2001

Factors affecting the success of Information Systems projects

Iain Martin
Edith Cowan University

USE OF THESIS

EDITH COWAN
UNIVERSITY

PERTH WESTERN AUSTRALIA

BACHELOR OF BUSINESS
(HONOURS)

THESIS

Submission Date: November 2001

FACTORS AFFECTING THE SUCCESS OF INFORMATION SYSTEMS PROJECTS:

NAME:	Iain Martin
SUPERVISOR:	Dr Wally Smith
FACULTY:	Business and Public Administration
SCHOOL:	Management Information Systems.

ABSTRACT.

This thesis looks at how accurately project managers can prepredict what problems may arise during an Information Systems project they are about to commence based on their experience of previous projects. The second part investigates what actions a project manager is likely to take to expedite a troubled project. Both of these questions are investigated and it is found that while project managers can confidently predict which problems could arise, and how to take actions to rectify such actions when they occur, these problems still arise.

I certify that this thesis does not, to the best of my knowledge and belief:

(i) incorporate without acknowledgment any material previously submitted for a degree or diploma in any institution of higher education;

(ii) contain any material previously published or written by another person except where due reference is made in the text; or

(iii) contain any defamatory material.

[redacted]

Iain Martin

Acknowledgment.

I would like to thank Dr. Wally Smith, my supervisor, Associate Professor Bill Hutchinson the marker, and the entire staff at the School of Information Systems at Edith Cowan University. For the last four years I have been here, their patience and time has certainly been welcome. Thanks also to Coca-Cola Amatal, whom without their Diet Coke ® I would have never completed this thesis, it kept me awake all those long nights.

The road has been long, starting working also effects the time taken to write such papers I find.

To anyone whom I have omitted my thanks too, I thank you also.

TABLE OF CONTENTS.

LIST OF FIGURES.

LIST OF TABLES.

LIST OF EQUATIONS.

1 INTRODUCTION

Most Information Systems / Information Technology projects run over budget, by as much 180% (Johnson, 1995) over the initial estimates. Reasons listed for these failures are; the nature of the people in the industry (Brooks Jr, 1975; Yourdon, 1997); the inherent guesses used for estimations; the complexity of the projects often being underestimated, and finally, the requirements of the system being unclear or incorrectly specified.

The cost of these failed or ill-managed projects is astronomical. It is said billions of dollars a year are presently being wasted in these type of projects (King, 1997). Information Technology projects can seem to be 'money sucking black holes' to many organisations. Ultimately the prevention of these types of projects lies outside the scope of this study; this study will however attempt to map the thoughts of various project managers to see how they work. Do they learn from their past mistakes and failures, can they foresee potential problems before they occur, and most importantly, when the problems occur, what actions do they take to correct them?

This study assumes that the tools used for Information Systems are computers, it will examine the management of projects that develop Information Technology solutions.

1.1. Major Questions to be Addressed.

The study is broken down into to parts; Part A. (the planning section), and Part B. (the expediting section). The questions this study addresses are as follows.

Why do problems re-occur with Information Systems / Information Technology projects? This question was addressed for both project planning and expediting.

1.1.1. Part A. Planning.

What factors can project managers readily identify as being the most important when they are evaluating or planning Information Systems project with large software composition?

How do the factors identified by the project managers compare with the factors identified by the existing academic literature on this topic?

1.1.2. Part B. Expediting.

What corrective actions do project managers readily adopt into their projects when the need to expedite a late or troubled project is required? How do the corrective actions identified by the project managers compare with the factors identified by the existing academic literature for this topic.

2 REVIEW OF LITERATURE

2.1 FAILURE

What is failure?

The Collins English Dictionary defines it as follows:

"*failure n.*

- The act or an instance of failing.
- A person or thing that is unsuccessful or disappointing: the evening was a failure.
- Non-performance of something required or expected: failure to attend will be punished.
- Cessation of normal operations: breakdown.
- An insufficiency or shortage: a crop failure.
- A decline or loss, as in health or strength.
- The fact of not reaching the required standard in an examination, test, course, etc.
- The act or process of becoming bankrupt or the state of being bankrupt."

- Source (Hanks & Wilkes, 1986, Pages 545 - 546)

2.2 *WHAT IS FAILURE?*

The word 'Failure' is usually associated with bad things as shown above in the definition provided. What it fails to mention is how beneficial failure is to us. Jung (1933) stated that psychotherapists learn little if anything from their successes, but their failures are priceless, as they may force them to change their methods and views. However, Ackoff (1994) points out that a mistake [a failure] is an indicator of a gap in one's knowledge. Fortune & Peters (1995) write an entire book on learning from your failures, they mention that failure goes beyond the simple definition of something not meeting expectations, or going wrong. Several types or categories of failures have been identified.

- **Type 1. Objectives not met.**

  This type of failure occurs when the aims / objectives of the user, designer or sponsor of some things expectations are not fully met. i.e. A toll bridge which carries barely any traffic.

- **Type 2. Undesirable side effects.**

  This type of failure occurs when the aims / objects of the user, designer or sponsor of some things expectations are met, but some undesirable side effects have presented themselves. Typical examples of this can be medical drugs.

- **Type 3. Designed Failures.**

  This type of failure occurs when certain conditions are met and aims typically to protect the user, designer or sponsor. An electrical fuse is a good example of a type 3 failure, when too much power attempts to enter the device the fuse 'fails' thus protecting the device.

- **Type 4. Inappropriate objectives**

  This type of failure occurs when the objectives that were set are met, with no adverse effects, but there is no longer any demand, need or market for it.

Lyytinen & Robey (1999) write about learning to fail in Information Systems development, in which they argue that organisations fail to learn from their experiences in systems development due to the limits of organisational intelligence, disincentives for learning and organisational barriers.

2.3 *PROJECT SUCCESS AND FAILURE.*

It is difficult to define what constitutes project failure. Each professional's definition of it differs. Yourdon (1997) writes that software projects are often 50 to 100 percent over budget, and on average likely to be 6 12 months behind schedule. In an article by Johnson (1995) he reports that out of 500 projects surveyed, none were within schedule or budget, and 40% were likely to be cancelled. Consultants KPMG defined the difference between a failure and a success at 30% overrun of estimated (Cole, 1995), where as Glass (1998) would not regard a project as a failure unless it had overrun estimates by 100%.

A project may also be a 'failure' if it does not meet the users requirements (Weinberg, 1991) regardless of if it was completed within budget or not. This alternate view is that the above over-runs may not be classed as failures if the program meets or exceeds the users needs. Boehm (2000) writes about the fact that a terminated project does not automatically mean a failed project; projects can be terminated for a variety of reasons, even if they were not classed as a failed project.

There is no absolute way to determine if a project is success or failure (Belassi & Tukel, 1996). The various stakeholders, users, project team and management might all view the same project from various differing perspectives, and thus can arrive at a different conclusion about the success of a project. For example, an article in 'Computing Canada' (Denies, 1988) reports on a study by a global communications consultancy that removes the blame for failed Information Technology from the Information Technology departments and places the blame directly with the top decision makers who fail to adopt the recognised best practices.

A study by the Standish Group collected information from 365 surveys that represented over 8,380 Information Technology projects conducted within the previous year (Johnson, 1995). The study showed that in excess of 31% of projects started would be cancelled before completion, and of those remaining, 53% of them will run over their initial estimates by 189%. The study also breaks the types of "resolutions" achieved into three categories:

- **Resolution Type 1** – A successful project, within time and budget
- **Resolution Type 2** – A challenged project, project is completed, but either
  - Over budget
  - Reduced functionality
  - Fewer Features

- **Resolution Type 3** – An Impaired project, The project was cancelled.

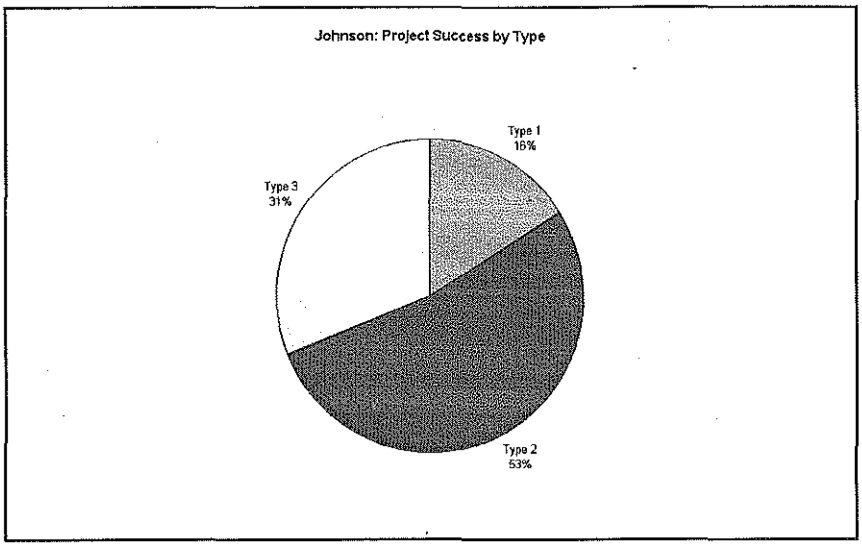


FIGURE 2-1. JOHNSON'S SUCCESS BY COMPANY TYPE.
- Source (Johnson, 1995, Page 5)

Figure 2-1 was reproduced from (Johnson, 1995) and graphically illustrates the findings of the Standish Group report. He (Johnson, 1995) further breaks the success / fail rate down into small, medium and large businesses, they company size was dependant on annual revenue ***Figure 2-2***.

