Original: English

ECONOMIC COMMISSION FOR AFRICA

Second Meeting of the Committee
on Development Information (CODI)

Addis Ababa, Ethiopia
4-7 September 2001

SOFTWARE CRISIS IN SUDAN

By: **Awad Elkrim Mohammed Yousif**

Contribution from: College Of Technological Sciences Information Technology Program

OMDURMAN – SUDAN

JUNE 2001

1.INTRODUCTION

- The S/W crisis is identified by S/W that is late, over budget, and fails to meet the customers system requirements [1] .

Over seventy percent of S/W development organizations develop their S/W through ad hoc and unpredictable methods [2]. These organizations (immature ones) do not have an objective basis for determining S/W cost and schedule or judging S/W quality. Practitioners and their management during the course of the project generally improve S/W development processes. The company does not have any standard practices or if they do, they are not followed.

Each manager manages by using what works best for himself during the last project.

Proved S/W engineering techniques such as in-depth requirements analysis, inspections, reviews, testing, and documentation are reduced or eliminated when the project falls behind in cost, schedule, and/or the customer demands more functionality without a corresponding increase in budget ...[3].

About two millions people are working on about 300.000 S/W projects in the U.S.A at this time. Between one third and two thirds of those projects will exceed their schedule and budget target, before they are delivered. Of the most expensive S/W projects, about half will eventually be cancelled for being out of control. Many more are cancelled in subtle ways .[4].

Studies have shown that for every six new larg-scale systems that are put into operation, two others are cancelled. The average S/W development project overshoots its schedule by half; large projects generally do worse. And some three-quarters of all large systems are 'operating failure', that either do not functions as intended or are not used at all. [5]

Today's major problems with S/W development are not technical problems, but management problems.... [6].

More S/W projects have gone away for lack of calendar time than for all other causes. These are the main issues that we are going to thoroughly discuss, taking Sudan as a case study.

2.S/W CRISIS IN SUDAN

It is common to hear about the "S/W crisis" but not all S/W engineering projects in Sudan are examples of bad management and engineering.

There have been some and very limited success in this field, some quite spectacular.

Without such very limited success, S/W would not have become the pervasive component of our society that it is. Indeed, even it is this limited success that have caused some

demand for S/W in increasingly demanding tasks, leading to unrealistic expectations and the inevitable failures in trying tc meet those expectations.

The rapid rise in demand for S/W combined with the effort required to maintain existing S/W has created a large and rising backlog of S/W applications that need to be written.

This is a tremendous problem that is blocking the progress of many companies, as well as public sector at large: new products cannot be introduced and new services cannot be provided because the S/W that is needed cannot be written quickly enough. This serious situation causes organizations to be constantly looking for solutions.

This chapter will focus on the elements of S/W crisis in Sudan, as well as a sufficient body of knowledge about these elements.

In Sudan;

How many S/W development projects are begun without first defining on paper what the product is going to look like to the customer?

How often does work proceed without having formulated firm specifications?

How many times have programmers been given work to code and test without being given written specifications of what they are supposed to produce?

How many times have specifications changed daily through discussions or rumors so that no one knows the latest decisions that have been and, worse, programmers are proceeding to produce code against outdated assumptions?

In other disciplines these practices would not be tolerated.

In this discussion we are going to analyse the situation in Sudan, taking the elements of the S/W crisis by turn i.e. schedule slip, overbudgeting and the customer system requirements.

This analysis is mainly based on and backed by the survey findings.

3.S/W PROJECTS SCHEDULE

- The first issue concerning S/W crisis is the S/W project schedule slippage.

It is important to accurately estimate the time required to develop a S/W product and to be able to perform schedule/ development effort trade-off. The schedule - related question to be answered is:
How long will the development take?

To address this question the project manager should be able to:

a) Estimate a development schedule, given that he knows (or have an estimate of) the size of his S/W product and how much effort it will take to develop it.
b) Make a trade-off between the length of the development schedule and the effort required to develop the product.

c) Determine whether a schedule given to him is compatible with the size of the proposed product and the required effort estimated for its development.

