

Melillo Consulting, Inc.  
285 Davidson Avenue, Suite 202  
Somerset, New Jersey 08873  
908-563-8460  
<http://www.mjm.com>

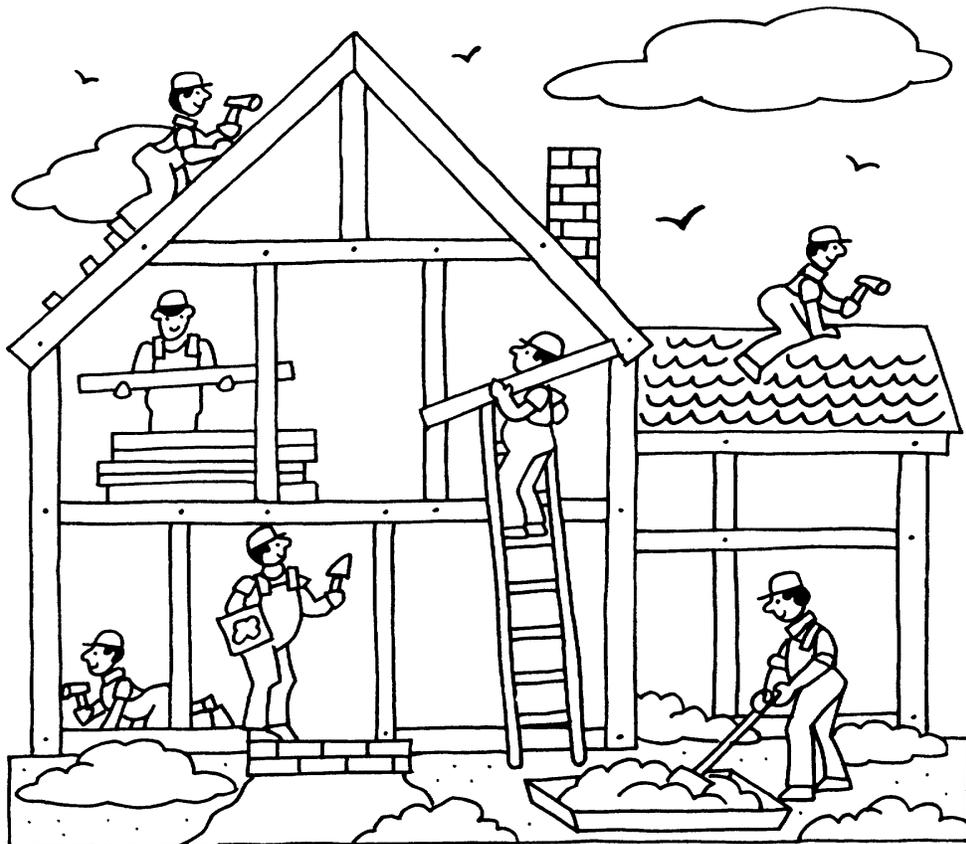
Mike Battistella  
Network & Systems Management  
Practice Manager  
[mbatts@mjm.com](mailto:mbatts@mjm.com)

---

# Blue Print for Success, or Recipe for Failure

---

## How to Handle Large Network and System Management Projects



---

# Blue Print for Success, or Recipe for Failure

## How to Handle Large Network & System Management Projects

### Approaching a Network or Systems Management Project

#### Introduction

This paper deals with an often forgotten approach to handling projects. This approach could easily be applied to any type of project, but its focus is on large-scale Network and Systems Management projects, primarily because that is what I do for a living.

Although there are many books, papers, and seminars on proper approaches to project management, I will not attempt to compete with them, nor will I try to sell this paper as a project management methodology, technically or otherwise. This paper deals with project approaches from the strictly practical, common sense point of view.

Although I have a Masters degree in Engineering Management, I have only limited formal training in “Project Management” styles, philosophies, and strategies. What I can offer are my first hand experiences of what has been successful, and where I have felt the most pain.