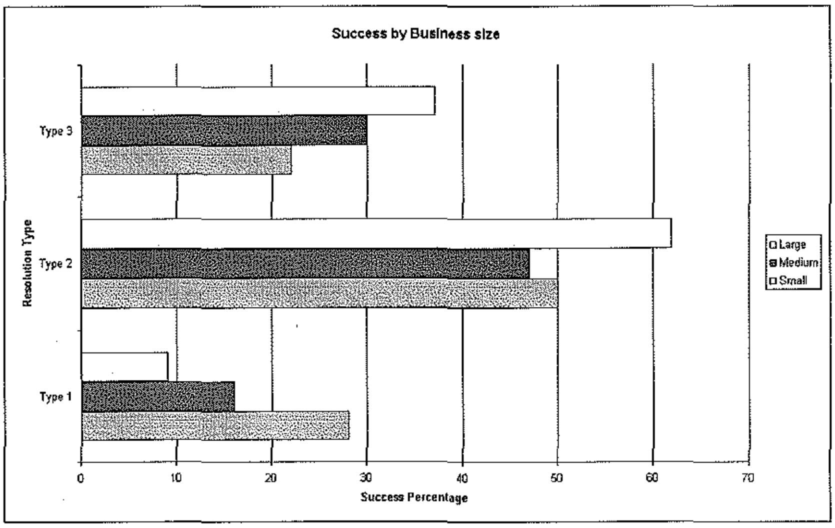


FIGURE 2-2. TYPE RESULTS BY BUSINESS SIZE.

Source (Johnson, 1995, Page 5)

As can be seen above in ***Figure 2-2***, the larger the company, the more likely they are to have their projects become more difficult to manage and increase the chance of project failure. From this data it shows that small companies have a greater chance of succeeding in projects than do medium and large organisations.

2.4 THE ROOTS OF PROJECT FAILURE

Ed Yourdon (1997) introduced the notion of a 'death march' project:

> "*I define a death march project as one whose 'project parameters' exceed the norm by at least 50 percent*".
>
> - (Yourdon, 1997, p2)

Yourdon likens such software projects to the 'death marches' of history. He implies that casualties will happen because of the urgency of the project, short time frame, and a perception by management that it will be finished on time. A combination of any of the below can constitute the creation of a 'death march' project:

- The schedule has been reduced to less than 50% of original plan.
- The staff has been reduced to 50% or less of the original plan.
- The budget has been cut, forcing shortcuts and cost saving.
- Functionality changes – this is common as the system needs to do more than it was designed to do.

In his classic analysis of Information Technology projects, Brooks (1975) states that most projects fail for 'lack of calendar time' than for all other reasons combined. The causes he cites are:

- Techniques of estimation are poorly developed.
- They are often optimistic 'all will go well'.
- Confusion over effort and progress – The assumption that men and months are interchangeable.
- Managers are uncertain about their estimates.
- Progress along the schedule is poorly monitored.
- When the schedule falls behind, more manpower is added, like feeding oxygen to a fire.

2.5 ESTIMATIONS AND PROBLEMS THAT COME WITH THEM.

The main problem within the software industry, some 40 odd years after its conception, is not building or writing the software. Echoing Brooks, Glass (Glass, 1998) argues that the main problems arise with the estimation of how long it will take to do the project.

Although the main purpose of Information Technology in most organisations is the storage of historical data, the software industry has been very poor at documenting its own history. The industry, generally, has not made a conscientious effort to keep records of past project costs and duration (Glass, 1998). Another major distinction between industries is that whereas you can visually see and assess progress in an engineering project to build. The progress of a construction of a bridge is much easier to gauge than that of a software product, due to the tangible nature of the bridge, the progress of the project can be observed easier due to the physical presence of a bridge, software has no such tangible presence.

Yourdon (1997) writes in these days there are many aides av ilable to the project manager to better assist their estimations such as:

- Commercial Estimating Tools - Computer packages that assist the manager to make better estimates, the problem with this, as with most computer programs is that the quality of the end result depends on the quality of the inputs. In other words garbage in, garbage out (Lam, 1998).

- Systems dynamics models – Mathematical models that have been developed to explore the relationship between entities involved in the project, for example the various COCOMO models (Boehm, 1981) and COCOMO 2 (Boehm et al., 1996).

- The Software Engineering Institute has done extensive research into this area, and has even published checklists and guidelines on it (Park, 1996).

Despite the fact that ample resources are available to aid in estimations, other problems arise. Thomsett (1996) wrote about what he termed "negotiation games". That is, games that managers and project managers often engage in. He mentions a number of these games that are described below:

- ***Doubling and add some***

  This technique involves the project manager making an initial estimate and then doubling it and adding say an additional 10% to that duration.

- ***Reverse doubling***

  Management uses this technique typically, when the project manager brings his estimates to his superior, the first thing the manager does is half the estimate as he recalls his project management days when he used *the Doubling and add some* method.

- ***"Guess the number I'm thinking of"***

  This is when the senior person already has an "acceptable" figure in mind i.e. the figure he told his boss. They ask their subordinate for an estimate, and then tell them its unacceptable and to give a more realistic estimate. This continues until the figure matches the duration he initially quoted, as the subordinate gave the estimate, he is held to it.

- ***Double dummy spit***

  This is when a project manager brings in his initial estimates to management and the manager goes into a hysterical fit about the length of the project, the first dummy spit. The project manager scurries away and reviews his estimates and returns with the revised estimations. The manager will have another hysterical fit at this time. The managers' reason for this is to create fear in the project manager so that they will agree with whatever they are told to prevent the fits.

- ***Spanish inquisition***

  This technique is when a project manager walks into a meeting with senior management and is asked on the spot for an estimation, often without knowledge of what he is estimating about. All of the managers wait for the reply from the project manager.

- **Gotcha**

  Project Managers typically do this technique for revenge, they accept totally unrealistic estimates. The theory being the company will be forced to face the reality of software project management, often after lots of money has been spent already.

- ***Chinese water torture***

  Instead of delivering the bad news of a blown project all at once, the manager delivers the news in small snippets, "Component *X* was only 4 days late" this does not sound too bad until you look at the entire project. It is like the dripping water in the Chinese water torture, drip, drip, and drip. No single drip (piece of bad news) will kill you, but the cumulative effect can be fatal.

2.6 PEOPLE IN THE PROJECT

(Programmer behavior in software projects)

If a post-mortem examination was done on most unsuccessful software projects it would find that most of them failed due to people related matters (Feibus, 1998).

The productivity of a programmer can also have a great impact on any project. Bryan (1994) tells of a study that analysed the productivity of a wide range of programmers. Bryan found that the top 27% of programmers did 78% of the work. These findings are close to the normative predictions of the Pareto theory, that 80% of the total output is produced by only 20% of the relevant population.

In a study by Sackman, Erikson, and Grant as referred to by (Brooks, 1975) the performance within a group of experienced programmers were measured. Within this small group, the productivity of individual programmers varied so widely that the ratio for productivity averaged 10:1 and on program speed and space taken up the ratio averaged 5:1.

In another study (Stackman, 1968) it is reported of productivity ratios of up to 25:1 for programming tasks, and up to 28:1 for debugging programs. All programmers involved in the study were familiar with that application development area. He further goes on to say that he would expect to find productivity ratios of 5:1 on most software projects.

The before mentioned study (Bryan, 1994) conducted on nearly 200 programmers over a period of 12 years brought forth similar results as Stackman (1968). The number of programmers working on the CP-6 project ranged from a low of 15, to a peek of 150. The data of how much work each programmer did was recorded on a database as part of the management of the project. The project was a 4.2 million-line program that was being developed; a productivity variation of 200 to1 separated the top programmer from the worst one.

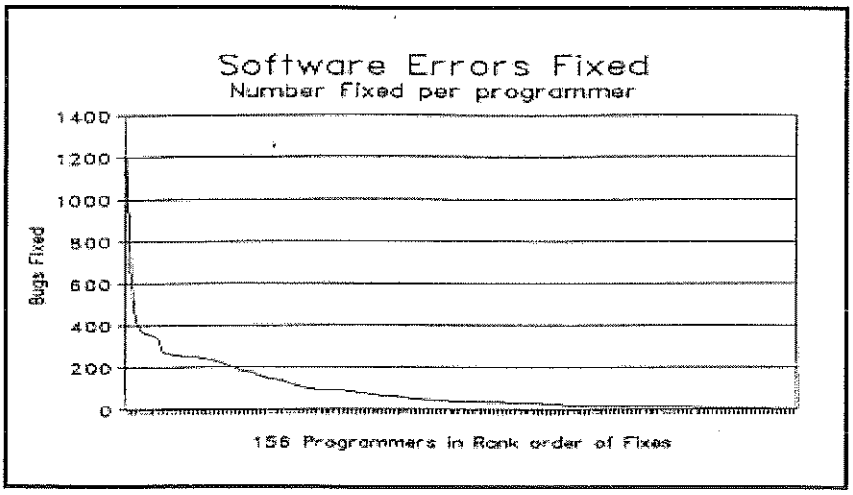


FIGURE 2-3. SOFTWARE BUGS FIXED BY PROGRAMMER.
-Source (Bryan, 1994,Page 350)

Figure 2-3 and ***Figure 2-4*** display respectively the number of bugs (STARs) fixed by the top 156 programmers, 36 programmers were not recorded as they had below 5 fixes. There is very little noticeable difference in the two figures, the general shape remains constant. Software "bugs" make up over half of the STARs reported (Bryan, 1994). Thayer (1997) states that STAR's is an acronym for Software Technology for Adaptive, Reliable Systems. It cannot be said if the meaning here is a different meaning than used in Bryan (1994).

