PANEL #2

TOOLS FOR SOFTWARE MANAGEMENT

D. Reifer, Reifer Consultants Inc.
J. Valett, NASA/GSFC
J. Knight, University of Virginia
G. Wenneson, Informatics General Corporation

SOFTWARE MANAGEMENT TOOLS: LESSONS LEARNED FROM USE

Donald J. Reifer, President
Reifer Consultants, Inc.
25550 Hawthorne Blvd.
Torrance, California 90505

Abstract: Over the last five years, considerable progress has been made in the area of software resource estimation, management and control. Numerous tools have been developed and been put into use that allow managers to better plan, schedule and control the allocation of the time, workforce and material needed to develop their software products for NASA applications. Currently, over 300 commercially available software project management tools exist including about 180 project scheduling and control packages for an IBM personal computer-based workstation[1]. In addition, numerous tools exist for estimating software costs, measuring software progress through earned value concepts which rely on reporting milestone completions, maintaining configuration integrity over the software product data bases and measuring software quality. The literature is full of promises and details when it comes to these tools and it becomes confusing when you try to sort out what they really can and can't do when you read the sales fiction. In addition, much of the experience associated with transitioning these tools onto operational projects where managers are trying to use such aids to reduce the time it takes them to plan and control the delivery of their complex software products has not been recorded or shared.

The purpose of this presentation is to remedy this situation by discussing the author's recent experiences in inserting software project planning tools like those mentioned above onto more than 100 projects producing mission critical software. The author will briefly summarize the problems the software project manager faces and then will survey the methods and tools that he has at his disposal to handle them. He will then discuss experiences his firm and users of the RCI developed Project Manager's Workstation (PMW) and the SoftCost-R cost estimating package have had over the last three years. Finally, he will report the results of a survey conducted by his firm which looked at what could be done in the future to overcome the problems experienced and build a set of usable tools that would really be useful to and used by managers of software projects.

THE PROJECT MANAGER'S WORKSTATION

The Project Manager's Workstation (PMW) was a prototype system that was built 3 years ago for a military client to research the following issues:

1. What tools does a software manager really need and what tools will he really use on the job?

2. What are the criteria which govern the acceptability of management tools by managers, not computer scientists?

3. Can management data be bridged between commercial tools developed by different manufacturers and resident on different machines?

---

[1] P. Kane, J. Bruscino, T. Pillsbury, D. Reifer and B. Strahan, Project Management Tool Survey Report, Note RCI-TN-145, 29 March 1985.

The PMW is a collection of management tools that runs on a dual floppy IBM personal computer with 512 KB. It has the following capabilities: resource planning, scheduling and control via a Work Breakdown Structure (WBS); Gantt and PERT chart (tabular and graphical) preparation and drawing; user-oriented report generation for cost-to-completes, schedule-to-completes and earned value determination; local bridges to packages like 1-2-3 and dBase on the personal computer and global bridges to packages like PAC-II and VUE on mainframes; and a personal time manager which allows relational development and searches of action item lists, calendars, distribution lists and telephone lists.

The PMW was designed as a rapid prototype with both usability and technical capability in mind. We hoped to learn from it as we put it into prototype use within organizations who were willing to try to employ it on their projects. It has been distributed to over 200 people over the last 3 years. Each user was required to attend a hands-on course on the system where he/she was taught how to use the package for managing a software project. A generic WBS was developed and inserted into the package to guide its users in consistent work task identification and cost data collection.

Recently, RCI surveyed the users of the package to get their feedback and to understand what their real requirements were when it came to project management tools. It was interesting to learn the following:

- The man/machine interface design makes or breaks the system. The user interface must be easy to learn and easy to use. It should be picture-oriented, function key driven and menu-based. Tool designers shouldn't assume managers know how to type, use a computer and/or will read manuals. They won't based upon our experience. To combat this, the package must have built-in "HELP" and safeguards against inappropriate usage.

- Most managers object to project management systems because they are required to do a lot of data input. Managers do not have the time, desire or skill to do it and often, don't do it right. Subordinates don't have the knowledge or the experience to do it correctly. Therefore, the system must support both working together to relieve the manager of the drudgery of getting the first set of workable plans into the system. To combat this, many tool designers should looking at "games" and should try to adapt their concepts to making data inputting "fun".

- Most vendors do not mechanize all the features and functions they put in their manuals. This makes it extremely difficult to interface packages together into an integrated system. File interchange performance is the critical issue because management users will not tolerate lengthy delays in getting responses to their questions. In the development of the PMW, we had to drop about half of the candidate packages from consideration and build our own modules to replace them as a result. Tool designers should therefore only rely on a core set of capabilities when they plan to use commercial packages.