#### What Have You Experienced?

Most people in technical fields today have been involved in a project of one form or another. If you were to overhear conversation about that particular project, you would hear some very consistent complaints about how the project was handled, who led it, and questions as to why the company even proceeded down that particular path. What is most amazing is that the typical problems faced during a project are just that: they’re *typical*. With all of these known issues, why is it that we just can’t seem to get it right? Once a project manager, and or project technical lead endure a very painful project, (i.e. one that was not run properly,) the common mistakes are rarely made again. The assumption here is that they have some authority to make sure they don’t happen again. The problem is that the issues don’t change, but the individuals leading the project typically do. Not uncommonly, the management asking for this particular project may not understand the technology involved, or never experienced the “pain.”

Most of the common complaints I’ve heard concerning large projects which have been unsuccessful are:

#### Management Backing

Without proper management backing, projects start behind the eight ball. Management appears to be in line with the project goals and outcome, but the actions they take don’t seem to align with what you’re trying to accomplish. Without management’s full commitment to the success of any large project, there will always be the concessions and tradeoff’s between doing it right and doing it on a shoe string; between being successful and being adequate.

#### Workable Plan

Many projects start without a plan. How can you build a house without a plan? This is the most common issue facing the people trying to get the project under their belts. A plan is what gets everyone on the project

going in the same direction. Without a plan, each member of the project team is pulling in his, or her, own direction. The project gets off to a slow start and stays that way: *slow*.

### **Proper Resources**

Let's face it, how many times do you get put on a project where everyone is top notch and knows exactly what to do? It will never happen. At times, individuals tied to the success of a project do not have the skill set, or the knowledge level to pull their piece of the implementation. Before you know it, there's tension from the other team members, and discouragement. Down the failure path again.

### **Proper Funding**

Without proper funding, how can anything be successful? Ask the U.S. Government. You can't be successful on a \$39 million dollar project with only \$3.5 million. It just doesn't work that way, and nothing you or I can do will change that. Money is part of the picture, but it has to be the right shade of green or the project dies an agonizing death.

### **Overall Objective**

Why are we doing this? What will this accomplish? How will this improve the way I do my job? How does this fit into the overall goals of the company? How does this affect *me*?

If these questions can't be answered, how can we expect the full commitment of those involved in making the project successful? The project must have an objective, and that objective must be clearly communicated to those who are part of the project team as well as those affected by the implementation of the project technology, or the process that will be put in place because of the project outcome.

### **Measurement of Success**

What will be the measurement of success? This question is asked at the management level, the consultant and project management level, and at the implementation level. All individuals involved in a project, especially a highly visible project want to know how they'll be measured. The project contributors want to know how the work they've completed will be viewed within the organization and what water marks will determine the success or failure of their efforts. Too many projects are open-ended, without defined goals, expectations, or outcomes.

### **How Does it all Fit?**

It's important to define how the various aspects of a project fit together in the overall view of the project. This is important within the project for obvious reasons. More importantly, it's important to understand how the overall project scope fits into the scheme of the organization. This helps the project team clearly understand the direction the project needs to go in, and how to best use existing company policies and procedures when making decisions that will have an effect on the broader organization

### **Project Approach**

There are many strategies, project philosophies, and methodologies that projects can follow -- none of which will be discussed here other than to say there must be a project strategy, philosophy, and methodology. Without these things, projects can experience chaos. Without a project methodology, how is *change* management handled? How are *issues* resolved? How is *conflict* managed? The importance of a Project Approach can not be over emphasized.

## Project Concepts

### Project Methodology

What kind of methodology should I follow? There are probably hundreds of them out there, and since this is not a paper on Project Management per se, I won't put a name on what I'll present. What I will show is what has worked best for me.

I like to follow the Top Down approach to larger projects. This involves design work to take place long before anyone is thinking about laying down bits on the system, or ordering hardware or software. We'll talk more about Top Down vs. Bottom Up shortly, but for now let me focus on the Project Flow.