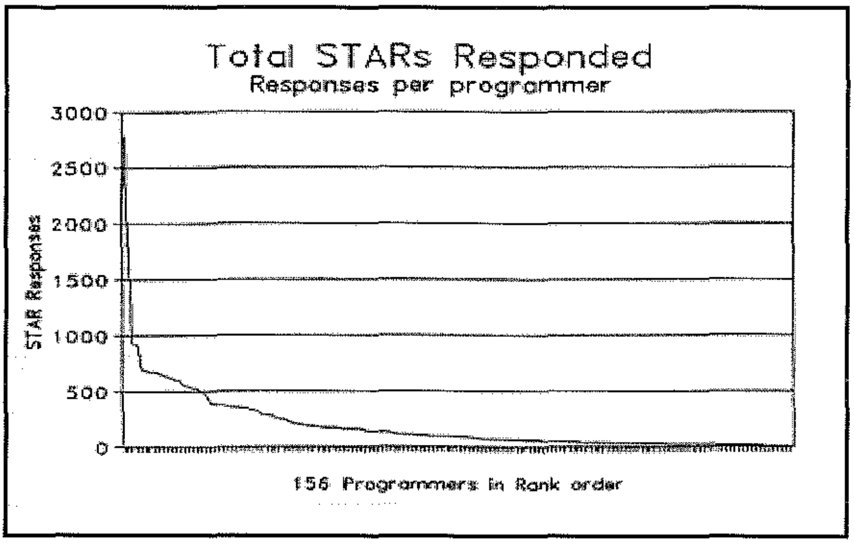


FIGURE 2-4. TOTAL STARS ANSWERED BY EACH PROGRAMMER.
-Source (Bryan, 1994,Page 350)

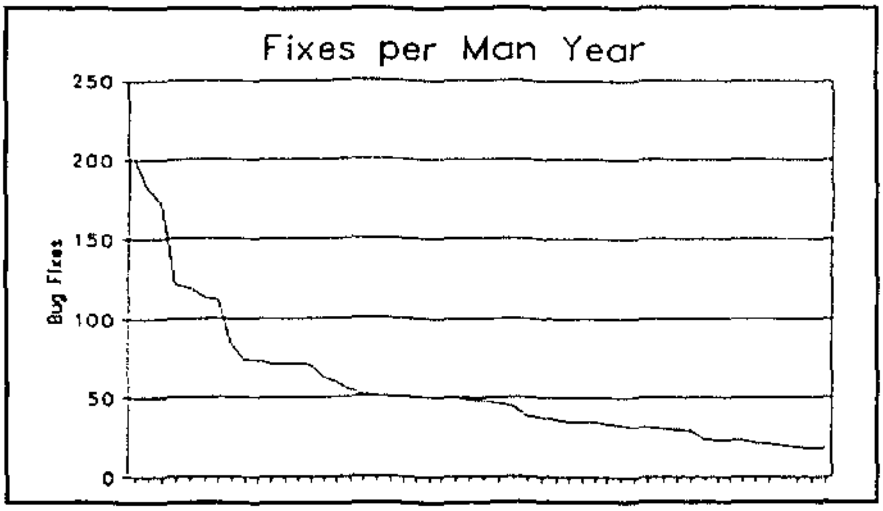


FIGURE 2-5. SOFTWARE BUG FIXING RATE BY TOP 51 PROGRAMERS.
-Source (Bryan, 1994,Page 352)

Figure 2-5 and ***Figure 2-6*** are adjusted figures, dependent on years employed, and proportion of development responsibilities, as opposed to management of people, teams etc. The gross time used over 12 years equalled 320 work-years, but was tracked at 217 work-years; it is doubtful if this number takes into account overtime. ***Figure 2-6***. shows the productivity in bug fixing per Man / Work year. They as a group fixed 11,151 bugs, which is 78% of reported bugs. The top programmer fixed 8% of the bugs himself, a remarkable feat, these were not just easy problems as one might expect. Rather he fixed some quite difficult problems ranging all over the project, most of the bugs he fixed he had not coded, nor had any knowledge of that component prior to debugging the code.

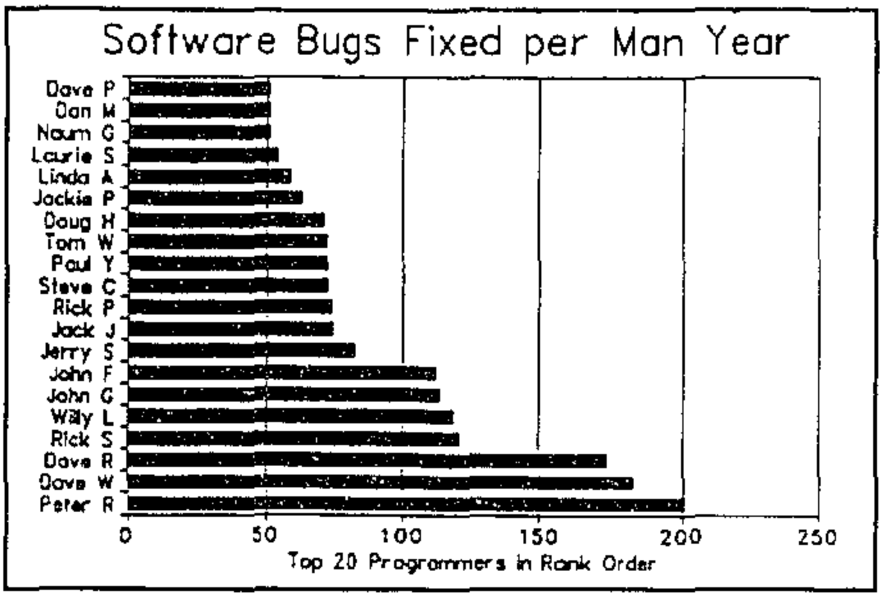
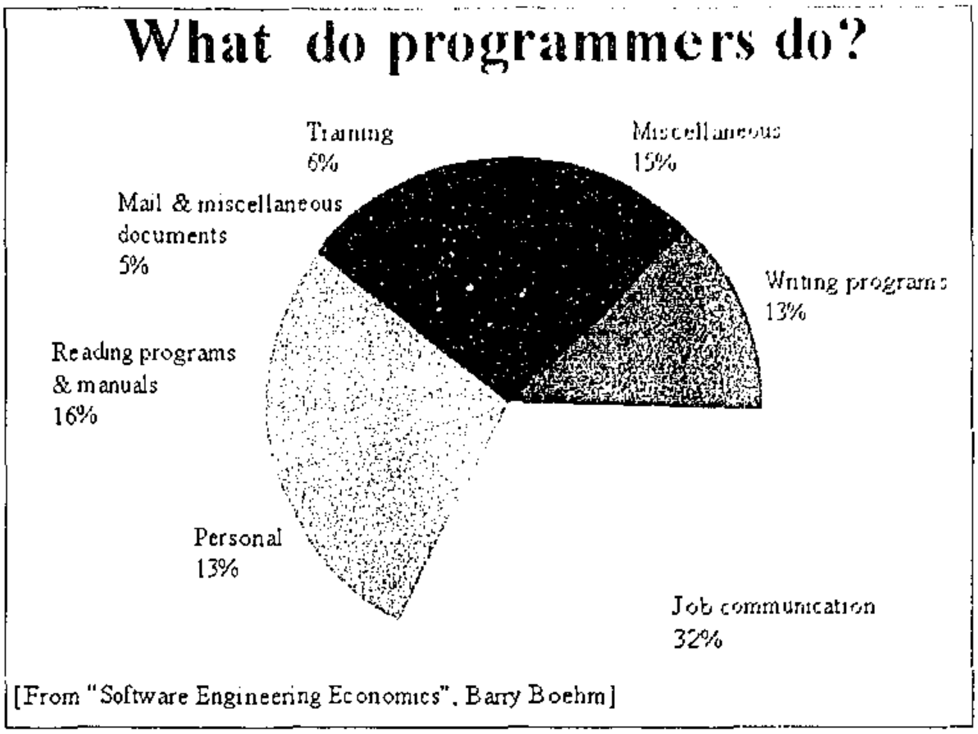


FIGURE 2-6. FIX PRODUCTIVITY BY TOP 20 PROGRAMMERS.
-Source (Bryan, 1994,Page 352)

Figure 2-7 graphically illustrates how a programmer's time is taken up. As can be seen a relatively small part of it (13 %) is actually spent programming. Most of their time is taken up with communications, reading and other learning activities.



What do programmers do?

[From "Software Engineering Economics", Barry Boehm]

FIGURE 2-7. WHAT TAKES UP A PROGRAMMER'S TIME
–Source (Boehm, 1981, page 341; Salt, 1999)

Brooks (1975) says that computer programs are optimists. He points to the relatively young age of coders, and argues "*the young are always optimists*". That combined with the programmers attitude and optimistic comments like "I just found the last bug" is sufficient to satisfy Brook's that they were optimists, and that, being optimists, it is likely that their estimates will also be optimistic. Yourdon (1997) mentions similar views as Brooks.

In the recruitment of programmers or technical staff, Zawacki (1985) reports on his experiences as a human resources person recruiting Information Technology people. He reports that mangers have great confidence in their opinions if the programmer which are formed on a short interview. The interviewer typically has a stereotype in mind for the job and attempts to match a candidate with that image.

While the previous section could be regarded as the emphasis for project failures being the responsibility of the computer programmer, that is not necessarily true. While it is true that some software components can be difficult to design and write and thus difficult to predict the required time for that component. There is no doubt that often the software can be the delaying factor, it is not always so, the final product can only be as good as the submitted design.