Normally when planning the development of a new S/W product, trade-off among cost, schedule, and size should be made. For example, for a lower cost, the product size must be reduced. Schedule (the period of time for S/W development) is a key consideration in planning for a S/W development project. It is expected that the effect of varying quality requirements to impact schedule, and for cost. Often, higher quality S/W could be ensured, in point, through more extensive testing.

Sometimes this may increase the development effort and the development time (schedule) over the time required for a S/W product that does not need to be of that quality level...[7].

There are many S/W estimating tools, most of them are far superior to manual estimate methods, in both ease and use and repeatability, and many are also more accurate. In comparison study of fifty manual estimates and fifty estimates produced by tools, two significant results were found... [8]. Manual estimates were wrong more than seventy five percent of the time, and errors were almost on the side of excessive optimism, that is, they significantly underestimated both schedule and costs.

Automated estimating tools came closer to matching historical data that is they predicted higher costs and longer schedule. That is the 'fail safe' mode of S/W estimation.

Of course even automated estimating tools give optimistic results if users exaggerate staff expenses, undertake application complexity, minimize paper work and quality control and ignore learning cures. These are common failures with manual estimates.

This introduction leads us to the status of scheduling estimation (manually which results in the following situation).

During the survey it was found that about seventy five of the S/W projects have gone away for lack of calendar time than for all other causes combined.

Why is this case of disaster so common?
First; the techniques of schedule estimations followed by the S/W houses visited are poorly developed.
Second; these estimating techniques, followed, fallaciously confuse effort with progress, hiding the assumptions that man and month are interchangeable.
Third; because these S/W houses are uncertain of their estimations, S/W managers often lack the courteous suborns required making people wait for a good product.

Fourth; schedule progress is poorly monitored.
Fifth; when schedule slippage is recognized, the natural (and traditional) response is to add manpower, which makes matter worse.

Adding more men then lengthen, not shorten, the schedule, i.e. thus begin a regenerative cycle, which ends in disaster.

It was realized also that, in nearly all the S/W houses visited, project managers do not get the chance to make realistic attempts to create an accurate schedule estimates.

Outside forces, such as marketing or S/W contractors that have been bid too low, can establish schedule ceiling. This can be far from probable schedule of S/W system.

3.1FACTOR SINFLUENCIN GSCHEDULE RISKS:

These are:

a) creeping requirements:
Project requirements slowly increase without a corresponding increase in schedule.

b) Schedule compression:
This was found to be brought about by pressures from marketing, upper management and customers which results in a non-linear increase in S/W cost.

c) Unreliable Budgets:
Budget estimation based on the price necessary to satisfy the market, upper management and/or the customer rather than to satisfy the technical respect.

3.2 THE WAYS FOLLOWED IN DOING ESTIMATED SCHEDULES IN SUDAN

It was also found that estimated schedule is done for every product but,

- No testing for the estimated schedule is done for compatibility with the size and estimated effort - this is the first reason for schedule slippage.

No tracking is performed for schedule growth during development.
This could be the second reason for schedule slippage.
No development and use of an experience DB to aid in schedule estimation.
This could be the third reason for schedule slippage.
The schedule, effort and size are not usually related together.
This is the fourth reason.
No trade-off studies are done between schedule and effort.
This is the fifth reason.
Usually no test for compatibility of schedule effort, and size is done.
This is the sixth reason.

3.3 HOW TO CONTROL A PROJECT ON TIGHT SCHEDULE

From the author's point of view, many project managers were in trouble.
The first step is to have a schedule. Each of a list of events, called milestones, has a list.
Picking the dates is an estimating problem crucially depends on experience.

For picking the milestones, there is only one relevant rule. Milestones must be concrete, specific, measurable events, defined with knife-edge sharpness. Concrete milestone, on the other hand are one hundred percent events.

These concrete milestones demarcate the vague phases of planning, coding, debugging ...[9].

It is more important that milestones be sharp-edged and unambiguous than that they be easily verifiable by the manager.

Sharp milestones are in fact a service to the team. The fuzzy milestone is the harder burden to live with. It is in fact a milestone that grinds down morale, for it deceives one about lost time, until it is irremediable. Chronic schedule slippage is a morale-killer...[10].