- Global bridging or linking a micro-based tool to a mainframe-based system is much more difficult than first expected. Vendors do not

like to give you the file interchange formats and reverse engineering is the only alternate solution to getting this needed information. As a consequence, it took us 3 times more effort than originally planned to provide this capability. Tool designers should not count on the vendors of packages to make their jobs easy. Instead, they should adopt a standard file format like DIF and consider only packages that implement it.

- According to our users the most useful tools were work planning oriented, the most used tools were time management oriented and the most wanted tools were "what-if" oriented. This is not surprising and should be factored into future system designs.

- Because the state-of-the-art is moving towards networking, managers wanted to evolve their tools so that they could interrelate what their people were doing at different sites via their management tools. According to their wish lists, they wanted to do things like schedule a meeting on their people's calendar electronically and to preview deliverables in their work units libraries via remote inquiry privileges.

SOFTCOST-R

In another effort, RCI developed a cost estimating package based upon the work of Dr. Robert Tausworthe called SoftCost-R[2]. In essence, RCI spent six person years of effort to productize the experimental work done for the Jet Propulsion Laboratory. SoftCost-R is hosted on an IBM personal computer and versions exist for all of its models including the PC/XT and PC/AT. The primary feature RCI implemented was usability. Learning from our PMW experiences, we built a user-friendly screen editor to make the package easy to learn and easy to use. Since we introduced our product earlier this year, over 20 organizations have acquired it and are using it to predict their costs. Most of these organizations work on small to medium-sized projects developing software for embedded applications. The capabilities of SoftCost-R are similar to other parametric and statistical cost models on the market today like COCOMO, PRICE/S and SLIM. The key difference has to do with the ease with which the management user can employ the model to answer the "what if" questions he so desperately needs to answer.

Again, RCI surveyed its users and members of its development team to determine what lessons could be derived from its experiences to-date. This was very valuable to us because we were in the midst of planning enhancements to our current product and wanted to factor these lessons into our future releases. It was interesting to learn:

- The number one issue on the minds of management when it comes to costing is sizing. How can one determine in advance how big the program will be when you don't have the foggiest idea of what the system architecture will be was one of the comments heard during one of our interviews. While some research in this area is underway, managers will be reluctant to accept the results of cost models unless some of it pans out.

---

[2] Robert C. Tausworthe, Deep Space Network Software Cost Estimation Model, JPL Publication 81-7, 15 April 1981.

- Most of our users employed at least two cost models to cross check each's results. The most popular model was COCOMO and most of our users employed it manually from the book.[3] The reason for this popularity seemed to be its availability. Unfortunately, many users in our survey did not seem to fully understand the model's scope or limitations and were misusing it on the job.

- Calibrating a cost model to the organization using it is the hard part. Most organizations using our model did not have cost data available to either calibrate the model or validate its accuracy. Even if they had data, it was hard to make any sense out of it. Less than 5% of our users collected cost data as a norm and few had a framework in place for cost estimating. While cost models like SoftCost-R forced these organizations to gather data, most of it was not statistically homogeneous. Models must therefore be architected so that their calibration points and sensitivities are known and easily altered. In addition, the model must come with a known calibration data base in order for its users to have enough confidence in the model to believe its results.

- Non-management user's put too much reliance on models. Because a model gives them an answer, many believe it is right and don't do any more homework.

- Management user's tend to be more skeptical and don't believe the results of models even if they are perfectly calibrated to their projects and their environments (which they are not). Often, this is because managers really don't want to know the truth - the software is going to cost more than they expected and they don't have sufficient budget allocated for it.

- Many simple and mundane packaging concepts can make a model acceptable to a management user who will sacrifice capability to get something he can get answers from. Good user engineering goes a long way with managers who neither have the time nor the desire to become professional parameticians.

CONCLUSIONS

While the results reported seem logically and self-apparent, few seem to have paid attention to them in the past. Considerable attention needs to be paid to the packaging of tools when they are exported to production organizations from tool developers. The author sincerely hopes that this presentation will stimulate renewed emphasis on this important topic. Afterall, the results are based upon a survey of over 200 management users and are not only the author's opinion.

---

[3] Barry W. Boehm, Software Engineering Economics, Prentice-Hall, 1981.

THE VIEWGRAPH MATERIALS

for the

D. REIFER PRESENTATION FOLLOW

SOFTWARE MANAGEMENT TOOLS: LESSONS LEARNED FROM USE

4 DECEMBER 1985

PRESENTATION AT THE 10th ANNUAL
NASA/GSFC SOFTWARE ENGINEERING WORKSHOP

D. Reifer
Reifer Consultants
6 of 22

Reifer Consultants, Inc.

