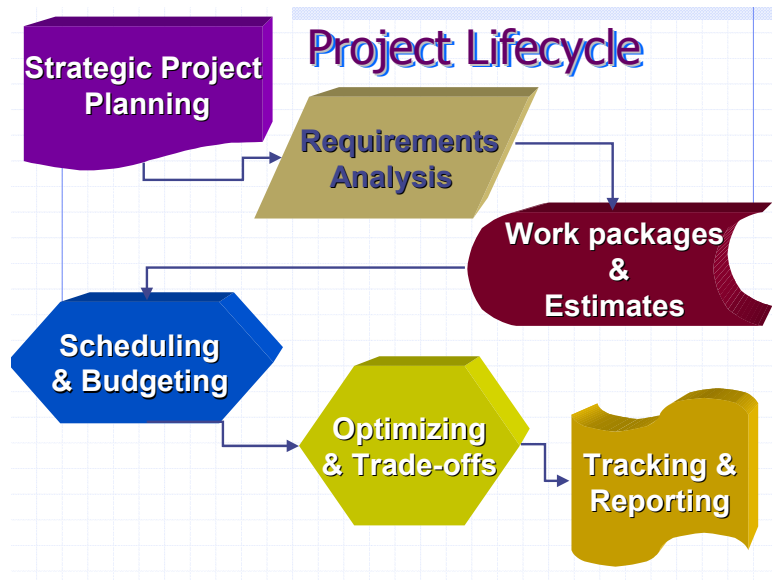


# Managing Information

# Technology Projects



By Dick Billows

Copyright © 2002 by Richard A. Billows. All rights reserved.

Published by The Hampton Group, Inc.

For credit card orders: The Hampton Group, Inc. <http://www.4pm.com>

3547 South Ivanhoe Street

Denver, Colorado 80237-1122

800-942-4323

303 756-4247

Other books by The Hampton Group, Inc.

Essentials of Project Management

Managing Complex Projects

Project Manager's Desk Reference

People, Budgets & Software

Microsoft is a registered trademark and Project and Windows are trademarks of Microsoft Corporation.

Screen shots reprinted with permission from Microsoft Corporation.

All other product names and services identified throughout this book are trademarks or registered trademarks of their respective companies. They are used throughout this book in editorial fashion only and for the benefit of such companies. No such uses, or the use of any trade name, is intended to convey endorsement or other affiliation with the book.

All rights reserved. The text of this publication, or any parts thereof, may not be reproduced in any manner whatsoever without written permission from the publisher.