2.6.1.1

2.7 WHY DO PROGRAMMERS DO THESE DEATH MARCH PROJECTS?

There are a variety of reasons why project managers and programmers get involved in these 'Death March' projects. Yourdon (1997) identifies the top reasons he has found:

- ***High risk, but also high rewards.***

  These projects may have a large risk associated with them from day one, but so are the rewards if they succeed. A good example of these would be Internet start-up companies, they pay their employees in stock options, and if the project is successful the shares are valuable. This worked well for Microsoft after all, which has one of the highest ratios of millionaires per workforce in the world (Cusumano & Selby, 1996).

- ***Mt Everest syndrome.***

  This category is for those people who desire challenges and go out of their way to attempt challenges. A project in failure could attract a type of person who really believes they can win the challenge.

- ***Youth naïveté and optimism.***

  This category is for those people who are still in their early to mid twenties, single, free of commitments. They still have their naivety and optimism, "can't everyone work 70-hour weeks and write perfect code at the same time".

- ***This or unemployment.***

  It was a choice of this project or to look for a new job, while this may not bother the young programmers who are free of children, mortgages etc that believe they can walk into a new job tomorrow. For the older programmers who have these burdens and perhaps outdated programming experience this can be a significant reason for going to the troubled project.

- ***For future advancement.***

  In some way your participation or the ultimate success of the project may influence your future career path, i.e. promotions, career pathway etc.

- ***Escape from bureaucracy.***

  Some organisations are notorious for their paperwork and strict rules that govern how a person does specific tasks, in troubled projects these formalities are often overlooked so as the people can work more efficiently. Some people may sign on to a project for such a reason.

- ***Revenge.***

  This category may seem a little strange as a motivating reason to do a troubled project; it is not unknown for battles to be fought amongst senior management. An easy way to make one of your "opponents" look bad is to sabotage their projects, i.e. insert incompetent staff into their project.

2.8 COMMUNICATION BETWEEN PEOPLE AND TEAMS

A problem common to all projects is time; Brooks (1975) addresses this issue. He developed what is known as 'Brooks' Law', which has been deliberately over simplified as:

> *"Adding manpower to a late software project makes it late"*
>
> - (Brooks, 1975,pp 25)

It must be said, however, Brooks' Law, (Brooks Jr, 1975) is true only of the tasks that involve communication between the various programmers. Any task that is independent (no dependent communications) would not fall into this category. Time is also needed to train the new people in work practices, and new staff will often modify work previously completed by other people. If this were to happen, a system to keep track of software changes etc like the one as described in Microsoft Secrets (Cusumano & Selby, 1996; Cusumano & Selby, 1997) would be required.

Some tasks can be completed faster by adding manpower; those are the tasks that require no co-ordination of activities between team members or components. Others duration can be increased, assuming they have a dependency upon each other. The final category of tasks are ones that will take *x* time regardless of how many people are assigned, an example would be cooking bread, It will take 3 hours regardless of how many ovens are used (Brooks, 1975).

A recent estimate reported that in the United States, between 70 to 80 percent of all businesses were using team concepts (Chaney & Lyden, 2000). Communication between members of any team is essential, Ambaye (1995) identifies the major cause of failure of Information Technology teams is the lack of communication between team members. As shown previously in ***Figure 2-7*** a large proportion of their time is taken up in communication activities.

While in the past, teams for software production may have been in the same location; presently with the use of Internet technologies this is no longer necessarily true. The Virtual Organisation, or the Virtual Team are taking shape and doing a large percentage of the software development. India is very prominent in this field; its Information Technology industry has grown from $3.9 billion in 1988 – 1989 to $200 billion in 1998 – 1999 due it outsourcing many American companies software development (Chand, 2000). Communication between virtual team members is vital as it is the only way the can succeed (Chase, 1999), this is doubly so with rapidly developed software (Bullinger, Warschat, & Fischer, 2000).

2.8.1 Communication Models.

Mantei (1981) reports that there are two main recognised group structures for managing programming projects, these are Bakers chief programmers team (Baker, 1972) and Weinberg's ego-less team (Weinberg, 1971). A third, a hybrid of the two was also found (Curtin University, 1999) and included.

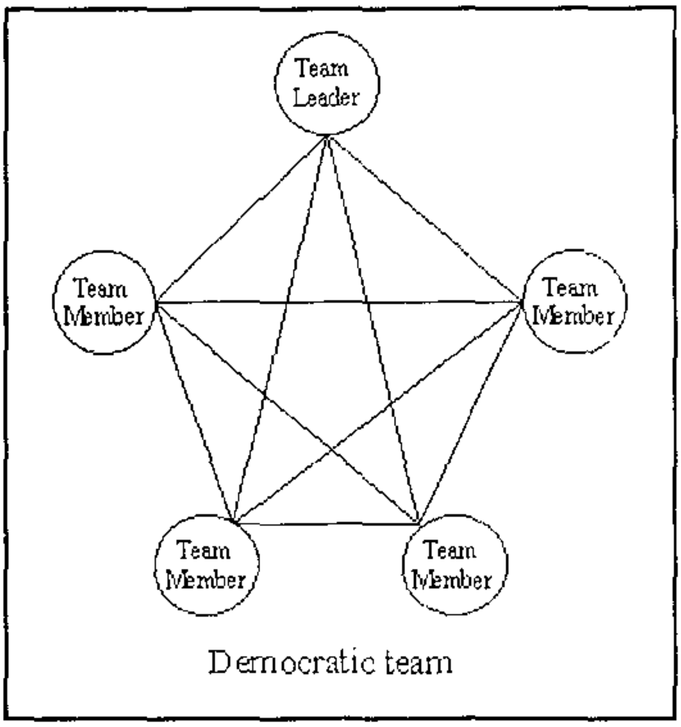


Figure 2-8. The Democratic Team.
- Source (Bennatan, 1995, page 74)

The above ***Figure 2-8*** uses a different name than Wienberg (1971) does, in this model each person in the team has the same say and input as each other. While there is a team leader, they foster open communications between all team members, the use of e-mail and group meetings is paramount in this model. One of the main problems with model is that team members often consume vast amount of time arguing over trivial matters (Mall, 1998), and personality conflicts.

$$\boxed{\max imum\_number\_of\_channels = \frac{n(n-1)}{2}}$$

EQUATION 1, MAXIMUM COMMUNICATION CHANNELS.

The above formula can be used to compute the maximum number of possible channels which could be needed for a team of any given size to communicate effectively, the assumption is that only one channel (medium) is used to communicate to each individual team member. The above example (***Figure 2-8***) uses a maximum of 10 channels, if they same figure had 15 Nodes, up to 105 channels could be needed. Clearly the more people / nodes the greater the time spent in reading / writing e-mail, meetings and the like, small teams are more productive for this and other reasons. It is stated that teams consisting of less than 10 people are much more efficient for programming projects (Brooks, 1975; Yourdon, 1997)

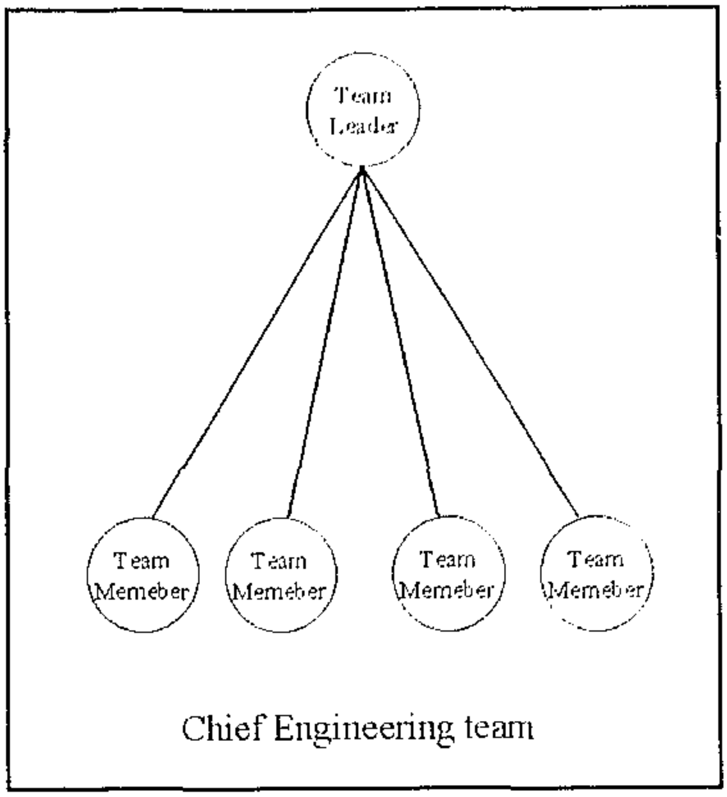


FIGURE 2-9. CHIEF ENGINEERING TEAM
- Source (Bennatan, 1995, Page 74)

The Chief Engineering Team model is named the same by both Baker (1972) and Bennatan (1995). The above diagram (***Figure 2-9***) illustrates the Chief Engineering Team model, there is far less communication in this model than in the Ego-less / Democratic team, in this instance the team leader assigns each team member what tasks they should perform. There is little feedback to the team leader, and as shown, little if any formal communication between team members, although no formal team communications occur, shoptalk may be done informally between team members at breaks, or out of work hours. In reality, the team members would communicate with each other as they need to.