25550 Hawthorne Boulevard, Suite 208/Torrance, California 90505

PURPOSE OF BRIEFING

- To discuss our experiences with our software management tools
  - Project Manager's Workstation (PMW)
  - SoftCost-R Estimating Package
- To communicate the lessons we have learned - the hard way through use
- To hopefully influence your efforts and help you avoid the mistakes we have made

THE MANAGEMENT PROCESS

BEGINS BY:

- PLANNING
- ORGANIZING
- STAFFING

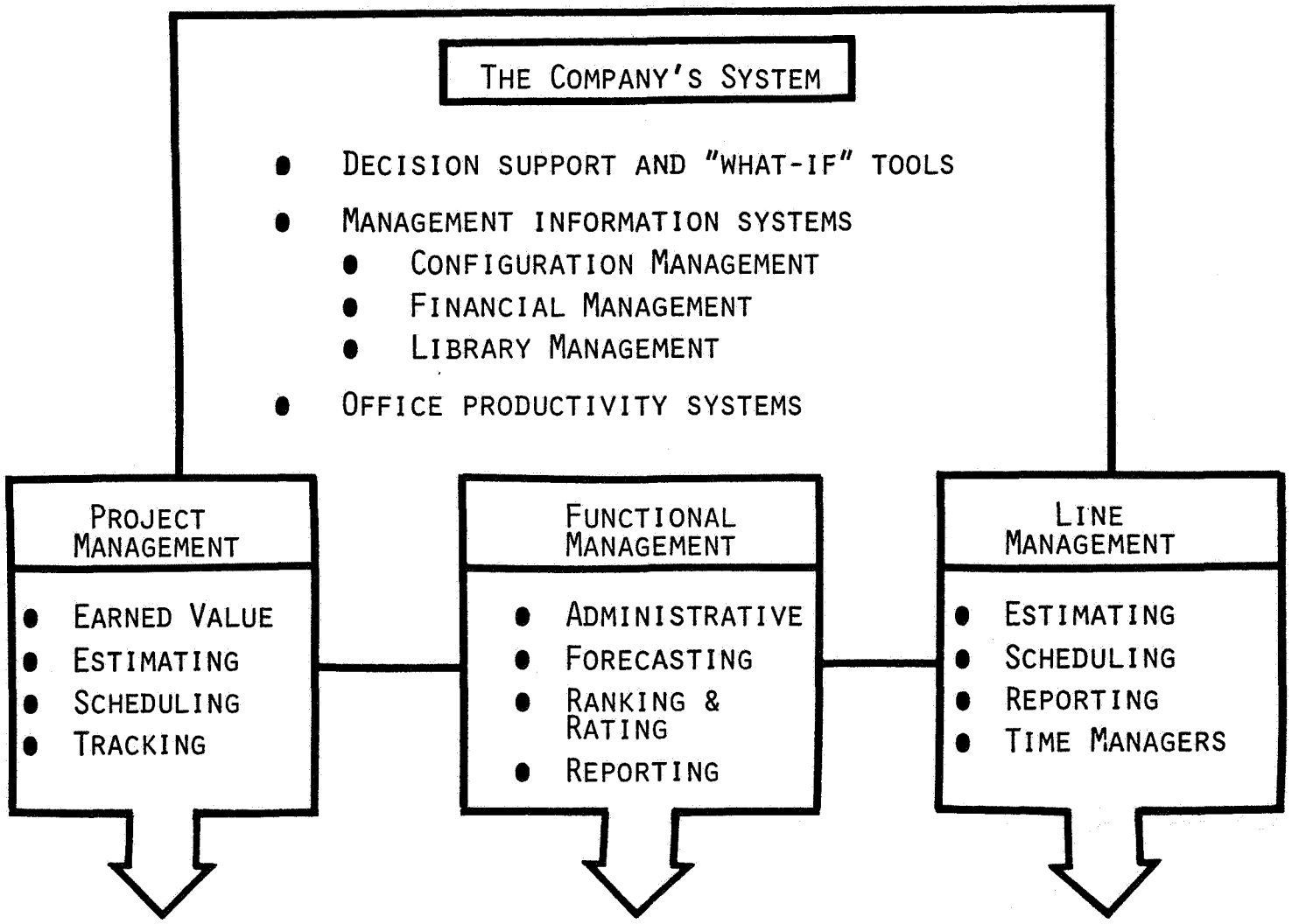
→ THE TEAM - AND THEN

- COMMUNICATING
- MOTIVATING
- INTEGRATING
- MEASURING
- CONTROLLING
- & DIRECTING

THE EFFORTS OF THE TEAM

THROUGH AN INTERATIVE PROCESS

NECESSARY MANAGEMENT TOOLS



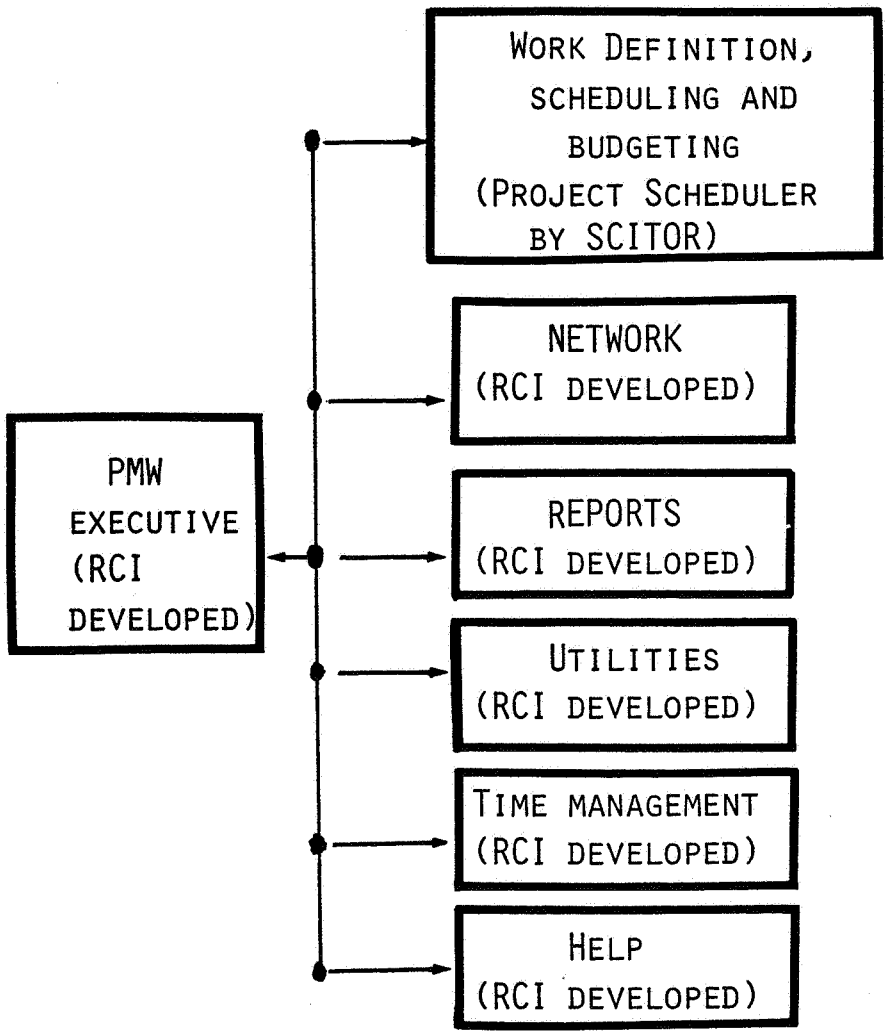
OVER 300 PACKAGES EXIST TO SUPPORT THESE FUNCTIONS

PMW: AN OVERVIEW

PMW IS AN EXPERIMENTAL, INTEGRATED PROJECT MANAGEMENT PACKAGE

- DEVELOPED TO RESEARCH THE ISSUES ASSOCIATED WITH WHAT TOOLS MANAGERS NEED AND HOW DATA COULD BE BRIDGED BETWEEN MACHINES