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

Library of Congress Catalog Card Number: 94-96302

ISBN: 0-9679761-2-X

**Table of Contents**

**1 - IT Project Lessons ..... 10**

BAD SURPRISES WHEN IT'S TOO LATE..... 11

CHANGES, CHANGES AND MORE CHANGES ..... 12

PROJECT TEAM WANDERING IN THE WILDERNESS..... 13

PLANNING, MISSION STATEMENT MUSH..... 14

ACHIEVEMENT-DRIVEN PROJECT MANAGEMENT..... 14

    MEASURE OF SUCCESS(MOS™) & ACHIEVEMENT NETWORK (HLA™)..... 16

    TRADE-OFFS AND SCOPE CONTROL..... 16

**2 - Strategic Project Planning ..... 19**

WHY STRATEGIC PLANNING IS SKIPPED ..... 19

THE ERP PROJECT ..... 22

    OUR PROJECT AT RIDGEWAY, INC..... 22

    THE ERP STEERING COMMITTEE: INVITATION TO THE ACTIVITY TRAP..... 24

AVOIDING THE ACTIVITY TRAP..... 25

STRATEGIC PLANNING AT THE RIGHT LEVEL..... 27

GAINING ACCESS ..... 27

    RIDGEWAY CASE: GAINING ACCESS ..... 28

    TRYING ANOTHER APPROACH ..... 29

    RIDGEWAY CASE - WORKING OUR WAY UP THE HIERARCHY ..... 29

THE STRATEGIC PLANNING SESSION ..... 31

RIDGEWAY CASE: THE STRATEGIC PLANNING..... 32

RECAP OF THE STRATEGIC PLANNING SESSION SO FAR ..... 38

    BACK TO THE SESSION: WEBSITE HIGH-LEVEL ACHIEVEMENT ..... 39

TRANSFERRED SHIPPING STATUS CALLS HLA™ .....	41
ORDER STATUS TIMELINESS .....	43
RECAP OF THE STRATEGIC PLANNING SESSION.....	45
THE NETWORK OF SUPPORTING ACHIEVEMENTS .....	45
SUB-DIVIDING A HIGH-LEVEL ACHIEVEMENT IN OUR PROJECT .....	46
RECAP OF THE DETAILING PROCESS .....	49
CHAPTER SUMMARY .....	50
<b>3 – Project Charter .....</b>	<b>51</b>
AUTHORITY & ACCOUNTABILITY .....	51
WHAT IS CROSS-FUNCTIONAL AUTHORITY.....	53
CHANGE CONTROL.....	57
CHANGE CONTROL RULES AND FORECASTED VARIANCES.....	58
CHANGE CONTROL PROCESS .....	59
ASSUMPTIONS AND RISKS.....	60
PRESENTATION OF THE STRATEGIC PLAN .....	62
MANAGING THE CROSS-FUNCTIONAL EFFORT.....	62
REWARDS FOR THE TEAM .....	63
RISK OF THE WORK ESTIMATES .....	64
CHAPTER SUMMARY .....	65
<b>4-Requirements &amp; Process Control.....</b>	<b>66</b>
MAINTAINING THE LINK TO THE USER AND THEIR BUSINESS RESULTS.....	66
ELEMENTS TO BLEND INTO THE PLAN .....	67
CONTROLS ON THE DEVELOPMENT PROCESS .....	68
SYSTEMS DEVELOPMENT LIFECYCLE .....	69

EVENTS-DRIVEN REQUIREMENTS ANALYSIS .....	70
BUSINESS EVENTS AND SYSTEM RESPONSE .....	72
BUSINESS EVENT SCENARIOS .....	74
DEVELOPMENT PROCESS CONTROL .....	74
REMAINING HIGH-LEVEL ACHIEVEMENTS .....	76
THE WATERFALL APPROACH .....	77
SPIRAL APPROACH.....	79
RAPID APPLICATION DEVELOPMENT (RAD) .....	80
REQUIREMENTS FOR USER ACHIEVEMENTS .....	81
CONCLUSION .....	83
<b>5 - Work Packages &amp; Estimates.....</b>	<b>84</b>
AN INVITATION TO MICRO-MANAGEMENT .....	85
ASSIGNMENTS ARE OUR MOTIVATIONAL PLATFORM .....	87
ACHIEVEMENT RATHER THAN “TO-DO” LIST THINKING.....	87
HOW MANY ASSIGNMENTS SHOULD I HAVE?.....	88
EXCEPTIONS TO THE RULE .....	89
CRAFTING GOOD ASSIGNMENTS FOR OUR PEOPLE.....	89
DEVELOPMENTAL MODE (TARGET 1 - 4 DAY DURATION).....	91
TASK MODE (TARGET 5 – 12 DAYS DURATION).....	91
SUBPROJECT MODE (15 TO 22 DAYS DURATION) .....	91
CONTRACTOR MODE (40 OR MORE DAYS).....	91
AVOIDING UNDER-STRETCHING AND OVER-STRETCHING.....	92
INDIVIDUAL ASSIGNMENTS AND TEAMS .....	92
ASSIGNMENT RISK-TAKING .....	93