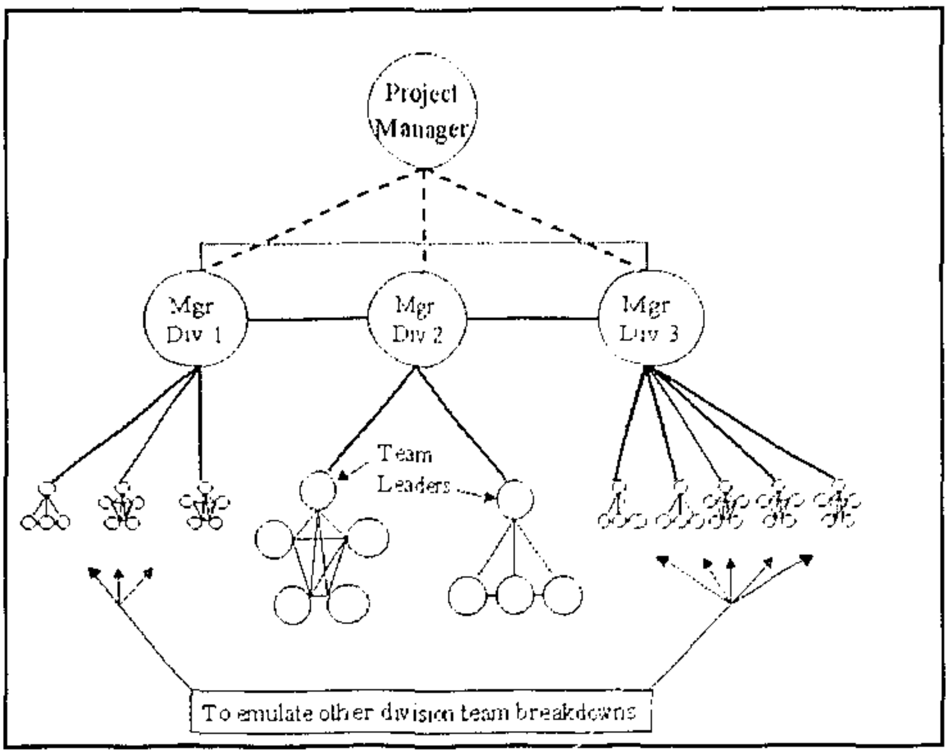


FIGURE 2-10. HYBRID MODEL OF TEAM COMMUNICATIONS.

The third model is a hybrid of both the previous two models, this approach is similar to that used in Microsoft (Cusumano & Selby, 1996). This model is effective when used on large projects, as systems are broken down to their components and ultimately small teams get assigned these components to build, they are free to use whatever team structure they please, as long as they achieve results.

Communication between theses divisions or teams may also be necessary; ***Figure 2-11*** below is an illustration of how teams could work in Microsoft. The structure at the top of the figure is correct, as reported by (Cusumano & Selby, 1996), artistic license was used on the lower levels. In it, the development group project managers communicate with each other so as to check how their latest improvements will affect each other, i.e. will the new update of Office 2000 have adverse effects with any of the Operating Systems. While not shown, some communication may also take place within groups, for instance all the project team leaders of the Office suite (Word, Excel, PowerPoint etc) may communicate regularly as integration of these products is essential.

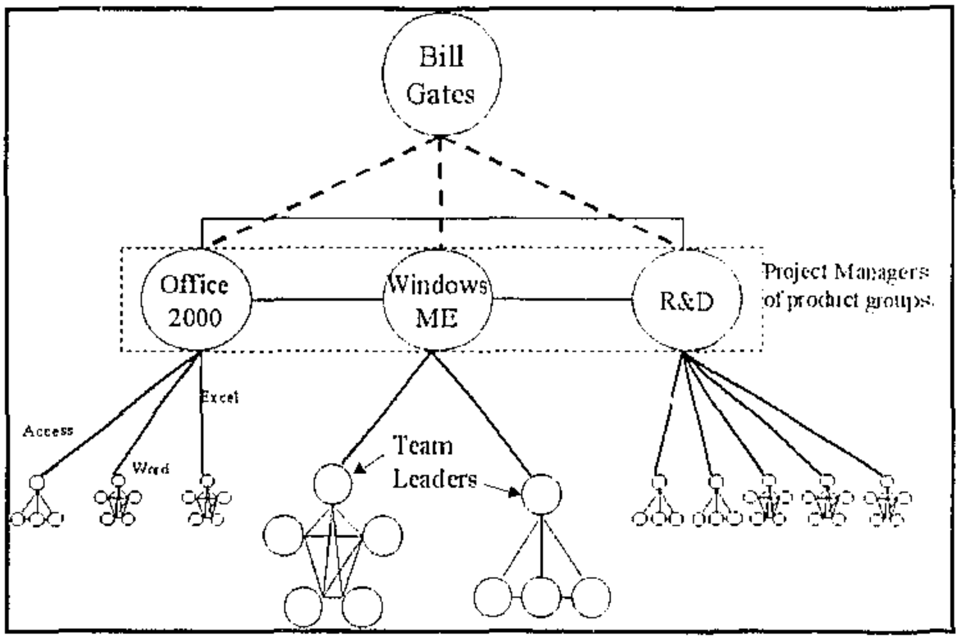


FIGURE 2-11. AN EXAMPLE OF HOW TEAMS COULD WORK IN MICROSOFT.

2.9 FACTORS UNDERLYING PROJECT FAILURE

Robert Glass (1998) describes some factors that contribute to projects running over budget / schedule:

- The size of the project - the greater the size of the project the more likely that it will have delays.
- Project failures normally result from not one single cause Normally it's a combination of several causes.
- Sales talk – what the salesman sold you is not what the system can do, the salesman promises anything for the sale.
- Technology – The technology was either new, non-existent or not understood by the project teams.
- Performance – Real-time systems are often too slow to do live transactions once built.

TABLE 1, JONES SUCCESSFUL VS UNSUCCESSFUL FACTORS

Unsuccessful project technologies	**Successful project technologies**
No Historical software measurement data	Accurate software measurement
Failure to use automated estimating tools	Early usage of estimating tools
Failure to use automated planning tools	Continuous usage of planning tools
Failure to monitor progress of milestones	Formal progress reporting
Failure to use design reviews	Formal design reviews
Failure to use code insepections	Formal code inspections
Generalists used for critical tasks: Quality assurance Testing Planning Estimating	Specialists used for critical tasks: Quality assurance Testing Planning Estimating
Inadequate design and specifications	Automated design and specifications
Failure to use formal configuration control	Automated configuration control
More than 30% creep in user requirements	Less then 10% creep in user requirements

- Source (Jones, 1995, Page 3)

Table 1 identifies the top 10 technology factors that can be associated with successful and unsuccessful software projects (Jones, 1995). This was taken from 60 risk factors that were identified by Casper (1994) and can affect the affect the overall outcome of a software project.

2.10 FLOWERS CRITICAL FACTORS.

Stephen Flowers has developed what he calls his Critical Failure Factors (Flowers, 1996). It consists of 17 factors, broken down into three phases of the project; organisational context, the management of the project and the conduct of the project i.e. actually doing the implementation which he has identified as causing project failures with the Information Systems area. In a later paper, Flowers (Flowers, 1997) only identifies 15 factors, mostly differently labeled and identifies for six infamous cases what the factors involved were, details on this information can be found in ***Appendix 5***.

These Critical Failure Factors (***Table 2***) are used as one of the building blocks for the **Part A.** of this research; below the table are listed definitions of each factor.

TABLE 2 FLOWERS CRITICAL FAILURE FACTORS

Organizational context

Hostile Culture
Poor reporting structures

Management of project

Over-Commitment
Political pressures

Conduct of the Project

Initiation Phase

Technology Focused
Lure of leading edge
Complexity underestimated

Analysis & Design phase

Poor consultation
Design by committee
Technical 'fix' for management problem
Poor procurement

Development phase

Staff turnover
Competency
Communication

Implementation phase

Receding deadlines
Inadequate testing
Inadequate user Training

- ***Source Flowers*** (1996,pp 158)

F1 Hosile Culture. This factor relates to the overall culture of the organisation / organisations being worked with. Do they shoot the messenger who delivers the bad news, do they look for 'scapegoats', are they reluctant to change their methods and generally made like difficult for outside consultants.

F2 **Poor reporting.** This factor relates to the overall reporting for the project, or the entities within the project; do status reports and requests for meetings and information get passed to the upper management echelons.

F3 **Over commitment.** This factor relates typically to the management or sponsorship of the projects. Managers may be too keen on the project succeeding and make it their personal vision, they may alter figure and put misleading information in reports to justify their projects survival, i.e. the oasis is just over the next sand dune.

F4 **Politics.** This factor relates to any political matters of influence that may creep into the project. It can be either within the Government, you development team or in the departments being worked on.

F5 **Technology focused.** This factor relates to the project being more related to the technical aspects as opposed to the human aspects of the system. This could be partly due to the stereotypical image of programmers and other techies.

F6 Lure of Leading edge. This factor relates to the technological factors of the project, while Information Technology can give a company a leading edge in the marketplace, it is by no means always true. Some new leading technologies will flop and is it wise to attempt the latest technology, or a tried and tested system.

F7 Complexity. This factor relates to the overall complexity of the project. This may happen because of lack of understanding about the project, or it could be in attempt to simplify the project for estimation reasons.

F8 Poor consultation. This factor relates to the lack of consultation with the major stakeholders of the project during the analysis phase.

F9 Design by committee. This factor relates to all the problems with a project, or the project requirements being designed by a committee. The problems concerning conflicting personalities, power struggles, game playing and alliances etc.

F10 Technical fix for management issue. This factor relates to those problems that are management issues and cannot be solved exclusively with technological solutions.

F11 Poor procurement. This factor relates to all procurement factors, this could be related to any hardware or software components that need to be purchased off the shelf.