- USES A PACKAGE CALLED "PROJECT SCHEDULER" AS ITS NUCLEUS FOR WORK STRUCTURING AND SCHEDULING
- ADDS VALUE BY INCORPORATING AN ADVANCED MANAGER/MACHINE INTERFACE, GRAPHICAL PERT, TIME MANAGEMENT FUNCTIONS AND IMPROVED REPORTING CAPABILITIES
- IS SCREEN-ORIENTED AND USES FUNCTION KEYS TO SIMPLIFY THE INTERFACE
- PRODUCES A VARIETY OF REPORTS INCLUDING:
  - ACTION ITEM LIST
  - ADDRESS BOOK
  - COST-TO-COMPLETE
  - CRITICAL PATH
  - KEY MEETING & BRIEFINGS
  - RESOURCE PROFILES
  - SCHEDULE-TO-COMPLETE
  - WORK BREAKDOWN STRUCTURE

PMW: FUNCTIONAL CAPABILITIES



- CREATE WORK BREAKDOWN STRUCTURE
- GANTT CHART FOR WBS ELEMENTS
- COST FOR WBS ELEMENTS

- NETWORK CHART

- GANTT CHARTS
- PROJECT COSTS
- COST-TO-COMPLETE

- WBS
- • •

- PAC II INTERFACE
- LOTUS "1-2-3" INTERFACE

- ADDRESS BOOK
- CALENDAR OF APPOINTMENTS
- MEETING/BRIEFING LIST
- ACTION ITEM LIST
- COMMANDS
- MODE ASSISTANCE
- TUTORIAL