Here's the plan I like best:

#### **Requirements Definition:**

This is the most crucial piece of the project. A design can't take place, and technology be implemented successfully unless the project team is aware of what it is that needs to happen. The Requirements Definition identifies what it is that is being addressed by the project and its technology. Within the Network and Systems Management arena, it also identifies what elements are to be managed, referring to Network and Systems Management of course. I heard it said that the typical customer only knows 10% of what needs to be managed. I don't know that I agree with that exact number, but I know the actual number is around or below 25%. This phase of the project is where all of the managed elements are identified, and a strategy is set in place for managing these elements. I like to use the OSI Model as my frame of reference. By going through each layer of the OSI Model, starting at Layer 7, the Application Layer, and working my way down to Layer 1, the Physical Layer, most managed elements can be identified. Using this information, I like to create a matrix of what will be managed, and any corrective action, or external notification defined.

Once the Requirements Definition is complete, all parties involved including management, work to agree on what needs to be managed, the priorities, and actions. A number of benefits have just been achieved. The scope of the project has now clearly been defined. Everyone now is aware of how large, or small, the project really is. Resources can now be identified because the management areas have been clearly defined, and the measurement of success can be calculated.

#### **Logical Design:**

This is the high level architecture that will lay the groundwork for the rest of the project. This is where the overall network and system management approach and the correct components that will make up the management solution will be laid out. In this phase, I like to identify the management tools I'd like to use and why. I like to size the system management systems, identify software components and requirements, and create software compatibility matrixes so there is no question that the plan is well thought out. This way the project team and others involved in the project know that the identified software products will play together.

The output, (or deliverable, for you consultant types,) of this phase of the project is a Logical Design Document laying out the required information. The Logical Design is presented at a formal design review. I like to give the Logical Design to individuals who are critical to the successful completion of the project for review prior to the design review. That way they have time to absorb the strategy we have chosen, and absorb what the project team will be presenting.

The desired result of the Logical Design review is that the approach, strategy and selected components will be formally approved, and the project can move forward. The formal acceptance of the Logical Design is the go ahead to move into the Physical Design phase of the project.

The Logical Design answers the question, “*What are we going to do?*”

### **Physical Design:**

The Physical Design phase gets into the details of the project. In the Physical Design phase of the project, we want to answer the question, “*How are we going to do what we said we were going to do?*”

In the Physical Design, we break down the Logical into very detailed components. Here the details of what is being managed and how, are addressed. We’re down into actual polling intervals, actual corrective actions or commands are spelled out. Network bandwidths are calculated, and configuration file setups are listed. The Physical Design gets into what, exactly, is going to take place.

The Physical Design is then again followed by a design review. By presenting the design, the project team achieves two goals. The first is that the design is communicated outside the project team. Secondly, once the Physical Design is approved, there is the go ahead to continue with the project. At this time, a detailed Implementation Plan and Project Schedule are put in place. Also the implementation team is then selected and can begin to be integrated into the project team.

### **Implementation Plan with a Project Schedule:**

The Implementation Plan takes the Physical Design to a level that implementation people can deliver. The Implementation Plan is written in such a way that a document can be handed off to an implementor, and that component of the project could be carried out with very little direction from the designer. In the consulting arena, architect types cost a project lots of money, where the implement types are usually less costly. By investing time and money in the Implementation Plan, the high priced architect types can be released, or their time commitments can be lessened, in favor of less costly team members.

This is where the Implementation Project Schedule is laid out in detail as well. I’m always amazed when I see a Gantt Chart for a project that shows details of the implementation when the design hasn’t even been done yet. I know, I know, it’s just an estimate, but when I work with a Gantt Chart, there’s always some project manager pointing down at some task that I had no input in coming up with its time duration, and wanting to know why I don’t have it done yet. It’s usually because I’ve been doing the fifty other things that are not on the Gantt Chart that should have been. Did I mention that an accurate Gantt Chart is a requirement to a successful project implementation? The project leads and the project manager must work together, and agree that the Gantt Chart is accurate, or there will be a lot of tension among the project team for obvious reasons.