WORK ESTIMATING.....	93
WORK PACKAGES.....	95
NEGOTIATING THE “CONTRACT”.....	100
WEBSITE ASSIGNMENTS.....	102
OUTSIDE CONTRACTORS.....	104
COMPLETED WORK PACKAGES.....	108
PROJECT TEAM CULTURE AND TALK.....	110
TEAM MEMBER WORK ATTITUDES.....	110
PROJECT MANAGER’S BEHAVIOR.....	112
CONCLUSION.....	113
<b>6-WBS, Predecessors &amp; Schedules.....</b>	<b>114</b>
WORK, NOT DURATION, ESTIMATES.....	114
BUILDING OUR PROJECT MODEL.....	115
WORK BREAKDOWN STRUCTURE (WBS).....	118
DESIGNING PREDECESSOR NETWORKS.....	119
PARALLELISM IN OUR DESIGN.....	121
RESOURCE CAPACITY AND SCHEDULING.....	122
RESOURCE CAPACITY.....	124
LABOR RATES & WHY WE WANT A PROJECT BUDGET.....	127
RESOURCE AVAILABILITY AND CALENDARS.....	128
FIXED COSTS.....	130
VENDOR CONTRACTS AND FIXED COSTS.....	130
THE INITIAL RIDGEWAY BUDGET AND SCHEDULE.....	131
CRITICAL PATH ANALYSIS.....	138

SLACK ..... 139

DELAY ..... 141

RESOURCE UTILIZATION ..... 142

CONCLUSION ..... 144

**7 - Optimizing & Trade-offs .....145**

    THE 4-CORNERS APPROACH ..... 147

    FINE-TUNING THE PLAN ..... 148

        CLOSED PREDECESSOR NETWORK ..... 150

        UNNECESSARY PREDECESSORS & PARALLELISM ..... 151

        RESOURCE UTILIZATION ..... 154

        SLACK & DELAY ..... 160

    TRADE-OFF ANALYSIS ..... 163

        DURATION TRADE-OFFS ..... 165

        BUDGET TRADE-OFFS ..... 172

    APPROVAL PRESENTATION ..... 176

    CHAPTER SUMMARY ..... 177

**8 - Tracking, Problems & Solutions .....178**

    LESSONS LEARNED ..... 180

        PRE-LAUNCH PEER REVIEW ..... 180

        ASSESSING PROJECT TEAM CULTURE AND LEADERSHIP STYLE ..... 181

    STATUS DATA ..... 185

        TRACKING DURATION ..... 187

        TRACKING BUDGET ..... 191

        ACTION PLAN ..... 192

A LIGHT TOUCH .....	193
EARNED VALUE ANALYSIS .....	194
COMPLICATED WAY TO UNDERSTAND EARNED VALUE .....	197
PROBLEM SOLVING AND REPORTING .....	198
HARRY'S EARLY FINISH .....	198
RAMON'S DURATION AND BUDGET OVERRUN .....	199
POLITICAL PROBLEMS WITH THE DIRECTORS REACHING CONSENSUS .....	200
SUB-PROJECT FINISHING LATE .....	201
ANOTHER SUB-PROJECT FINISHING LATE .....	202
CONCLUSION .....	205
<b>Appendix: Step-by-Step Instructions.....</b>	<b>206</b>
THE MS PROJECT® WINDOW .....	206
SETTING UP A PROJECT .....	207
SETTING USER PREFERENCES .....	207
RESOURCE LEVELING .....	209
CLICK ON TOOLS THEN RESOURCE LEVELING .....	209
ENTER WORK PACKAGES TO CREATE THE WBS .....	210
ASSIGNING RESOURCES & PREDECESSORS .....	211
PREDECESSORS AND RESOURCES .....	213
ASSIGNING PEOPLE TO TASKS: WORK & UNITS .....	214
FINE-TUNING AND TRADE-OFFS .....	215
CRITICAL PATH .....	215
SLACK AND DELAY .....	216
APPLYING COST TABLES .....	217

TRACKING ACTUAL RESULTS ..... 218

    SAVE THE BASELINE ..... 219

    CHANGE THE CURRENT DATE ..... 220

    ENTER ACTUAL RESULTS ..... 220

    ANALYZING VARIANCE ..... 222

**About the Author ..... 225**

TABLE OF SELECTED FIGURES AND ILLUSTRATIONS ..... 226

INDEX ..... 228

# 1 - IT Project Lessons

**W**e will begin by sitting in on a "lessons learned" meeting for a failed IT systems development project. Though a gloomy way to introduce the topics we're going to cover in this book, it gives us an inventory of many of the PM problems that are all too typical in today's systems development environment. After all, the point of this book is to help you avoid each and every one of them.