3.4 ADDING MANPOWER TO A LATE PROJECT

This is then is the demythologizing of the man-month. The number of months of a project depends upon its sequential constraints. The max. number of men depends upon the number of independent subtasks. From these two quantities one can drive schedules using fewer men and more month (the only risk is the product obsolescence). One cannot, however, get workable schedules using more men and fewer months.

More S/W projects, in Sudan have gone away for lack of calendar time than for all other causes combined....[11].

3.5 CASE STUDY:

In one project, during the survey, it was found that a programming team missing schedules by about one-half, each job was taking approximately twice as long as estimated. The S/W house thought that the team estimating man-hours for several subtasks very carefully did the estimates.

When the slippage pattern appeared, he asked them to keep careful daily logs of time usage. These showed that the estimating errors could be entirely accounted for by the fact that his team was only realizing 50% of the working week as actual programming and debugging time.

Machine down time, higher priority short unrelated jobs, meetings, paper work, sickness, personal time ...etc accounted for the rest.

In short, the estimates made were unrealistic assumptions about the number of technical hours per man-year.
The project manager found that the situation is as follows:
Day-by-day slippage is harder to recognize, to prevent and harder to make up. Yesterday a key man was sick, and a meeting could be held. Today the machines are all down, because of the power cut-off.

Tomorrow, jury duty, family problems, emerging meeting ...etc, the list goes on and on. Each one only postpones some activity by a half-day or a day. And the schedule slips, one day at a time.

How does one tell which slips matter?

There is no substitution for a PERT chart or CP Schedule. Such a network shows who waits for what. It shows who is on the CP.

It also shows how an activity can slip before it moves in the CP.

3.6 CONCLUDING REMARKS

Accuracy in scheduling can sometimes be more important than accuracy in costing. In a product-oriented environment, repricing or amortization can absorb added cost over a large number of sales.

A missed schedule, however, can reduce market impact, create dissatisfied customers, and raise internal costs by creating additional problems during system integration.

When we approach S/W project scheduling, a number of questions must be asked:
- How do we correlate chronological time with human effort?.
- What tasks and parallelism are to be expected?.
- What milestones can be used to show progress?.
-How is effort distributed throughout the S/W engineering process?
-Are scheduling methods available?
-How do we physically represent a schedule and then track progress as the project commences?

For a schedule to be effective, it must possess several major characteristics. It must be:

a) Understandable by those who use it.
b) Sufficiently detailed to provide a basis for measurement and
Control of project progress.
c) Capable of high lighting critical tasks.
d) Flexible easily modified and updated.
e) Based upon reliable time estimates.
f) Conform to available resources; and.
g) Compatible with plans for other projects that shares these same resources.

4. S/W PROJECT IS OVERBUDGET:

4.1 INTRODUCTION:

Budgeting is the process of placing cost figures on the project plan..[12]. The project manager is responsible for determining the cost of the project and allocating the budget to project tasks. Cost is the common denominator for all elements of the project plan, requirements for personnel, computer, travel, office

space, equipment, and so forth can only be compared and cost-trade-off made when these requirements are measured in terms of their monetary value.

Budgets are usually determined using effort (people x time) as the primary cost factor.

However, as observed by Brooks..[13], people and time are not interchangeable. The nonlinear increase in cost with decreasing schedule may result in a very high risk that the project cannot be completed within the budget.

Senior manager wants S/W be delivered on schedule and within cost, a rarity in S/W development projects.
The estimator must estimate the effort (person- hours) and duration (calendar - days) for the project to enable managers to determine important business such as product costs, return on investment, and time to market.

The estimation process is difficult for several reasons:
- Conflicting project goals.
- Lack of a detailed product description

i.e. the estimates are required before the product is well defined....[14].

4.2 MAJOR ISSUES CAUSING S/W PROJECTS OVERBUDGETING

From the survey of those 26 S/W houses it was realized that there are many issues that causes S/W projects over budgeting in Sudan. Some of them are:

a) Nearly all companies do not collect projects cost data on past projects. This made it difficult to use past projects as a basis for estimating future projects.
b) Precession of estimates is rather greater than its accuracy.
c) Estimates are usually based on the number of available programmers.
d) Cost models are not used with any degree of regularity.
e) Most S/W houses reduce their S/W cost estimates when they want to lower their bid.