F12 Turnover (staff). This factor concerns the turnover of staff from your project; while some turnover of any project is natural, excessive turnover could imply that your project in is trouble for a variety of reasons.

F13 Competency. This factor concerns the competency of your staff in all aspects of your project.

F14 Communication. This factor concerns the channels of communications both with the organisation and the project team. Examples of use here could be to combat the rumor mills.

F15 Deadlines. This factor concerns the slippage of schedule, missed milestones, and other time goals of the project not being met.

F16 Testing. This factor relates to testing the new systems both at the conceptual level i.e. the Entity Relation Ships and Data Flow Diagrams, and the physical level i.e. the programs and systems themselves.

F17 Training. This Factor relates to all training matters, wither it be development staff, users, user-support training.

2.11 EXPEDITING ACTIONS FOR RUNAWAY PROJECTS.

Cole (1995), on behalf of KPMG conducted a study in 1994 to study runaway projects within the Information Technology area. This study was actually a follow up to an identical study run five years previously.

The original study in 1989 was conducted upon 250 major organisations within the United Kingdom from a variety of different sectors. The interviews were conducted by an independent research body via the telephone, and lasted up to 45 minutes with senior personnel in Information Technology, Finance or Operations (Cole, 1995).

In 1994, the same independent research company was again commissioned to re-run the same test on the original participants. Approximately half of the original respondents were willing to be in the survey, 120 companies in all (Cole, 1995).

The summaries of Cole's findings will be listed below. [1]

[1] These figures are estimates only. They based on the original results that were released in graph format only.

Figure 2-12 below shows the remedies that the respondents said that they would apply to fix a runaway project in graphical format,

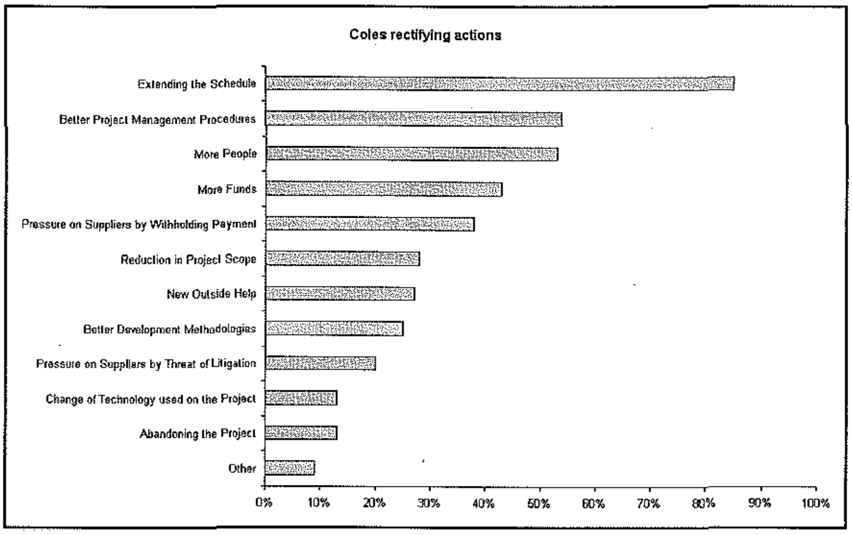


FIGURE 2-12. THE ACTIONS IDENTIFIED FOR RECTIFYING A RUNAWAY PROJECT – Source (Cole, 1995,Page 2)

Table 3. below, is the same information in a tabular format.

TABLE 3. COLE'S RECTIFYING ACTIONS

Action	Percentage
Extending the Schedule	85%
Better Project Management Procedures	54%
More People	53%
More Funds	43%
Pressure on Suppliers by Withholding Payment	38%
Reduction in Project Scope	28%
New Outside Help	27%
Better Development Methodologies	25%
Pressure on Suppliers by Threat of Litigation	20%
Change of Technology used on the Project	13%
Abandoning the Project	13%
Other	9%

SOURCE (Cole, 1995, p2)

The study (Cole, 1995) also asked of the respondents what if any adverse effects the runaway projects had on their company as a whole ***Figure 2-13*** below shows the responses to this graphically, ***Table 4.*** displays the same data in tabular format. Wasted resources (time, money, etc., although they all equate to money) and reduced moral within the company were the main adverse effects discovered in the study.

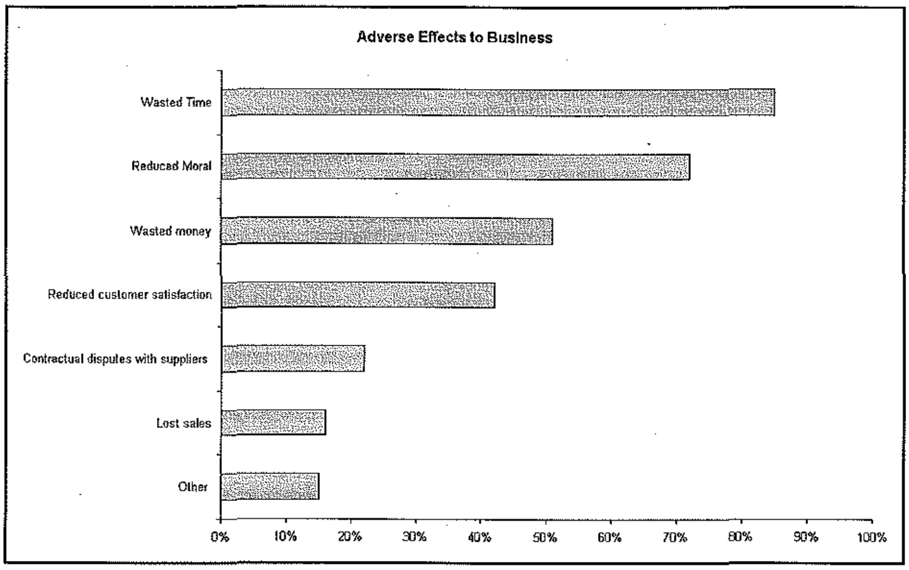


FIGURE 2-13. THE ADVERSE EFFECTS THE RUNAWAY PROJECT HAD ON BUSINESS.
- Source (Cole, 1995,Page 2)

TABLE 4. COLE'S ADVERSE EFFECTS TO THE BUSINESS

Action	Percentage
Wasted Time	85%
Reduced Moral	72%
Wasted money	51%
Reduced customer satisfaction	42%
Contractual disputes with suppliers	22%
Lost sales	16%
Other	15%

- Source (Cole, 1995,Page 2)

Another part of the research also looked at what actions or procedures they respondents would pay more attention to in the future (Cole, 1995). ***Figure 2-14***. displays these actions in graphical format. ***Table 5*** displays the data in tabular format. The majority of the factors identified in the study were project management issues, both at the design and implementation stages.

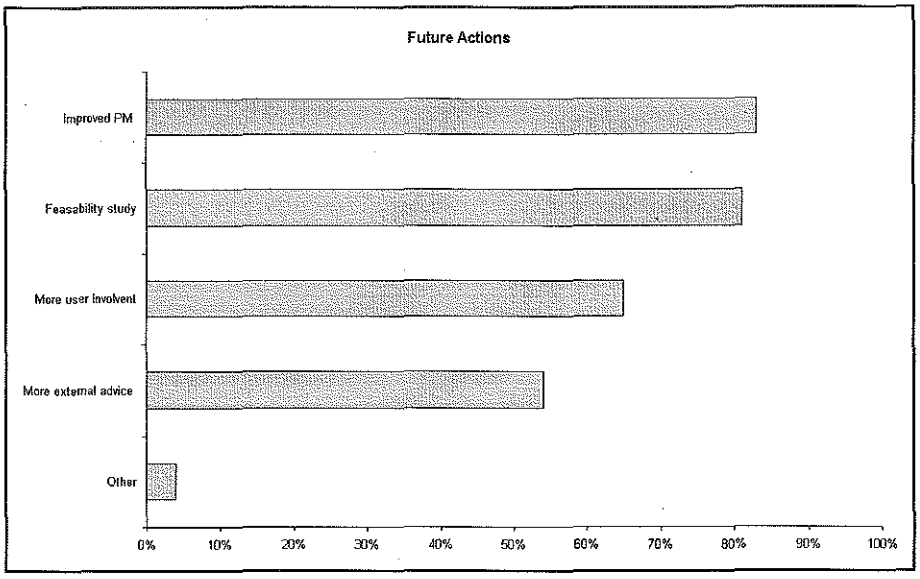


FIGURE 2-14. ACTIONS IDENTIFIED THAT WILL BE ATTEMPTED IN THE FUTURE FOR PROJECTS.

– Source (Cole, 1995,Page 2)

TABLE 5. FUTURE ACTIONS THAT WILL BE TAKEN TO AVOID RUNAWAY PROJECTS.

Future Action	**Percentage**
Improved Project Management processes	83%
Feasibility study	81%
More user involvement	65%
More external advice	54%
Other	4%

– Source (Cole, 1995,Page 2)

The last question of the study (Cole, 1995) was in regards to why the project failed. ***Figure 2-15*** and ***Table 6*** below show the responses to this question both graphically and in tabular format. Most of the factors again come back to the project management, and the failure to plan the project properly.