Pat Milbarge doodled with a green fountain pen in the margin of the "Lessons Learned" document, sketching figures of project managers hanging from a noose. Pat had notified all the project's players of the meeting but everyone was late, just like during the project. A few of the project team members straggled in. Pat received crisp nods from some team members but many just went to their seats, eyes downcast. They'd all worked pretty hard. But their hard work had produced nothing for them except association with a failed project. Oh, there were a couple that goofed off and played some games with duration estimates. But the resentment on most of their faces clearly signaled that they blamed Pat for the project's failure.

The users tromped in as a group led by the VP who'd had almost no involvement at the beginning of the project and whose time investment grew exponentially as deadline after deadline was missed.

"My gosh, yet another meeting," the VP sneered. "We're still trying to fix the mess upstairs! If anything, the number of customer complaints is even worse than before we started this disaster."

Pat capped the fountain pen, thinking that this was a wonderful way to begin the meeting and said, "Well, the idea of the "lessons learned" meeting is to try to identify what went wrong so we can improve the way we do projects."

"You people," the VP snapped, "have to do a lot better! We cannot keep having these project disasters."

"We delivered every requirement you specified," barked a senior systems analyst, already red in the face.

The VP snapped back "Go tell that to the customers who are still complaining about our bad service and how long it takes us to straighten out problems."

Pat knew it was time to regain control of the meeting. "One of the problems with our planning was that we didn't focus on reducing the number of complaints. In the beginning, we only talked about the new reports, screens and functionalities that you wanted. Then the list of requirements kept growing every week."

From the expressions on the faces of the first-line supervisors from the Billing department, Pat knew the last comment was a mistake.

"It kept growing because you never gave us what we wanted," one of the supervisors said.

The VP pushed back from the table and stood up, "This is getting us nowhere!" Then the VP pointed a finger at Pat's face and said, "You were seven months late and \$300,000 over budget and we still have problems with a lousy system."

The VP turned and took two steps toward the door before whirling back and saying, "And what I like the least about the way you people do projects is that all the bad news always comes at the end, when we can't do anything about it!"

## ***BAD SURPRISES WHEN IT'S TOO LATE***

As the project team and user group continued the debate, Pat thought about the VP's last words. There had been a lot of bad news late in the project. Sure, some of it could be attributed to overly optimistic estimates and some to scope creep. But as the completion date kept getting pushed out, people got real nervous about reporting any more slippage. Whenever they did, the sky fell on them from all quarters; including, Pat had to admit, from the project manager. Although they should have been honest enough to report problems, Pat could have done a better job of accepting bad news and protecting them from executive tongue-lashings. Pat knew that getting bad news was better than not hearing about it because then no corrective action was possible. Another real problem was not having the tools to spot small problems early. With the project plan they had built, both the team members' status reporting and Pat's reporting to the IT and user management groups was subjective. No one really knew how the project was going until they got near the end.

A supervisor's angry voice broke Pat's reverie, "We never understood that technical mumbo jumbo you made us sign off on! Signing a list of requirements when you don't understand it means nothing."

"Well if it meant nothing, why did you keep adding to the list each week?" an analyst snapped back. "If you want these projects finished on time you can't keep changing things!"

We "frame" the project within business requirements that link user and IT achievements within measured business results.

## ***CHANGES, CHANGES AND MORE CHANGES***

Both the analyst and the supervisor were right. Sure, they'd tried to "freeze" the requirements and they'd gone through a very thorough approval and sign off process on the technical specifications. But then every week the list of features and functionalities grew. The users saw a report or screen layout and said, "This won't work for us the way you've got it." Then the project team member said, "Well, I'll have to fill out a change request because that's a change and it'll take more time and cost more money." The two would go around and around debating whether this was or was not a change and it would be escalated. Then the same debate would occur at a higher level with everyone becoming more and more angry. Most times the change was added to the plan but usually with no corresponding increase in budget or duration. If Pat insisted on budget and duration increases to reflect the cost of a change in the project, the team was blamed for doing a poor job of laying out the requirements.