PMW: LESSONS LEARNED I

- THE MANAGER/MACHINE INTERFACE MUST BE USER-FRIENDLY
  - PICTURE-ORIENTED, FUNCTION KEY DRIVEN AND MENU-BASED
- DON'T ASSUME MANAGERS KNOW HOW TO TYPE, USE A COMPUTER AND/OR WILL READ MANUALS
  - THE PACKAGE MUST BE EASY TO LEARN AND MUST HAVE BUILT-IN SAFEGUARDS AND "HELP"
- THERE IS NO WAY AROUND THE PROBLEMS OF INITIAL DATA ENTRY
  - MANAGERS DON'T HAVE THE TIME, DESIRE OR SKILL TO DO IT
  - SUBORDINATES DON'T HAVE THE KNOWLEDGE OR EXPERIENCE TO DO IT
  - THE TOOL MUST SUPPORT BOTH WORKING TOGETHER TO GET THE JOB DONE PERHAPS USING GAMES
- VENDORS DO NOT MECHANIZE ALL THE FEATURES/FUNCTIONS IN THEIR MANUALS

PMW: LESSONS LEARNED II

- VENDORS DO NOT MAKE IT EASY TO INTERFACE PACKAGES TO OTHER PACKAGES
  - FILE INTERCHANGE PERFORMANCE IS THE CRITICAL ISSUE BECAUSE USERS WILL NOT TOLERATE DELAYS IN GETTING RESPONSES TO THEIR QUESTIONS
- GLOBAL BRIDGING OR LINKING A MICRO-BASED TOOL TO A MAINFRAME-BASED SYSTEM REQUIRES LARGE EFFORTS
  - EVERY ORGANIZATION IS RUN DIFFERENTLY
- THE MOST USEFUL TOOLS ARE WORK PLANNING ORIENTED
- THE MOST USED TOOLS ARE TIME MANAGEMENT ORIENTED
- THE MOST WANTED TOOLS ARE "WHAT-IF" ORIENTED
- THE STATE-OF-THE-ART IS MIGRATING TO NETWORK-ORIENTED
- THE PMW-II SHOULD TAKE ADVANTAGE OF FACTS AND TRENDS

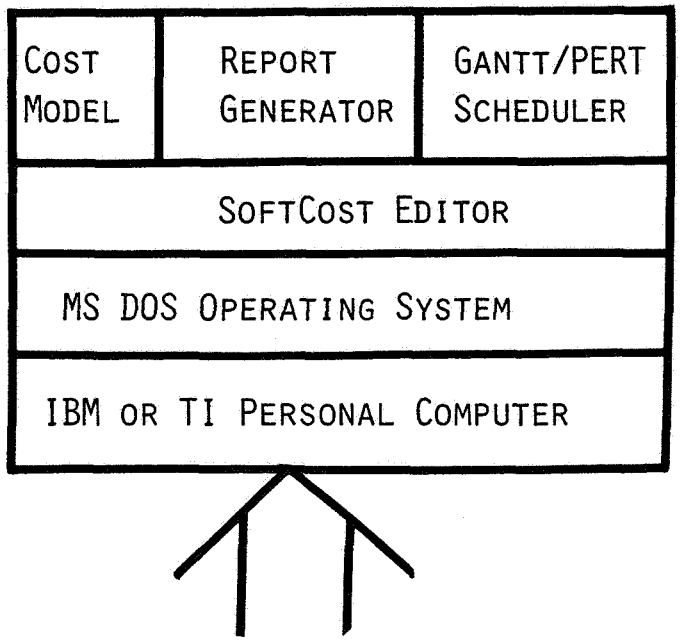
SOFTCOST-R: AN OVERVIEW

- DEVELOPED WITH USER-FRIENDLINESS AND ACCURACY IN MIND
- USES ABOUT 60 SIZING AND PRODUCTIVITY FACTORS TO PREDICT RESOURCES, DURATION AND STAFFING REQUIREMENTS
- COMPUTES CONFIDENCE FACTOR FOR DELIVERING ON-TIME AND WITHIN BUDGET FOR ANY PROPOSED EFFORT AND DURATION
- PROVIDES POWERFUL "WHAT-IF" ANALYSIS AND PLOTTING FEATURES
- GENERATES SCHEDULE AND RESOURCE ESTIMATES FOR ABOUT 50 TASKS MAKING UP A PROJECT
- PRODUCES A STANDARD WORK BREAKDOWN STRUCTURE FOR SOFTWARE DEVELOPMENT TASKS
- IS SCREEN-ORIENTED AND PERMITS ALL MODELS PARAMETERS TO BE CHANGED BY SIMPLE EDITING PROCESSES

SOFTCOST-R: OVERVIEW DIAGRAM

INPUTS

- MASTER MENU
- SIZING SCREEN
- COMPLEXITY FACTORS
- MANAGEMENT FACTORS
- STAFFING FACTORS
- ENVIRONMENTAL FACTORS
- HELP SCREENS