### **Implementation:**

There's not much to say here other than this is where we actually do what we said we would do. This is the most exciting for the sponsors of the project. Up until now, all they've seen are volumes of paper. Resist the temptation to begin implementing something while the design is happening. I've often been put into situations where I've been told to implement something to keep management happy while the design is still in progress. The problem here is that resources are tied up for something that will need to be trashed anyway. Usually, whatever has been implemented must be redone. Today the way I handle it is to ask to see the project plan for this portion of the implementation, and then redoing this portion of the setup and configuration. If it hasn't been accounted for in the Gantt Chart it will certainly have a negative cost and time impact on the project; probably not a good idea.

### **Test:**

"If you design it, design a test for it." That's what I tell my design team. During the Physical Design phase of the project is the best time to define the test for that portion of the project. If it doesn't happen there, it will be less than thorough. What usually happens when you go back to define a test is it's hard to remember what was going through your head at 10:00 AM that morning when you added this particular piece to the design. You glance over it with some superficial test to satisfy the requirement instead of defining an accurate test prior to going live.

### **Go Live:**

You've defined what needs to be done; you defined how you're going to do it, you've detailed every step along the way, and then you've done it, and tested its accuracy. Time to go live.

### **Bottom Up Vs. Top Down**

These approaches are appropriate under different circumstances. The Bottom Up approach seems to work well when a pilot is the goal of the project, or proof of concept is of concern. The goal is to get involved in new technology and see how it fits into the organization. Large dollar amounts are not at issue and if the end result is a point solution, the Bottom Up approach works well.

Unfortunately, many times the Bottom Up approach is used when it is entirely inappropriate. The Bottom Up approach does not typically involve a design, or take into account the rest of the organization. If at any time, the entire organization could be taken into account, the Bottom Up approach usually involves scrapping what has already been done, so that scaling can be taken into account. The Bottom Up approach should really be reserved for small projects, point solutions, and relatively few Network & System Management (NSM) areas are involved.

When the project is strategic to the company, needs to scale to the organization or enterprise, and involves many NSM areas, the Top Down approach must be followed. The Top Down approach involves viewing the project with the end in mind. The project leads must know where it is they are trying to go, and then put a plan in place to get there. Once this is laid out, the project should then be broken down into sub projects, and then sub project components. Each component, and sub project, must be tied together by the string of communication.

Once this breakdown occurs, a high level Logical Design, or Architecture can be worked on. One of the most common problems I've seen is that hardware and software are being ordered based on the project being sold, based on proposals, or just on "best guess." This has always amazed me. How can you order what is required to successfully complete the project when the Logical Design has not been completed. Put off ordering what is needed until at least the Logical Design has been completed and there is overall agreement of what the project really is, and what is required to implement the design.

I was on a project that was very specific. SAP was to go live and system management needed to be in place by the go live date. There were a small number of systems, and a very specific application. The number of Network and System Management (NSM) areas was limited to four. This is the typical bottom up approach.

We were able to successfully meet expectations, and this relatively small project was actually quite successful. Now here's where the curve ball comes in. Now this company's headquarters saw the benefits of what we were doing, and what had been accomplished. Now they wanted to go global with what we had done locally. A team from headquarters was sent out to evaluate what we had done, how we had done it, and how it could be done on the enterprise level. Only about fifty to sixty percent of our work could be reused because of scalability issues. If the project direction was that this would eventually go to the enterprise, and the organization was willing to fund the design effort in a top down approach, much more, if not all, would be reusable and the design work would have taken scalability into account. In case there is any question, let me state it very clearly; design and implementation differ tremendously when going to the enterprise.

### **Have the Right Tools**

Too often we try to take on projects without the right tools. Then by the time we've suffered the pain long enough, and made enough of a nuisance of ourselves, we finally get what we need. Time is the valuable resource lost by not having the right tools in place. Part of the project planning should be to get the right tools in place prior to, or at the start of, the project.