The fact was the users did not understand the technical language of the requirements and there was very poor linkage between those technical requirements and the improvements in operating performance. It was also true that neither Pat nor the analysts had a clear understanding of the business and performance results the users were seeking from the project. Of course, at the beginning the users did not seem to understand what success was either. In fact, their role in the project, beyond a vague mention of training, was not included in the plan. There was tremendous pressure to get started producing code and little interest in integrating user process changes into the planning.

Another loud voice shouted, "The software was crap!"

"We didn't develop a bad system," a programmer retorted. "You didn't decide what you really wanted until we were in user testing."

Pat had to admit that there had been some problems with the quality of the system that didn't surface until user testing. But there had been so many changes throughout the project that they'd cut back on the time allocated to testing and software quality processes. The result was serious flaws that came to light during acceptance testing and had to be fixed on top of another round of user changes. Those changes late in the game were really expensive and cost hundreds of times what the same changes would have cost during requirements analysis.

The programmer, whose face was an ugly blotch of red and white, finished shouting at the users and stormed from the room. Pat knew the programmer had worked very hard on the project, spending a lot of evenings and weekends to get the work done.

## ***PROJECT TEAM WANDERING IN THE WILDERNESS***

As everyone watched the programmer's exit, Pat scanned the angry frustrated faces of the project team members. They were a pretty good group. Pat speculated that many would be gone before the next project. The original completion date had been "plucked from the sky" before Pat had even been selected as project manager, much less done any analysis. Pat argued about the date but got nowhere trying to convince the executives of the impossibility of reaching it. As a result, the duration and work estimates were jokes and everyone on the project team knew they would fail to hit the due date before they even started work. As a result, they had no commitment to their individual due dates. The situation got worse when the users kept adding new requirements with few increases in budget or duration.

What really dragged the project team into their deep depression was the hours they had to work. Most of them were on several projects but no one was managing their overall workloads or setting priorities in terms of which project's work should come first. All those functional managers' promises of "full support" proved to be worth very little because they pulled their people off the project whenever something came up.

Just then, one of the users raised the 60 page project plan over her head and threw it at the garbage can, missing by four feet and scattering Gantt charts all over the floor. Pat looked at all those Gantt charts fluttering to the conference room floor and wondered if the project plan had been a little too detailed. They started the planning with a lot of talk about the business outcomes and clear direction. But all the pressure to get started with the work led to the project plan being little more than a very detailed list of IT micro-activities. Was that micro-management, Pat wondered? And the plan had not really specified the things required from the user, like process changes and staffing. Either way the project plan had been useless. They weren't but two weeks into the effort when people started saying, "We've already done that" or "We can't do that yet because..." so all those details in the project plan really didn't provide the project team with guidance. And some of the more experienced people seemed to make a point of doing things in sequences other than what was laid out in the project plan. They explained that they'd found a better way to do it but Pat often felt that their point was just doing it differently than the plan.

In the still rapidly deteriorating "lessons learned" meeting, a project team member screamed, "Why didn't you tell us in the beginning about the big problem with customer complaints?"

Team members who have to guess the end result that is expected from them don't give us their "best work" nor do those we micro manage.

"We did!" chorused the three remaining people from the user area.

"Oh, you mentioned it but two-thirds of what we did had nothing to do with reducing customer complaints."

The user said, "But as long as you were making changes to the system, we wanted to fix everything that was broken."

## ***PLANNING, MISSION STATEMENT MUSH***

The last three supervisors from the user group walked out of the meeting followed shortly thereafter by the last two members from IT. That last exchange captured the essence of the project. As dumb as it sounded, they had done a great deal of irrelevant work and it wasn't until toward the middle of the project that they understood how the user would measure the business success of the project. Pat sighed in exasperation. That lack of understanding of the user's business purpose had also made change control impossible. Instead of being able to evaluate the change requests on the basis of whether or not they contributed to the desired business results, they were left with only the ability to argue about the vague "value" of changes.

Pat wondered what they could have done differently. During the planning, the VP hardly gave them a moment of time; delegating planning to lower-level decision-makers who were equally unaware of the criteria that would be used to judge the project's success. Why wouldn't the VP give them any time? Probably because every meeting they ever had with the VP was devoted to detailed technical discussions that simply were not of interest to that level of decision-maker. Instead of setting up a network of business achievements that were objectively measurable, Pat settled for mission statement mush which got approved because there was nothing in it but vague generalities.