4.3 WHY ARE S/W COST ESTIMATES IN SUDAN ARE SO UNRELIABLE?

There are many reasons why S/W cost estimates in Sudan are unreliable and as a consequence there are more than 90% of the projects budget overrun - sources of them are:

a) Lack of historical data from past projects on which to base estimates.
b) First-of-a kind system i.e. past history on similar projects does not exist.
c) All S/W houses lack the expertise by the estimators in making estimates.
d) Failure to update estimates, when the project or environment changes.
e) Premature estimate. Estimation must be given before a proper understanding of the project is available or developed.
f) Undue (unfounded) optimism on the part of the developers.
g) Costs are not updated and are based on marketing needs, not system requirements.

h) Always it was difficult to estimate the size and complexities of the S/W project in order to make a realistic cost estimation.
i) Risk factors are usually not assessed or managed.
j) Outside forces, such as marketing or S/W contracts that have been bid too low, can establish a cost ceiling. This can be far from the probable cost of the S/W system.

Forcing a low estimate will drive up the cost of the system. This causes S/W developers to make many false starts trying to deliver against unreliable, low cost goals.

INACCURATE PLANNING

It was found that poorly understood requirements, typically lead to overly optimistic estimates, which then come back to haunt the project leaders when inevitable overruns occur. The top five causes for poor S/W cost estimations all relate to requirements engineering: these are;

a) Frequent requirements change was reported.
b) Missing requirements are also frequently occurring.
c) Insufficient communications with the customers.
d) Poor requirements specifications.
e) Insufficient requirements analysis.

Premature estimates based on limited information or limited thinking can easily be off by a factor of two or more

5. THE S/W PROJECT FAILS TO MEET THE CUSTOMER'S REQUIREMENTS

5.1 INTRODUCTION:

The ultimate deliverable of a S/W development project is a S/W system that meets the customer's needs and expectations. The requirements are an essential step on the path from the product concept to satisfied customers.

"Requirements are a specification of what should be implemented. They are descriptions of how the system should behave, or a system property or attributes. They may be constraints on the development process of the system ..[15]

From this definition, requirements help to:

a) Achieve higher customer satisfaction.
b) Reduce, maintain and support costs.
d) Improve the quality of project's requirement early in the development cycle, which reduces rework and improve productivity.
d) Meet schedule objectives controlling scope, creep and requirements change.

The hardest single part of building a S/W system is deciding precisely what to build.

No other part of the conceptual work is as difficult as establishing the detailed technical requirements. No other part of work so cripples the resulting system if done wrong. No other part is as difficult as to rectify later.....[16].

5.2. REQUIREMENTS RISKS

It was found that in all the projects investigated, that many risk factors for S/W projects resulted from requirements that are inadequate in one or more of the following ways:

a) INCORRECT REQUIREMENTS:
i.e. requirements that do not correctly state user needs and customer expectations.

b) INCOMPLETE REQUIREMENTS:
Requirements that do not stake desired production features or particular aspects of desired product feature.

c) INCONSISTENT REQUIREMENTS:

- Requirements that conflict with other requirements in the same specifications.

d) UNCLEAR REQUIREMENTS:
Requirements that have more than one Semitic interpretation.

e) UNVERIFIABLE REQUIREMENTS:
Requirements for which no finite process exists to verify that the product meets the requirement

f) UNTRACEABLE REQUIREMNTS:
Requirements for which there is no audit trial from requirements to tested code.

g) VOLATILE REQUIREMENTS:
Requirements that are constantly changed, continual additions of new requirements ...[17].

For many of the S/W system developers, requirements are their biggest S/W engineering problem.

In nearly every S/W project that fails to meet performance and cost goals, requirements inadequacies play a major and expensive role in project failure and S/W crisis..[18]

Requirements are the most critical S/W development problems..
Requirements errors are the most costly. The earlier in the development process an error occurs and the later the error is detected, the more expensive it is to be corrected ...[19]

In spite of presumed advances in S/W engineering methodologies and tools support, the requirements problem has not diminished.