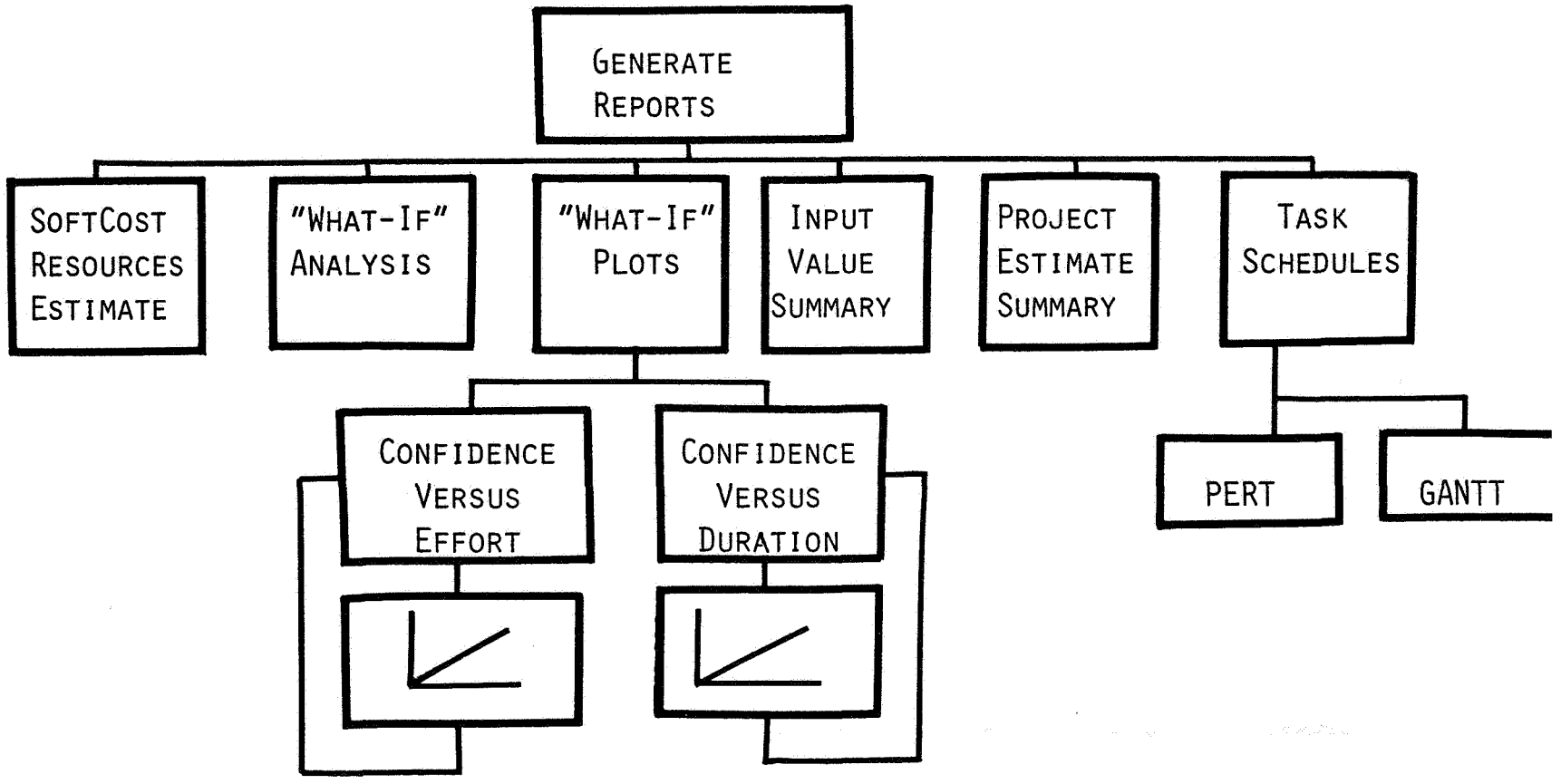
OUTPUTS

- SOFTWARE RESOURCES ESTIMATE
- "WHAT-IF" ANALYSIS REPORTS
- "WHAT-IF" PLOTS
- INPUT VALUE SUMMARY REPORT
- PROJECT ESTIMATE SUMMARY REPORT
- TASK SCHEDULES