Pat tiredly rose and left the room, thinking two thoughts. First, this happens to us over and over again but we never learn any lessons from it. Second, wouldn't it be nice if I could do it all over again.

## ***ACHIEVEMENT-DRIVEN PROJECT MANAGEMENT***

We'll develop techniques to address these problems in the remaining chapters of this book. The foundation for these techniques will be our Achievement-driven Project Management Methodology (AdPM™). We build this foundation with unambiguous business achievements that define success for the project as a whole and each of its components before we start. Measured achievements require that we think about end results, rather than just activities, before we start work. But the payoff for the PM who makes this intellectual investment comes each week in the form of:

- ❑ Team members who know what is expected of them before they start work
- ❑ Executives who understand what they are “buying” from the project and, as importantly, what they will not get
- ❑ Scope and change control processes that are based on hard-edged objective data, not opinion
- ❑ Small project plans that are easily maintained and updated so the PM knows exactly where the problems are.

It is normal for all of us to think in activity terms; what we want people to do. To conceive measured achievements we need to go a step further. We think about what we want people to do and then how we will assess their performance when they are finished. It is this later measurement that is our measured achievement. Let’s consider a few examples of activities and their conversion to measured achievements:

Activity	Measured Achievement (Success Definition)	Type of Measured Achievement
Complete design	User Committee approves design	Approval achievement – We measure success by the user or a technical reviewer signing off
Develop customer history GUI	Data for top 5 customer inquires accessed on 1 screen	Metric that we can objectively observe
Design the database	Database design approved by QC as in compliance with Standard Operating Procedure #6	Approval achievement with reference to an IT department development process control
Acceptance testing	User test meets specification for Events #1 through 7 in Work Package 7-3-5	Yes/no achievement with reference to detailed specification or a work package list of measured requirements
Train the users	90% of Customer Service Reps score 80% or higher on written test of new procedures	Metric that we can calculate
Install efficient packaging system	85% of customer orders packaged and labeled in less than 120 seconds	Metric that we can objectively measure

While the measured achievements in the middle column are in a number of forms and at a number of levels, each gives us, or the person doing the work, a clear and unambiguous performance expectation. It tells them when they are done and also makes clear what level of performance is good enough. For an executive, measured achievements detail what they are getting and also what they are not getting. Quantifying expectations before we start work, not halfway through, is a real

asset in scope control. Consider the customer service training above. Laying out this clear end result tells the Trainer the standard that the trainees must meet after the class. It also tells an executive how well the reps will be trained. If that level is not good enough, we can change the plan now rather than having to redo the training after the class.

Let's look at how we'll use this measured achievement thinking in our project management process.

## MEASURE OF SUCCESS(MOS™) & ACHIEVEMENT NETWORK (HLA™)

The biggest measured achievement in our plan is the measure of success (MOS™). It quantifies business success for the project as a whole. Once we have an understanding of the MOS, we'll craft a network of high level achievements to deliver on that MOS. Not all of these HLAs™ will be systems or IT related. Some will capture the customer's business achievements that, in combination with the IT effort, will deliver the MOS™. As an example, say we have a project to build a customer history system for our customer service department. When we understand their MOS™ of decreasing by 25% the time it takes to respond to a customer's order status inquiry, we can build a high-level achievement network that includes not only the system performance but also the customer's achievements for training the reps and altering their internal processes. While we have no accountability for those user achievements, including them in the plan gives us the information about how our systems effort has to support their operational plans. We find out what those departments need from us before we start work, not halfway through.

We may track the entire plan for a customer decision-maker or they may track it. The important point is we want to manage our IT effort as part of an integrated project effort because that improves the odds of success on our part of the effort.

Before we work through the planning process, there is a core technique that we'll use in Broadbrush planning and throughout the project; it's called quantified trade-offs.