- The goal of requirements is to establish and specify precisely what the S/W must do without describing how to do it.

The benefits of good requirements come at a cost.

- The coming part does not survey the literature but, from the survey, seeks to provide an understanding of the underlying issues. It is known that requirements definitions begin with the planning of a series of interviews with the information user personnel, people who will have to make decisions and take actions based on the information supplied by computer information processes. If their approach is carried out with active participation on the part of the users and constant checking back with the users, we could have expected to produce an accurate and reasonably complete statements requirements.

This statement of user information required, which properly structured to reflect the desired processing function, will become the key successful systems development activity.

But in the case of Sudan, it was found that there is a lack of user input, incomplete requirements and most of the time changing requirement are found to be the major reasons why S/W projects do not deliver all of their planned functionality on schedule and within budget, that is why we find in most of the S/W houses visited, problems are getting worse rather than better, and most of the S/W systems are rarely developed on time, within budget, and up to specifications i.e. there is a real S/W crisis.

5.3 CURRENT REQUIREMENTS ENGINEERING PRACTICES IN SUDAN

To calibrate the current requirements engineering practices in Sudan, and identify areas to reinforce, the survey team prepared a set of selected questions, addressed to project

leaders. The team nearly got the same answers, which are considered to be the most that closely describes the way they currently deal with the S/W requirement issues.

- After each answer to those questions by the project leaders, there is the author's, point of view to each question

Q.1: How trained and how experienced are the requirement analysts?

- A.1: They have little experiences and no specific training in developing requirements. They would rather write codes.

- Author: They should have system analysts with extensive experience in collaborating with users, understanding both the application domain and S/W development process.

- Q.2: What techniques are used to analyze the customer's problem?

- A.2: They usually only ask users what they want, write it down roughly, and then build it.

- Author: Talk with users about business need, and their current systems and write a system requirements specifications

Q.3: What approaches are used to identify all specific S/W requirements?

A.3: They often begin with a general understanding, write some code and modify that code until they are done.

Author: The customer representative has to tell them what features and functions the product should contain.

Q.4: How is the project's scope defined, communicated, and used?

A.4: sometimes there is a written project vision statement somewhere.

Author: All the proposed features and requirement changes should be evaluated to see whether they lie within the documented vision and scope.

Q.5: How are the S/W requirements documented?

A.5: they piece together oral history, interview notes, and meeting notes.

Author: Write the requirements in structured natural language at a consistent level of details according to a standard SRS.

Q.6: How are nonfunctional requirements such as S/W quality attributes, elicited and documented

A.6: Most of them do not know even what are "S/W quality attributes".

Author: Certain attributes, such as performance and security requirements, are documented.

Q.7: How are priorities for individual features or requirements established?

A.7: All of them are important. The customer usually tells them what requirements are most important to them.

Author: All requirements should be labeled as high, medium, or low priority by customer consensus.

Q.8: what techniques are used to prepare a partial solution and verify a mutual understanding of the problem?

A.8: None, they just build the system
Author: It is better to create a prototype for both user interface mock-ups and technical proofs-of-concept when appropriate.

Q.9: How are S/W requirements traced back to their origin?

A.9: They are not.

Author: All requirements have an identified origin.

Q.10: How are requirements used as the basis for developing project plans?

A.10: The ship date is usually set before they begin gathering requirements. They cannot change either the schedule or the requirements.

Author: first the product size is estimated from the requirements, and schedules and plans are then based on the estimated effort needed to implement the required functionality. Plans and commitments are to be updated through negotiations if requirements change or the schedule slips.

Q.11: How are the requirement, used as a basis for design?

A.11: If they have documented requirements, they may refer o them during programming

Author: the requirements document has to include interface design and other aspects of the solution they intend to implement.

Q.12: How are changes to the requirements managed?

A.12: Uncontrolled changes creep into the project regularly.