#### **Hardware:**

The right hardware would consist of whatever servers, workstations, PC's, printers, faxes, network connection devices (i.e. bridges, hubs, routers) necessary to complete the task at hand. Each member of the team should have the equipment that will help that individual be successful. I was once on a project where a good portion of the technical design team did not have laptops or PC's to work on. It took close to four months before the Program Office was able to commandeer laptops for each design engineer. Try to imagine the lost productivity. Better yet, try to imagine the improved productivity of having the laptops in their hands from the start.

On this same project, it took two months to get a network and a network server in place. With a project team of twelve (on the NSM side) and no connectivity, file and information sharing was relegated to sneaker net and floppies. Again the productivity was greatly diminished because the right tools were not in place. Unfortunately, this productivity loss is hard to measure, and is often ignored.

#### **Software:**

Having the right software in place could mean a variety of things. What I'm referring to here is a working Network and/or System Manager with all the necessary software that is needed for the project. This enables the design team to work a design and then do a quick verification when necessary, or hop on the system and try something out real quick. We'll talk more about a project test lab in a little bit.

Also, you need to have the documentation software in place. Not only do you need the product itself, but consistent revisions among the team members. Standardize on the desktop publishing package, the presentation tool, and spreadsheet tool. Make sure everyone has access to the engineering diagramming tool. Keep around the latest copy of virus scanning software.

Make sure everyone has the same, or close to the same PC operating system. I was once involved in a project where some of the team was running Windows 3.1, others running Windows 95, and a couple of us running NT. We all had a common project file server, and the Windows 3.1 guys were constantly complaining about the rest of us using long file names. Imagine trying to setup a project library with short file names. What

about the guys using versions of Power Point that the Windows 3.1 guys couldn't even read? It's very important to standardize on the tools being used.

### **Communication**

Communication is literally the single most important aspect of a successful project, especially when talking about verbal communication among team members, clients, staff, and the outside community that may, or may not have a stake in the successful completion of your project.

The communications I'm referring to in this section, however, are the tools used for effective communication. There are many communication processes, but I currently focusing on the tools.

Communication tools include the effective use of voice mail. Keep it short, keep it simple. If more detail is needed, or the communication is critical, use email. Email is the single most effective communication tool because you are able to articulate in detail the message you want to get across. It is also an effective documentation tool. Keep a designated mailbox for the project and dump all electronic communication into it for reference. Believe me, you'll never know if it's needed. And, if it is, you'll want to have saved it.

Everyone on the project team should also have a pager for those emergencies when immediate communication is necessary. Anyone who knows me personally, and reads this will be shocked! I never was, and am not, a fan of pagers. I was famous for not giving my pager number to anyone but my wife. I started wearing it when she was pregnant with my first child and took pride in not handing it out to anyone. I even went through two managers without passing it on to them. Sorry guys if you're reading this. After having spent some time on a couple of difficult projects it became very clear the importance of being able to get in contact with someone quickly when the circumstances dictated it. I also learned the importance of having my team be able to get me quickly when something critical arises. I'll still scold one of my team members if I'm paged unnecessarily, or when voicemail or email will do, but I've changed my opinion on the importance of quick communication. I also recommend text pagers so that a message comes with the page so that some of the unknown is taken out of a typical beep, beep, beep.

Another great tool for communication is a Project Library. I typically setup a share in the Project Office that everyone has access to. The whole file system structure is defined so everyone knows where to go to get information. I also have a branch dedicated to subprojects within the overall project so that documentation created tied to a specific project area is also easy to find. It goes without saying that the project library must be backed up regularly. The project library also includes a spreadsheet that lists exactly what documents are in the project library, when they were last modified, who owns the document, and who last modified the document. This may seem a bit painful, but it is a great time saver later in the project when you've just been asked to bring a particular document to your 10:00 AM status meeting. Oh, by the way, it's 9:45 AM.