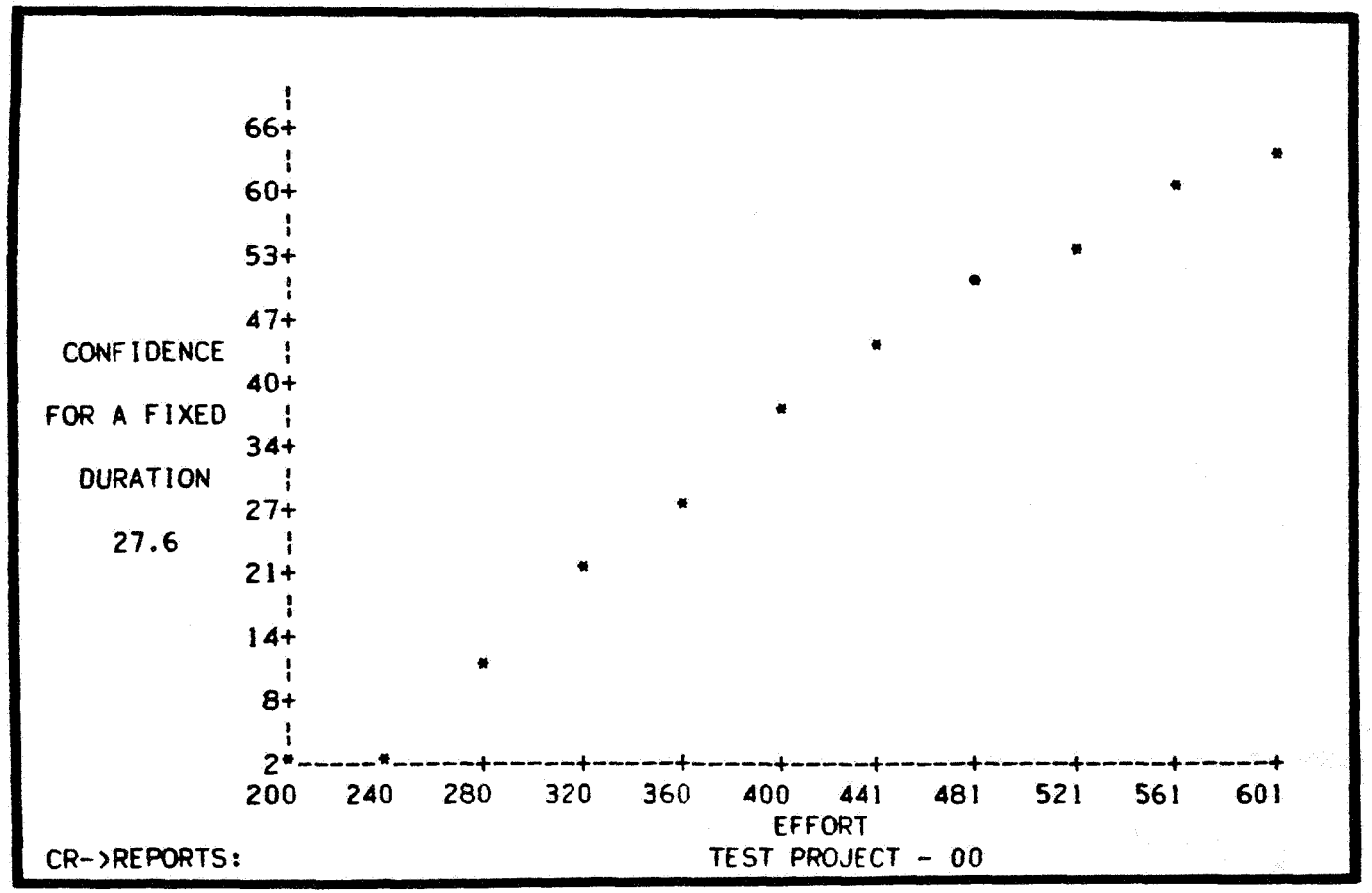
SOFTCOST-R IS EASY TO USE

- EASE OF USE AND LEARNING ARE PRIMARY DESIGN GOALS
- SOFTCOST-R IS MENU-ORIENTED USING FILL-IN-THE-BLANKS NOTATION AND FUNCTION KEYS
- SOFTCOST-R USES "HELP" LINES ON A SPLIT SCREEN TO LOCALIZE THE DEFINITION OF TERMINOLOGY AND ASSIST USERS IN INTERPRETATION
- SOFTCOST-R HAS BUILT-IN LIMIT CHECKS AND IS HARD TO MISUSE
- SOFTCOST-R IS DELIVERED WITH AN EXAMPLE CALLED "TEST" INTEGRATED INTO ITS USER'S MANUAL AND DELIVERED WITH THE PACKAGE
- SOFTCOST-R IS VALIDATED PRIOR TO SHIPMENT AND IS MAINTAINED UNDER STRICT LIBRARY CONTROL
- USERS HAVE BEEN ABLE TO GENERATE THEIR FIRST ESTIMATE WITHIN ONE HOUR OF GETTING ON THE SYSTEM