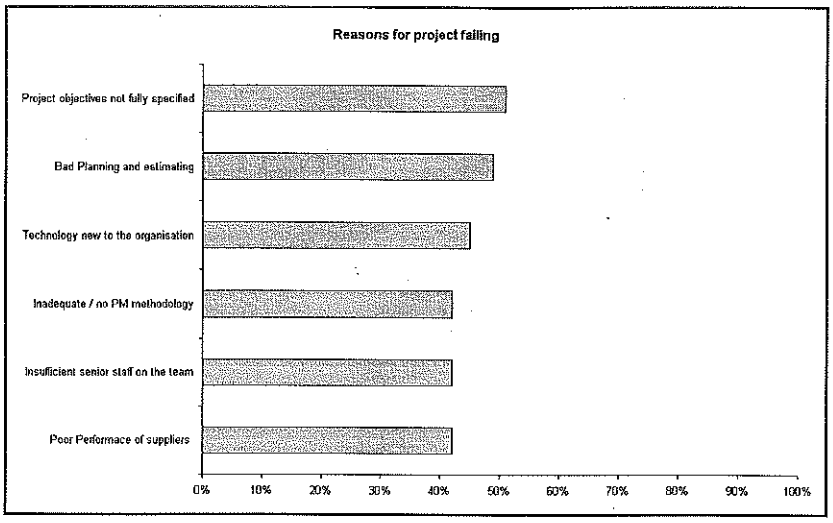


FIGURE 2-15. THE IDENTIFIED REASONS FOR PROJECTS FAILING.
– Source (Cole, 1995,Page 3)

TABLE 6. THE REASONS IDENTIFIED FOR PROJECTS FAILING

Reason for failing	**Percentage**
Project objectives not fully specified	51%
Bad Planning and estimating	49%
Technology new to the organization	45%
Inadequate / no Project Management methodology	42%
Insufficient senior staff on the team	42%
Poor performance of suppliers	42%

– Source (Cole, 1995,Page 3)

2.12 CAN I RECOVER THIS PROJECT ?

Jones (1995) conducted a study to examine if large systems projects are likely to be cancelled and if so why, and various other factors. He also identified some major problems that can effect the project ***Table*** *7*, which stage of project the event happened, what the triggering factors were, and the likelihood it can be rectified, both as a percentage and a rating.

*TABLE 7. **HOW LONG IS TOO FAR GONE***

Development Phase	**Recovery prospects**	**Successful projects**	**Triggering factors**
Early Planning	Excellent	95%	2.12.1.1.1.1 Prudence
Requirements	Excellent	90%	Sizing, cost estimating
Initial design	Very good	80%	Creeping requirements
Detail design	Good	65%	Creeping requirements
Coding	Fair	45%	Schedules, cost overruns
Integration	Poor	30%	Schedules, cost overruns
Testing	Very Poor	15%	Schedules, poor quality
Deployment	Nonexistent	0%	Poor quality
Maintenance	Nonexistent	0%	Poor quality

- Source (Jones, 1995, Page 7)

2.13 PROBLEMS WITH IDENDIFYING FAILURE FACTORS.

While this study uses two sets of factors (Cole, 1995; Flowers, 1996) as the basis of the research it should be noted that they represent the particular interpretation of each analyst. Belassi (1996, Pp 142 - 143) has demonstrated how the factors considered critical by different authors vary considerably ***Table 7.***

TABLE 8. COMPARISON OF SEVEN CRITICAL SUCCESS LISTS.

Martin (1976)	*Locke* (1984)	*Cleland & King* (1983)	*Sayles & Chandler* (1971)	Baker, Murphy & Fisher (1983)	*Pinto & Slevin* (1989)	*Morris and Hough* (1987)
Define Goals	Make project commitments known	Project summary	project manager's competence	Clear goals	Top management support	Project objectives
Select project organisational philosophy	Project authority from the top	Operational concept	Scheduling	Goal commitment of project team	Client consultation	Technical uncertainty innovation
General Management support	Appoint competent project manager	Top management support	Control systems and responsibilities	On-site project manager	Personnel recruitment	Politics
Organise and delegate authority	Set up communica-tions and procedures	Financial support	Monitoring and feedback	Adequate funding to completion	Technical tasks	Community involvement
Select project team	Set up control mechanisms (schedules, etc)	Logistic requirements	Continuing involve-ment in the project	Adequate project team capability	Client acceptance	Schedule duration urgency
Allocate sufficient resources	Progress meetings	Facility support		Accurate initial cost estimates	Monitoring and feedback	Financial contract legal problems
Provide for control and information mechanisms		Market intelligence (who is the client)		Minimum start-up difficulties	Communication	Implementation problems
Require planning and review		Project schedule		Planning and control techniques	Trouble shooting	
		Executive develop-ment and training		Task (Vs. Social orientation)	Characteristics of the project team leader	
		Manpower and organization		Absence of bureaucracy	Power and politics	
		Acquisition			Environmental events	
		Information and communication channels			Urgency	
		Project review				

Source (Belassi & Tukel, 1996, Pages 142 - 143)

3 METHODOLOGY.

3.1 DESIGN

3.1.1 PLANNING

The first part of the study was based on a semi-structured interview in which project manager were given a scenario based on a "disguised" historical case taken from existing literature. They were asked to think about the main planning and management issues that they could predict which may complicate the project. A very simplified description of the required system was given, technical matters such as number of users, workstations, response times etc were only made available when specifically asked for. The project manager was free to use whatever technology, computer language, methodology etc that they feel comfortable using.

3.1.2 EXPEDITING

The second part of the study is also based on a semi-structured interview in which a progress report was given about the disguised scenario after a certain period of time has elapsed in the project. Participants were asked what actions they as the project manager of this scenario they would likely take to rectify the problems faced in this situation.

3.2 SAMPLE.

The sample consisted of eight Information Systems professionals who have experience in project management within software development field. The sample was not a random one, due to the requirements set by myself, that is, the nature of the scenario, it was modeled on a very large and complex project. The project mangers selected for interviews must come from a company, or have experience in a company, that could actually undertake a project of such magnitude, the question of had they worked on a project of this size was not addressed. These requirements eliminated most possible companies who deal in web development and most small computer companies.

The companies that were initially selected for canvassing had to be large enough to either:

a) Have their own in-house software development team.

b) Be in a position to commission the same development services to other companies.

From this pool of companies, each was contacted via the telephone and asked if they would be willing provide a project manager to participate in the study at a later date.

The final sample contained companies within the accounting big 5 companies, education, computer consultancy, finance, and high technology research. Of the ten interviewed project managers, only eight of the interviews were useable, due to tape recorder malfunctions.

3.3 THE SCENARIO

The scenario is based is the MANDATA project (***APPENDIX 1***). This project was initiated in 1970 and abandoned in 1981 by the Australian Commonwealth Government (Sauer, 1993). The aim of the project was to centralise the Australian Public Service (APS) records of employees, job records, job descriptions, payments of money, job applicants, job vacancies and organisational data.

Until this project was proposed each government organization kept its own records, usually in the form of index sheets stored in files. The Auditor General had always been critical of the record keeping of these organizations. By centralising the records in a computerised method the savings of money over incorrectly maintained records would be significant. While the previous manual systems seemed simple, take the Postal Service, they employed 150,000 staff, of these 1860 were employed in the employee record keeping in over 400 locations across Australia (Sauer, 1993).

Table 9. Reasons why MANDATA failed.

S1	Commitment of the various departments to the project.
S2	Control over the government departments to carry out tasks for the project.
S3	The potential of withdrawal of departments to build their own system
S4	The existence of a competing system (the Treasury was building a distributed payroll system).
S5	Bureaucratic decision-making caused by the hierarchical structure of the body governing the project.
S6	The need for specific resources to be assigned to the consultation processes between the various parties involved.
S7	Staff shortage caused by lack of available skilled IT professionals.
S8	Early flaws in the design which are inevitable but which need to be corrected.
S9	Physical accommodation of system which was mainframe based
S10	Delay in the supply of components from manufacturers.
S11	Flexibility of contracts with suppliers, so that arrangements can be adjusted to fit progress.
S12	The need for reviews to be built into the project so that early problems can be formally identified.
S13	Economic crisis in Australia in the mid 1970s.
S14	Change of government following the demise of Whitlam.

The reasons for MANDATA failing, as summarised by the researchers (Martin & Smith, 2000) after carefully reading the case study has been displayed in ***Table 9***.

The scenario was changed to disguise as many of its features as possible and to convert it to a 1990's project instead of a 1970's one. The majority of the information about the project remained unchanged, it was assumed that the Sauers' (1993) case study was not a well-known one. The London Ambulance Service, SABRE and TAURUS projects (Flowers, 1996) were looked at as possible case studies to use, but it was decided that these were too well known and some contained large technology focused areas to be effective case studies.

A list of the Factors affecting the success/failure of the 'Mandata' project during phases of initiation and initial development (1971-1976), reported by Sauer (1993) and relevance to this study is available in ***Appendix 2***.

3.4 THE INTERVIEW

The interview was designed to last approximately 40 minutes, that time broken down into three sections;

a) The Planning section	15 Minutes.
b) The Expediting section	15 Minutes.
c) The Debrief	10 Minutes.

One hour was requested of each project manager's time, all interviews ran within the allocated time limit. The entire interview apart from the debriefing was audio recorded for later analysis. The scenario (***Appendix 1***) was given to the project manager one section at a time, and asked to say whatever they thought in regards to the project as they read, most preferred to answer the questions as they read them.

The debrief was introduced to answer any questions that they had about the case study, the research being carried out, or to clear up anything they mentioned that wasn't too clear on during the previous two sections. They were also told about the real case study if they wished to know about it.

Each interview was recorded with the participants' permission. These tapes were then listened to by myself and brief notes were written about what was said by the interviewee, these were broken down into generalised statements about what was talked about i.e, Size of project, Technologies etc.

Each of these general statements was then applied to the factors that Flowers (1996) and Cole (1995) listed in their respective literatures using a Labelling system of F1 to F17 for the Flowers' table and C1 to C11 for Coles'. When a statement did not fit into any of the mentioned factors, a note was made and it was assigned a XF or XC number.