I'm also a great fan of the Project Folder. Any customer communication, decisions that have been made and why, and documentation as it becomes official or formal goes into the project folder. Again, over time it proves its worth. The project folder can be used as a legal document if there were ever a serious problem with the project, and it can also be used as a project deliverable. Make sure there are two copies, one for the Program Office, and one that would be handed off to the customer upon completion of the project.

### **Environment:**

Without the proper working environment, productivity comes close to a stand still. I was once on a project where my team was sitting in the cafeteria trying to get something accomplished. Part of my team came in from out of town and insisted on working in their hotel rooms so they could concentrate. This doesn't seem to fit the mold of effective verbal communication among team members. Each individual on the team should have his or her own desk with a phone, voice mail, email access, and a PC or laptop. Many times I go into a company with a team of people, and the last thing they want to do is give up space to an outside team when

they barely have room for themselves. Understandable, but unacceptable. To have a project succeed, the project team needs a place they can call home.

Another aspect of the project team environment is a meeting room, for obvious reasons, and a war room. This room is for brainstorming sessions, laying out diagrams on whiteboards and flip charts for the rest of the team to see what and how, and plain old loud conversations and debates. I guess that's why it's called a "war" room. This too is a vital part of the project environment that is often forgotten about or just impossible to get.

I've mentioned the Program Office quite a few times by this point. I guess I should explain what I mean by it, since I've alluded to it quite a few times. The Program Office may, or may not be an area where the project team is situated. It is where the status meetings take place and the project documentation is stored. More importantly, on very large projects, the Program Office is where the senior staff of the project team spends most of their days and, if I may add, nights.

The Project Office is the nerve center of the Project. The Project Office should be a physical office where all the key project personnel for the project work together. It is only a concept, albeit a powerful one. From the Project Office the Program Managers, key HP Project Managers and Project Leaders make all important decisions guiding the Project through the complexities and uncertainties of the custom project environment.

There are many people required to make a project successful, but the Project Office is comprised of the key personnel responsible for leading and controlling the Project. The success of the Project lies squarely on their shoulders.

### **Manage What?**

There are so many items to manage in any project. Far too many to go into here, so I tried to hit some of the top hitters.

### **Problem or Issue Management**

What is the method, or process for handling issues as they arise? Is there a program office in place? Is there staff meetings scheduled with "Issues" listed as one of the agenda items? Whether this is a project staffed completely by consultants, a joint consultant, internal staff project, or a totally internally staffed project, issues are going to arise. There must be an agreed upon process, that all team members know about, for managing issues or problems as they arise (i.e. in a timely manner) or these problems, or issues can quickly become show stoppers.

### **Change Management**

How will change be managed? This is especially important if this is a fixed priced engagement being staffed by an outside consulting firm. The project is scoped and priced based on a set of deliverables. I can guarantee there will be changes to the originally defined scope. Is it fair to the delivery team to just expect them to "throw in" additional activities just because you, the customer, are spending so much money into the project already? Well, that certainly depends on what the addition, or change is. Certainly, if the impact and time investment is small, consultants tend to be pretty flexible. But if there is an impact on the profitability of the project, there is a significant time investment, or a definite resource impact, it is not unreasonable for there to be a cost associated with the change.

The key here is having a defined and agreed upon process for handling changes in project scope.

## **Schedule Management**

Who's managing the project schedule anyway? Hopefully the project manager has this one covered. Unless this is a small project, say \$250K or less, don't put your technical leads in the position of having to be project managers on top of the already large task they have. I have an especially strong opinion about this, and you can only begin to imagine why.

The project schedule needs to be tracked for a number of reasons not least of which is identifying the critical path. If the critical path is not identified and clearly communicated, it won't matter how quickly other tasks are completed, the project will not complete on time. You need to be very careful with engineer types, of which I am one; we like to work on the tasks that interest and challenge us, not necessarily ones that need to get done. The project schedule helps us understand what needs to be done when and what impact it will have if it is not completed on time.

**Note to Project Managers:** Do not beat up your team with the project schedule. Use it to motivate the completion of tasks in a timely fashion. Once it's used as a bat, you lose the value of the tool.