## TRADE-OFFS AND SCOPE CONTROL

Our measured achievement focus lays the groundwork for managing with quantified trade-offs. The trade-offs we will work with come from the "four corners" of the project we will develop. Instead of describing the project with a budget, due date and a long narrative, we describe it with four quantified dimensions. These 4-Corners are:

- Measure of success (MOS™)

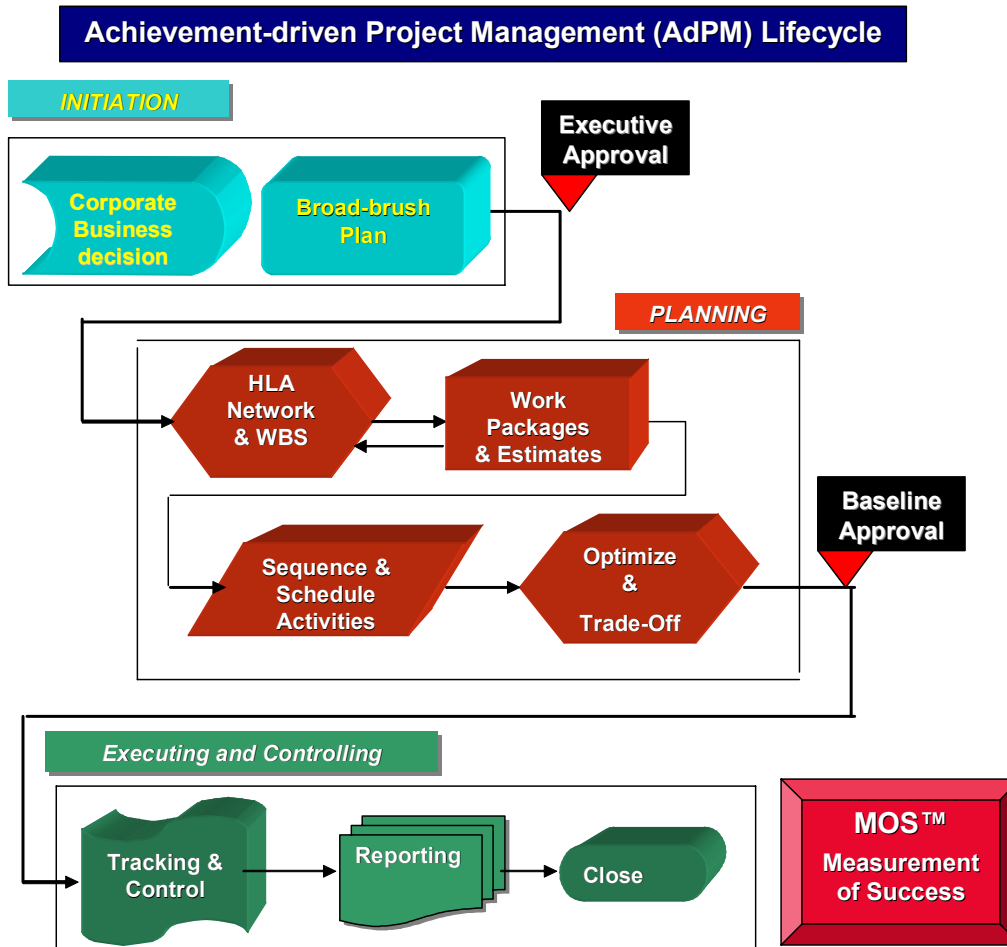
Executives are accustomed to evaluating projects with one measurable dimension, duration, or two, budget & duration. We'll give them four and better decision-making.

- Budget
- Duration
- Risk (probability of success).

Our intent is to build a project plan where each of these four corners is quantified and we can discuss quantified trade-offs between them. We'll establish the idea of trade-offs between these four corners early in the project and then we will use it for scope control during:

- Detailed planning
- The final project approval presentation and
- Every week as we track actual results and deal with changes and problems.

Rather than try to "fight" with the user/client executives about changes to the plan or changes to the requirements, we will present data on these trades-off before they ask. If the executives wish to shorten the duration of the project, we will calculate the impact on one or more of the other "corners" of the project. We can certainly shorten the duration but the trade-off may increase the cost, reduce what we achieve for the business or lower the probability of success. This trade-off mentality is the key to maintaining a high probability of a successful project by giving us an effective and data-based approach to change control. It also allows executives to exercise strategic control over what they are "buying" from the project and that can earn us many benefits in the relationship.