Author: Changes are to be made according to a defined change control process that uses a tool to collect, store, and communicate change requests. The impact of each change has to be evaluated before deciding whether to approve it, and a change control board makes those decisions.

Q.13: How are the requirements used as basis for testing?

A.13: There is no direct connection between requirements and testing.
Author: Always write system test cases against the use cases and functional requirement.
Q.14: how are the system requirements allocated to the S/W portions of the product?
A.14: S/W and H/W engineers discuss which subsystem should perform which function.

Author: Portions of the subsystem requirements have to be allocated to S/W subsystems and traced into specific S/W requirements. Subsystem interfaces should be explicitly defined and documented.

5.4 ANALYSIS OF THE SITUATION

the development team and the customer usually do not agree on the products capabilities and characteristics and eventually the most likely outcome is one of those unpleasant S/W surprises that we would all prefer to avoid it was found that.

a) Nearly all companies do not involve customers representative early and extensively.

b) They do not develop requirements iteratively and incrementally.

c) They do not even represent the requirements in several ways to make sure that every one understands them.

d) Most of them do not assure the requirements completion and correctness with all concerned groups. In addition, when requirements change, often drastically, cost and schedule estimates remain unchanged. This is caused by the reluctance of the project managers to ask the customer for more resources, and the failure to recognize that even minor changes can have a drastic effect on cost and schedule of the project.

e) They also do not control the way requirements changes are made.

The system requirement specifications (SRS) is the basis for all subsequent projects planning, design, coding as well as the foundation for system testing and user documentation. SRS precisely states the functions and capabilities that a S/W system must provide and the constraints that it must respect.

During the survey it was also found that:

a) Customers and marketing dependant do not rely on the SRS to know what product they can expect to be delivered.

b) Most of the project managers do not base their plans and estimates of schedules, effort and resources on the product description that should be contained in the SRS.

c) The development team usually does not rely on the SRS to understand what it is to build.

d) The testing group often does not use the product behavior, which should have been described in the SRS.

In driving test plans, cases, and procedures the S/W maintenance and support staff do not even refer to the SRS to understand what each part of the product is supposed to do.

e) Developers and customers make assumptions most of the time.

f) Some desired functional and nonfunctional requirements are part of the agreement and are expected to appear in the product, are usually not identified in the SRS.

5.5 SOURCE OF REQUIREMENTS ERRORS

It was also found that the source of most requirements errors, when found, is in the failure to adequately accomplish one of those goals, that is:
a) The developer failed to understand what was required from the S/W by the customer, end user, or other parties with a stake in the final product.
b)The developers did not completely and precisely capture the requirements or subsequently communicate the requirements effectively to other parties involved in the development.
c)The developers did not effectively manage the effect of changing requirements or ensure the conformance of down-stream development steps including design, code, integration test, or maintenance of the system.

The above situation resulted from:
a) Lack of user input, incomplete requirements, and changing requirements which are the major reasons why S/W projects do not deliver all their planned functionality on schedule and within budget.

b) Many S/W developers are not conformable and proficient at gathering requirements from customers because practical requirements engineering techniques are not widely known to developers and project participants often do not even agree on what a "requirement" is.
c)Most often a number of changes have to be made in the S/W before it is truly effective in an operating environment.

Sometimes the S/W has been delivered as agreed upon but never used because of changing in management philosophy-time delivery was too much to be accommodated by management.

As a result the most frequently cited situations are:
a) Budget/cost overruns are fairly common in more than 70% of the on-going S/W projects.
b) Calendar overruns occur in nearly 75% of the on-going projects.
S/W, as agreed upon between the users and system development project managers, is extensively reworked before it can be used because of poor communications between users and programmers/analysts.
c) Number of changes have to be made, in most of the cases, to the S/W before it is truly effective in the operating environment add to those difficult numerous situations when S/W has been delivered as agreed upon but never been used because of changes in management philosophy, or S/W projects abandoned at some point because the size overrun.
Or time delay was too much to be accommodated.

This is S/W crisis in reality.
From the author's point of view, the essential difficulties that attend each (in some cases all) of the requirements are:

a) COMPERHENSIVENESS:

Most customers, in our case, do not know what they want. They do not begin with a precise and detailed understanding of what functions belong in the S/W, what the output must be for every possible input, and how one decision will affect another.

b) COMMUNICATIONS:

Usually S/W requirements are difficult to communicate effectively. The conceptual structures of the S/W system for our customers are complex, arbitrary and difficult to visualize.

c) CONTROL:

In all projects investigated, it was found that there are inherent difficulties attend control of S/W development as well. The arbitrary and invisible nature of S/W makes it difficult to anticipate which requirement will be met easily and which will decimates project's budget and schedule if, indeed, they can be fulfilled at all.

d) INSEPARABLE CONCEPTS:

Usually the issues cannot easily be separated and dealt with piecemeal. Arguments approach that does not account for both technical and human concern can have only limited success.

There is a number of practical S/W engineering proven techniques, if used systematically can help to:
a) Achieve higher customer satisfaction.
b)Reduce maintenance and support costs.
c) Improve the quality of the project requirements early in the development cycle, which reduces rework and improve productivity.
d)Meet schedule objectives by controlling creep and requirements changes.

In nearly every S/W project that fails to meet performance and cost goals, requirements inadequacies play a major and expensive role in the project. .....[20]

Also advanced programming tools play major roles in solving some problems associated with S/W crisis, some of them are:
a)Structured programming techniques.
b)Object oriented languages.
c) Structured analysis
d) Numerical analysis.
e)Data flow diagrams.
f)Object oriented design....etc.

REFERENCES

[1] Gibbs,W.W., " software chronic crisis," S/W Engineering project management, 2nd ed., IEEE computer society press, los Alamitos, Calif., 1997.

[2] Zubrow, D., Herbsleb, "process maturity profile of the S/W community 1995 update," NOV.1995.

[3] Paulk, M.C.," the capability Model for S/W in software engineering," IEEE computer society press,Los Alamitos, claif.,1997, P.P 427-438.

[4] MC Connell, steve,"softwre project survival guide," Microsoft press, Redmond, washington,1998.

[5] Gibbs, W.." Software charonic Crisis," from scientific American, 1998 p.86 - 95.

[6] Brooks F.P, "Report on the Defense science Board Task force On Military software," office of the undersecretary of Defense for Acquisition, Dept. of defense, Washington D.C., Sept, 1991.

[7] Gaffney, Jr., "How to Estimate software project schedules," The software measurement Guide book, Thomson computer press, Boston, MA. 1995.

[8] Caper's J., " Software Estimating rules of thumbs," Inst. IEEE, vol.29, no.3, March 1996, P.P 116-118.

[9] Reynoids, C.H.," What is wrong with computer programming Management? ," on the management of computer program. Ed., 1991, p.p. 35-42.

[10] King, W.R., and Wilson, T.A., " Subjective Time Estimates in critical path planning - a preliminary analysis, " Management science 13,1987, P. 307 -320.

[11] Brooks, F.P., " The Mythical Man-Month," Addison-wesley publishing computer Inc., 1975, P.P 12-26., 88- 94.

[12] Richard, H., Thayer, "Software Risk Management," software Engineering project Management, IEEE computer society , 1997.

[13] Brooks, F.P., p.202.

[14] Brooks, F.P. P. 218.

[15] Sommerville, I., and Sawyer, P., "Requirements Engineering. A good practice Guide," chichester , England: John wley and sons, 1997.

[16] Brooks, F., " No silver Bullet: Essence and Accidents of Software Engineering, " computer, Apr. 1987, P.P. 10-19.

[17] Thayer, Richard H., " Software Risk management ," S/W engineering project Management, IEEE computer society, Los Alamitos california, 1997, P.P 1980.

[18] Alford, M., and Lawson, J., "Software Requirements Engineering Methodology (development)," RADC-TR-79-168, US. Air Force Rome Air Development center, June 1997.

[19] Boehm, B., "Software Engineering Economics," Prentice - hall, Englewood Cliffs, N.J.,1987.

[20] Alford, M., and Lawson, "Software Requirements engineering Methods Development ,"
RADC-TR-79-168, U.S. Air force Rome Air Development center, June 1979.