Another aspect of schedules is an up to date calendar from each project team member. Oh, I know they hate to provide it, but it's essential for tracking who is coming and going, and for planning meetings: status meetings, brainstorming and design sessions, and regular team get-togethers. One of my favorite places to congregate for such meetings has become a local brewpub. Ok, I'll save that for another paper. I hope my wife isn't reading this.

## **Staff Management**

As the Network and Systems Management projects I've been involved in have gotten larger, one change is very apparent. One single vendor does not typically have the staff to work the project entirely on their own. The project teams are comprised of vendor staff, outstanding and talented channel partners like that Melillo Consulting group, independent consultants, and client team members. The total team has to be coordinated and managed in a, sometimes, delicate manner. It takes a skilled project manager to manage such a staff. The goal is to very quickly, and early on in the project, break down the walls of separation. The team must function as a single unit with a common goal.

I was most impressed on a recent project where my team was staffed with HP TC's, Channel Partner consultants, and a couple of independents. It was a team makeup like I'd never seen. They worked as a single team. An outsider could not tell where one portion of the team left off and the next began. That's essential for a project to go smoothly. All right, enough bragging.

## **Communication Management**

How are you communicating? Are you over-communicating? It's a tough balance. It has been said, refer to any book on project management, that poor communication is the single most important reason projects fail. Then why, I ask, is it the one thing we do most poorly? Mainly, because we don't know how to communicate. We setup status meetings, ask for status reports, want emails or voice mails daily, any thing to add additional work. What ever happened to just sitting down and having an intelligent conversation? That's the main reason I like having brainstorming session. I like to get a group of design engineers in a room, define the problem, and let the ideas start flowing. Yes, it can get testy at times, but the results are, at times, astonishing. Some of the most creative ideas have come through disagreement. Some of the team I work most closely with, most often will attest to that.

Status meetings are necessary, status reports are needed without question, meetings with the customer are essential, but don't forget to get together with the team and brainstorm. I find some of the best discussions and ideas happen over food. Don't forget to take the team out now and then; the investment is well worth it.

### **Action Item Management**

Network and System Management projects today can be very complex to manage. To make things manageable, many tasks are delegated to managers and engineers on the Project. To ensure that nothing slips through the cracks, the Project Office should develop and manage an action item matrix (sometimes called an Action Item Register or Log). This log can be maintained through a joint administrator. The purpose of this matrix is to enter activities which must be done jointly, and track them to completion. Items to be included in an action item matrix include:

- Action Item Number
- Date opened
- Date closed
- Status (e.g. open, closed, canceled, on-going, on-hold)
- Owner - who is responsible to manage the action item to completion
- Description - briefly describe what needs to be done
- Notes - free space to post status update and comments

### **Risk Management**

Project risk management includes the processes concerned with the identifying, analyzing, and responding to uncertainty. Risk management is comprised of the following major processes. For every issue that is escalated above the Project Manager level, resources, timelines, and schedules need to be adjusted according to the risk assigned.

### **Risk Identification**

Risk identification consists of determining what sources of risk and which risk events may reasonably be expected to affect the Project. Risk identification is not a one-time event; it should be considered as a primary responsibility of all project team members and performed on a regular basis. The primary inputs to risk identification are:

- Product Description (The nature of the product may be new or substantially modified)
- Scope Statement (Cost & schedule aggressive)
- Observations contained in the Assumptions and Constraints portion of the Project Plan (Factor's that limit the Project team's options or control)
- Historical Results (Actual results of previous projects)
- Project Team Knowledge
- Project Resource Plan (Limited or critical skills or access to resources)
- Work Breakdown Structure (Clear and detailed)

The primary outputs of the risk identification are:

- Sources of risk are categories of risk events (e.g., uncertain estimates, team turnover)
- Probability that the risk even could occur
- Shareholder risk tolerances
- Risk symptoms (Poor morale, schedule delays)