Our project management lifecycle diagram summarizes the achievement-driven project management process we'll be following throughout the remainder of the book.

The project plan flows from the customer's decisions about business objectives in measured achievement form. It gives us a strategic framework which we will verify with our Broadbrush planning process. Then we'll detail the plan to build a foundation for executive decision-making and our tracking of project progress.

## **2 - Strategic Project Planning**

In this chapter, we'll work through the process of strategic project planning, seeking to "frame" our project within boundaries of measured business outcomes in the user's operations. We'll also work with the user to establish and gain their approval of the strategic framework of the project and the processes we'll use to deliver their end result. The strategic plan includes:

- ❑ An objective measure of project success (MOST™ )
- ❑ A high-level achievement network (HLA™ ) that lays out, in measurable terms, our path to the MOST™ and quantifies the boundaries of the project's scope, including both user and IT achievements
- ❑ Assessment of risks, mitigation strategy and strategic assumptions
- ❑ The project charter, including authority structures and accountability relationships
- ❑ Change control processes and decision-making rules.

Having this framework in place before we start work provides solid scope control and substantially increases the probability of the project being a success in the user's eyes. However, strategic planning is a difficult process, which is why it is skipped on so many projects.

### ***WHY STRATEGIC PLANNING IS SKIPPED***

Most IT projects start with the assembly of a grocery list of requirements which grows each week during the project because there is no strategic plan to restrain the expansion of the project or target its success. We have little ability to define what's in and what's out of the project. Oh, we see long narratives supposedly defining scope and objectives but they rarely contain objectively measurable definitions of success and the measured steps we'll take to reach that end business result. So why do people skip strategic planning? Because it requires:

- ❑ Access to executive level decision-makers
- ❑ Learning the user's business

When projects are defined solely as the delivery of new functionality by IT, we are unlikely to have a satisfied user at the end or produce value for the business.

- ❑ Speaking the user's language
- ❑ Negotiating performance commitments up front
- ❑ Coping with a certain amount of conflict that occurs whenever we make tough decisions
- ❑ Conceiving a cross-functional effort that spans functional "silos" and nudges the hierarchy.

No wonder people skip this strategic planning; it's so much easier just to start work.

Let's explore several of these challenges in more detail.

First, the language of strategic planning is not technical; it is measured performance improvement in the user's business and operating functions. We talk their language, not ours, with a focus on measured business achievements. We talk about the "accuracy of invoices" and the "time it takes to process a transaction" not the GUI, database or network capacities required. It is difficult to keep the planning discussions at a business achievement level so we can reach agreement on the measured business benefits that will be our target. Many who will be in these discussions will want to get into the delicious technical details or talk about what the screens should look like, not what we have to achieve in the business. The PM and IT project team members can be their own worst enemy in this regard. We are more comfortable talking about areas of our expertise than the users' business. But we need to engage executive decision-makers in this process, and we quickly lose them if we let the planning sink into the "activity trap" of technical features and functions.

Second, few PMs enjoy conflict and strategic planning triggers it. Rather than burying disagreements, a strategic planning process that focuses on hard-edged and measurable business results brings them to the surface. We're not creating conflict. Rather, we want to resolve as much of the existing conflict over business results and "what's in the project" as early as possible. No IT project can meet all of the user group's expectations about the features, functionalities, reports and screen displays. The question is when in the process we face this issue and control the scope. We can defer the existing conflict or smooze it over in the interest of "getting off to a good start." Then we face it toward the end of the project, when changes are most expensive. Or, we begin scope control early, which is much better than waiting until the duration and budget start to slip.

So those are some of the challenges we face in strategic project planning. As a rule of thumb, every hour spent on this process saves 10 hours during the life of the project. We'll use a two-step strategic planning process to frame our tactical planning. It substantially increases the probability of delivering the business results the user wants within the time frame and budget to which we will commit. With an approved strategic plan, we'll begin the project with: