of Stakeholders (Who has interest in the project? See the project definition for an initial list of stakeholders. Be sure to include both business and technical stakeholders.)

- Information Needs (What kinds of information about the project are of interest? Consider need to communicate plans, status and progress reports, changes, major events, availability of prototypes and demonstrations, etc.)
- Communication Methods (What information will be communicated to what groups in what ways? Common methods include reporting and documentation, email, meetings, and web sites.)

Staffing (Inappropriate Skills, Lack of Number of staff):

Staffing is one of the most critical elements of a project's success. Without staff, there is no project. Once you have defined the project and are clear about at least some of the project's initial tasks, you can define your staffing needs. It's important to know the type of staff that the project needs, e.g. database administrator, one or more programmers, and technical writer. Once the type of staff has been defined, you need to get individuals assigned to your project. The best places to go for staffing resources are the project's sponsor and stakeholders.

You should be prepared to answer the following questions that might come up when you ask for staffing resources:

- What percentage of their time will you need?
- How long will you need this person?
- What are the benefits of this particular person working on the project?
- How do the skills needed and this person's skills match up?
- How many members do you need to share workload?

Most IT projects require a diverse range of skills; the project must have the right people to do the right job.

For example, programmers need to have experience in the technology before counting on them, so they should be selected wisely. Furthermore, managers can perform poorly if they lead projects that do not match their expertise. The project manager should have enough experience and knowledge, preferable passed similar projects before, so that same mistakes will not be repeated. Projects which deal with high technology need managers with solid technical skills. In such projects, authority must reside with people who understand the implications of specific technical risks.

However, the best technologists are not necessarily always poised to be the best managers. The skill set for management and programming are disjoint. The larger the project, the more need there is for people with excellent planning, oversight, organization, and communications skills; all excellent technologists do not necessarily have these abilities.

The solution to skill-driven challenges is easy to define but difficult and expensive to accomplish: Attract and retain the most highly skilled and productive people, that’s mean when making up teamwork to select higher-paid people with the right specialized skills is worth far more per dollar to an organization than a group of lower-cost people who need weeks or months of fumbling through a new process or technology before they can start being productive. In a straightforward phrase "You get what you pay for".

Poor Testing:

The developers will do a great deal of testing during development but eventually users must run acceptance tests to see if the project meets their business requirements and this stage should be before the project implementation. In fact skip the testing phase because the project is way behind schedule will lead to a downright failure.

However testing often fails to catch many faults before a project goes live because:

- Poor requirements which cannot be tested.
- Poorly or non planned tests meaning that the project is not methodically checked.
- Inadequately trained users who do not know what the purpose of testing is.
- Inadequate time to perform tests as the project is late.

Users, in order to build their confidence with a project, and to utilize their experience of the business, should do the acceptance testing. To do so they need good testable requirements, well designed and planned tests, be adequately trained, and have sufficient time to achieve the testing objectives.

Technology Illiteracy:

It's related to the failure in aligning business objectives with IT and its processes; this usually occurs when the company's internal controls have material weaknesses or when it is in non-compliance with various processes, because of that each project should have Internal or external auditors who have really an obligation to publicly report facts.

Sometimes adopting new technology may lead to a failure, even though it is successfully tested, implementing it for the first time in the project is in itself a risk. Will the team use it in the right way? Will they have enough practice while they don’t have expertise? Will it satisfy the project requirements?

Hidden Costs of going "lean and mean":

Any failure will be viewed as a direct result of underperformance, even though underperformance is not often a significant factor in the failure of most projects. Instead, failed projects often have goals that were inherently unattainable, poor staff, etc.

Late Failure Warning Signals:

The early project milestones involve diagrams, designs, and other documents that do not involve working code, these and other project milestones then go by or less on schedule, and testing may start more or less on time, so that errors which discovered days before the deadline of the project will cause the project not to be completed even close to its deadline.

References

Glaser, J (2004) **Management's role in IT project failures** Healthcare Financial Management, October.

Grossman, Ira (2003) **Why so many IT project fail, and how to find success** Financial Executive, Volume 19, Issue 3, page 28.

Humphrey, W (2005) **Why Big Software Project Fail: The 12 Key Questions** The Journal of Defense Software Engineering, March Issue.

Armour, P (2005) **Project Portfolios: Organizational Management of Risk** Communications of the ACM, Volume 48, Issue 3, page 17.

James P. Lewis **Fundamentals of Project Management**, 3rd edition.

James P. Lewis **Team-Based Project Management** in Back Matter (1), Back Matter (2), and Back Flap

Betts, M (2003) **Why IT Project Fail** [Online journal] Computerworld, Volume 37, Issue 34, Page 44. Available from Academic Search Premier at http://www.ebscohost.com [Accessed July 21, 2005].

Jenster, P and Hussey, D (2005) **Create a common culture between IT and business people to reduce project failures** Computer Weekly, March 22.

Coley consulting (2001-2005), **why projects fail**, Available at http://www.coleyconsulting.co.uk/sitemap.htm

Sue young (2003) CEO of ANDA Consulting, **Computer world IT management**, August 25.

Simon Wallace (2004) the ePMbook, Available at

www.epmbook.com

Ephraim Schwartz (2004) online research **IT Myth 5: Most IT projects fail**, August 13.

Pual Chin (2003) online research **Cold Case File: Why Projects Fail**, may 6.