## **Risk Quantification**

Risk quantification is primarily concerned with determining which risk events warrant response. Some forms of quantifying risk are used including schedule simulations, decision trees, statistical sums, etc. The major outputs from risk quantification is:

- A list of opportunities that should be pursued and threats that require mitigation
- A list of opportunities to ignore or threats to accept

## **Project Blue Print**

An excellent idea is to have a Blue Print of the project. This would typically be defined during the Requirements Definition and is a great vehicle into the Logical Design. This should be a very high level definition of what the project is, what the goals of the project are, and diagrams depicting communications between various components of the project. There is no need for detailed drawings: that will come through the Logical and Physical Designs. What's required here is nothing more than block diagrams showing what the managed elements are, and the communication links between all the managed elements and components of the Management station, or stations.

The value of this cannot be overstated. It really gives the project team, and those outside the project team, a very clear picture of what the project is. It becomes a communication mechanism to those within and outside the project team. It also becomes a communication tool for the individual team members. If I'm responsible for one NSM area, for example, and my box has lines touching a box that someone else is responsible for, I know that I had better have a thorough conversation with you about what that line represents.

The term "Blue Print" may be a bit misleading. When I think of an actual blue print, a very detailed drawing comes to mind. I think of the project blue print as more of a floor plan that will be used to get to that very detailed drawing. It is a time investment, but really adds tremendous value in getting the project completed.

## **Test Lab**

I have often been called upon to do a project where the only box available to work on is the system that will be the System Manager, or currently is the System Manager. I'm not real fond of doing development work on a production system. During the early stages of a new project it only makes sense to make use of a system that may be identified as the future manager, but once you get to a certain point in a project that system has to be treated very respectfully. I'm not very interested in having a wide variety of hands on that box to "try something."

The Test Lab is where all the "trying" should be taken place. Any project considered large in nature should have a test environment setup. Initially, it can be used to pilot a portion of the larger project, or be used as a proof of concept. Then the test lab can be used to do development work over the duration of the project.

The unfortunate thing about a test lab, and when it is so often hard to come by, is the dollars associated with it. This is a necessary evil that I recommend any large project consist of. Imagine what would happen if you were developing some custom code, or tying a new piece of software that trashed a production system, or the main network or system manager. I can tell you, the screaming is deafening.

## **Training**

Why is training always forgotten about, or an after thought? Companies spend hundreds of thousands of dollars on technology, but very little on the training of the individuals that will be using that technology. Maybe it's my past life as a technical instructor, but training seems to me to be the one sure way of ensuring a successful handoff from project team to the internal staff that will be using the technology on a day to day basis.

Training comes in many different packages. There's formal classroom training. There's informal training. And, there's the one on one that happens when one individual works side by side with one of the team members. Each has its place, and any project should include each. To leave off any one of these forms of training has a negative impact on the overall success and perception of the project.

My preference is to have members of the project team involved as instructors during the training. It gives them a real sense of ownership in making the client successful in taking over and understanding the technology involved in the project. It also gives the team some additional visibility that they may not otherwise get during a project. It gives each team member additional credibility that is at times difficult to get. It gives them a chance to "strut their stuff" and really have an opportunity to show why things are the way they are. And lastly, it creates a great environment of communication. Training benefits both sides of the project team.

## **Recipe for Failure**

From projects we've been involved in, what has caused them to either fail, or not be as successful as we would have liked them to be.

- 1)** Jumping right in with both feet
- 2)** Not working by a detailed design
- 3)** Proceeding without upper level management buy in
- 4)** Proceeding without the support of the organization
- 5)** Biting off more than you can chew
- 6)** What about Training?
- 7)** All you have to do is .....

## **Blue Print for Success**

From experience, what can, and should be done to make a project successful.

- 1)** Planning, Planning, Planning
- 2)** A Working Blue Print or Design
- 3)** Backing from Upper Level Management
- 4)** Participation, and buy in from the organization
- 5)** Project Methodology
- 6)** Formal and Informal Communication
- 7)** Clearly Defined Goals
- 8)** Have the Right Tools in Place