SOFTCOST-R: REPORT GENERATION FLOW



SOFTCOST-R: "WHAT IF" PLOTS



SOFTCOST-R: LESSONS LEARNED 1

- ORGANIZATIONAL PRECONDITIONING IS NEEDED
  - MOST ORGANIZATIONS DID NOT HAVE COST DATA AVAILABLE TO EITHER CALIBRATE THE MODEL OR VALIDATE ITS ACCURACY
  - EVEN IF THEY HAD DATA, IT WAS HARD TO MAKE SENSE OUT OF IT
  - FEW ORGANIZATIONS COLLECTED COST DATA AS A NORM AND FEW HAD AN ESTABLISHED FRAMEWORK IN PLACE FOR COST ESTIMATING
  - COST MODELS FORCED ORGANIZATIONS TO GATHER COST DATA AND ESTABLISH AN ESTIMATING FRAMEWORK
  - IN SOME CASES, COST MODELS FORCED CHANGES IN BUSINESS PRACTICES THAT SEEMED DISRUPTIVE BUT REALLY WERE NOT

- CALIBRATING THE MODEL TO THE ORGANIZATION IS THE HARD PART
  - MODELS MUST BE ARCHITECTED SO THEIR CALIBRATION POINTS AND SENSITIVITIES ARE KNOWN AND EASILY ALTERED
  - ORGANIZATIONS ARE DYNAMIC AND MODELS MUST REPLICATE THEM

- THE MOST POPULAR MODEL SEEMS TO BE COCOMO BECAUSE OF ITS AVAILABILITY
  - MOST USERS EMPLOY IT MANUALLY WITHOUT UNDERSTANDING ITS SCOPE OR LIMITATIONS

SOFTCOST-R: LESSONS LEARNED 2

- USER'S OFTEN PUT TOO MUCH RELIANCE ON MODELS
  - JUST BECAUSE A MODEL SAYS IT DOESN'T MAKE IT RIGHT
- THE WEAK LINK IN ALL COST MODELS IS SIZING
  - SOFTCOST'S SIZING FORMULAS ARE JUST AS SUSPECT AS OTHER MODELS
- USER'S DO NOT ALWAYS BELIEVE THE RESULTS
  - EVEN WHEN PERFECTLY CALIBRATED, USER'S DON'T WANT TO BELIEVE THE TRUTH (ESPECIALLY MANAGERS)
- MANY SIMPLE AND MUNDANE PACKAGING CONCEPTS CAN MAKE A MODEL USER FRIENDLY
  - TOO OFTEN, MODELS ARE DEVELOPED FOR CAPABILITY INSTEAD OF USABILITY
  - GOOD HUMAN ENGINEERING GOES A LONG WAY IN MAKING A PRODUCT ACCEPTABLE
- USER'S NEED HELP IN WORKLOADING (ALLOCATING WORKFORCE TO A SCHEDULE) AS MUCH AS THEY NEED HELP IN ESTIMATING
- MOST USER'S REALLY DO NOT KNOW HOW TO ALLOCATE THEIR EFFORTS OR TO ARCHITECT THEIR WORK PLANS
  - MODELING PACKAGES MUST BE DESIGNED TO HELP IN THIS AREA

TOOLS IN DEVELOPMENT

- ASSET IS A PC-BASED SIZING TOOL WHICH EXTENDS THE FUNCTION POINT WORK OF ALBRECHT AND GAFFNEY INTO THE REALMS OF SCIENTIFIC AND REAL-TIME SOFTWARE
  - STATISTICALLY-BASED AND EMPIRICALLY VALIDATED AGAINST A DATA BASE OF 15 PROJECTS

- PMW-II IS A PC/AT MANAGEMENT TOOL SYSTEM BEING DEVELOPED SPECIFICALLY FOR PROJECT MANAGERS
  - AI FRONTEND OPERATING UNDER GEM TO IMPROVE USER-FRIENDLINESS
  - INTEGRATES ASSET, SOFTCOST AND PROJECT WITH A SPREADSHEET FOR "WHAT-IF" ANALYSIS

- SOFTCOST-ADA IS A NEW VERSION OF SOFTCOST-R SPECIFICALLY CALIBRATED FOR ADA DEVELOPMENTS

ALL OF THESE EFFORTS BUILD ON OVER THREE YEARS OF USER EXPERIENCE

IN CONCLUSION

- I'VE DISCUSSED OUR EXPERIENCE WITH SOFTWARE MANAGEMENT TOOLS
- THE MAJOR LESSONS WE HAVE LEARNED INCLUDE:
  - PAY AS MUCH ATTENTION TO PACKAGING AS YOU DO TO FUNCTIONS OR FEATURES
  - MAKE YOUR SYSTEM "MANAGER-FRIENDLY" NOT "PROGRAMMER-FRIENDLY"
  - PROVIDE "WHAT-IF" CAPABILITIES AND A LOT OF SMALL USEFUL TOOLS
  - DON'T ASSUME VENDORS DELIVER WHAT'S ADVERTISED
  - WORRY ABOUT BRIDGING BETWEEN PACKAGES AND DON'T ASSUME IT IS EASILY DONE
  - REALIZE TOOLS MAY ACT AS THE CATALYST FOR ORGANIZATIONAL CHANGE