This information was then transferred to a spreadsheet for analysis, no record was make of what order the statements were made in, nor how often they were repeated. The mentioning of material related to that factor once was enough for this study.

3.5 ANALYSIS OF DATA

The data contained within the spreadsheet, the plan was to cross-tabulate each identified factor and convert it into a percentage of interviewees who identified that factor. ***Table 10*** below is an example of how the factors will be loaded into a table and the percentage of respondents who mentioned it calculated.

TABLE 10. AN EXAMPLE OF THE DATA WAS FORMATTED FOR ANALYSIS.

		Interviewees					
		I1	I2	I3	I4	I5	Proportion Considering factor
Y1	Factor 1	✓	✓		✓		60%
Y2	Factor 2	✓		✓		✓	60%
Y3	Factor 3		✓	✓		✓	60%
XY1	Extra Factor *x*				✓		20%
XY2	Extra Factor *x+1*	✓	✓	✓	✓	✓	80%

It was decided to only use the data that was collected for each part in its own question. While many of the factors identified in each question could relate to each other part, it was decided to keep the data separate.

3.5.1 PART A. PLANNING.

The factors gathered from Part A (Planning) of the interview were checked against the Critical Failure Factors (Flowers, 1996) ***Table 2 Flowers Critical Failure factors*** (page 2-42 of this dissertation) as identified by Flowers to see how his factors compare with those identified by Western Australian software industry.

Information on what Flowers said constitutes each of the Critical Failure Factors can be found in the preceding chapter "Literature Review".

A second table was made to store all of the issues raised by the project managers that Flowers did not mention. The resulting two tables will then be combined and sorted into three sub-classifications as shown below in ***Table 11***. The information was then ordered on percentages to discover which factors are most important to the Western Australian software industry

TABLE 11. PROPOSED LEVELS

Least Important / Common Factors	1% - 34%
Important / Common Factors	35% - 59%
Most Important / Common Factors	60% - 100%

The factors can then be referenced against what Flowers identified as his factors to see if the interviewed project managers agree with his factors. No direct comparison is possible against the Flowers (1996) factors as no percentages are listed in his table.[2]

3.5.2 Part B. Expediting.

The factors gathered from Part B (Expediting) of the interview will be checked against the rectifying actions ***Table 3. Cole's rectifying actions*** (page 2-48 of this dissertation), as adopted from Cole (1995) to see how his factors reflect upon those used within the Western Australian Industry.

[2] While in his 1996 book no figures are given, in an article the following year he identifies which case studies had what factors.

A second table was made to store all of the rectifying actions which were raised by the project managers that Cole (1995) did not mention. The resulting two tables will then be combined and sorted into three sub-classifications as shown above in ***Table 3.*** This information was then ordered on percentage to discover which are the more commonly used methods for expediting a late project in the Western Australian software industry. Factor C12 will not be used, the answers from the second table could answer that, but doing so would have C12 reading 100% in this study.

With this information ordered on percentages to discover what are the more prevalent actions that project managers take to fix late projects within the Western Australian software industry. This data was also checked again books on project management, in particular software related projects to see if they follow any trends set in literature.

3.6 PROBLEMS WITH RESEARCH

3.6.1 THE COMPARISONS.

3.6.1.1 PART A.

No foreseeable problems.

3.6.1.2 PART B.

A few problems arise with using this study:

3.6.2 THE METHOD OF INTERVIEW:

While little information is given on the methodology of Coles' study, it does mention that it was conducted over the telephone (Cole, 1995). It does not mention if the person was given a list of likely options to pick from, or asked to say what they would do without any prompting. This study was conducted using face-to-face interviews and the participants were not given a list of possible actions, rather forced to think about the problem with minimal interaction from the interviewer.

3.6.3 THE TYPE OF PERSON INTERVIEWED:

Cole (1995) stated that he interviewed a wide range of managers including CEO's, IT managers, Department managers etc. Whilst this research concentrated exclusively on Information Systems managers, who are, or have been in project management roles, in particular software development projects. This is only a problem insofar as the participants of this research actually are the project managers, not the administration roles which Cole surveyed.

3.6.4 Size of Sample:

Is a sample of eight project managers within Western Australia a fair representation of the entire Information Systems projects environment within Western Australia? The answer to this question the author believes is 'no', but for a study of this proportion it was decided that the sample of eight would be sufficient.

3.6.5 Geographic Location:

This is not a major issue, but one that must be raised; the Cole (1995) study was done in the United Kingdom whilst this study was conducted in Western Australia. This should not affect the results, as the corrective actions by project managers should be very similar in nature regardless of the location.

3.6.6 Transcribing the Interviews.

If independent raters were to analyse the content of the interviews, they might have provided more objectivity. However, it was reasoned that adequate interpretation required familiarity with the MANDATA case and the particular interview. Only the researcher had sufficient familiarity to make the subtle judgments needed.

3.6.7 AGE OF SCENARIO.

The case study used for this study was based on 1970's technology and practices. This type of project would never be done today, in those days' huge banks of programmers, projects that lasted years, unclear specifications were all quite normal. Today, projects tend to be much shorter in duration. For instance, in web development companies three months is classed as a long project.

A more recent case study would have been better, but gaining a useful example proved difficult. No organisations that were approached were willing to disclose sufficient information about there failed projects to be of any use. A Westpac bank case (Glass, 1998) looked promising, but little information was available.

3.6.8 SIMPLENESS OF SCENARIO.

This point was brought up in several of the interviews; the scenario was deliberately made simple. It was presented as summarised information totalling two pages, with this was as many factors as thought reasonably possible. Those two pages covered over 200 pages of material in Sauer (1993).

The project managers interviewed repeatedly asked for feasibility studies, budgets, information requirement statements, specifications etc. Many managers found it difficult to come to terms with the fact that these were not available for perusal, stating that they needed more information for some questions asked. By allowing project managers access to additional paperwork, and facts the interviews could of easily become side tracked.

4 RESULTS.

4.1 Part A. The Planning Process.

4.1.1 Flowers (1996 and 1997)

The original study (Flowers, 1996) had no percentages within it, the following year an article (Flowers, 1997) was published, this article had the factors for each case he used in his book (***Appendix 5***). From this ***Table 12*** was created that gave a baseline for comparison. Several of the identified factors changed, some were added, others deleted from the original study. ***Table 13*** below shows the studies percentages against the categories addressed by Flowers (1996, 1997).

TABLE 12. FLOWERS (1997)UPDATED CRITICAL FACTORS

Factor (1997)	Percent
Fear Based culture	50%
Poor Reporting	83%
Over Commitment	67%
Politics	50%
Technology focused	50%
Leading edge system	100%
Poor consultation	50%
Technical fix sought	33%
Weak procurement	50%
Project timetable slippage	100%
Testing	33%
Training	50%
Changing requirements	50%
Development sites split	33%

4.1.2 FLOWERS FACTORS BY INTERVIEWEES.

TABLE 13. FLOWERS FACTORS AS RATED BY INTERVIEWEES

ID	Action	Percentage
F1	Hostile Culture	75%
F2	Poor Reporting	38%
F3	Over Commitment	0%
F4	Politics	88%
F5	Technology focused.	75%
F6	Lure of leading edge	38%
F7	Complexity underestimated	50%
F8	Poor consultation	50%
F9	Design by committee	13%
F10	Technical fix.	75%
F11	Poor procurement	38%
F12	Turnover (staff)	25%
F13	Competency	13%
F14	Communication	38%
F15	Deadlines	88%
F16	Testing	13%
F17	Training	50%

4.1.3 FLOWERS (1996;1997) COMPARED

TABLE 14. FLOWERS (1996)(1997) VS MARTIN

ID	Flowers (1996) data.	Flowers (1997) data	Expected	Finding
F1	Hostile Culture	Fear Based culture	50%	75%
F2	Poor Reporting	Poor Reporting	83%	38%
F3	Over Commitment	Over Commitment	67%	0%
F4	Politics	Politics	50%	88%
F5	Technology focused.	Technology focused	50%	75%
F6	Lure of leading edge	Leading edge system	100%	38%
F7	Complexity underestimated	Complexity underestimated	0%	50%
F8	Poor consultation	Poor consultation	50%	50%
F9	Design by committee	N/A	0%	13%
F10	Technical fix.	Technical fix sought	33%	75%
F11	Poor procurement	Weak procurement	50%	38%
F12	Turnover (staff)	N/A	0%	25%
F13	Competency	N/A	0%	13%
F14	Communication	N/A	0%	38%
F15	Deadlines	Project timetable slippage	100%	88%
F16	Testing	Testing	33%	13%
F17	Training	Training	50%	50%
F18	N/A	Changing requirements	50%	0%
F19	Development type (XF8)	Development sites split	33%	38%

Table 14 above is a comparative view of both Flowers data (Flowers, 1996, 1997) with this study. A comparison of both sets of factors was done, and 19 factors identified, any with N/A meant it was missing from the other published list. No direct statistical comparison between the two is possible.

4.1.4 *FACTORS NOT ADDRESSED BY FLOWERS*

The below ***Table 15*** represents all the factors which were identified by the interviewees which Flowers (1996; 1997) did not identify.

TABLE 15. FACTORS NOT IDENTIFIED IN FLOWERS

ID	Action	Percentage
	Most Important	
XF1	Duration / size	100%
XF2	Sponsorship	88%
XF3	Documentation of plan	75%
XF4	Business rules / processes	75%
XF5	Response time	63%
XF6	Geography	63%
	Important	
XF7	Better system knowledge	50%
XF8	Development type	38%
XF9	Change management	38%
XF10	Existing infrastructure	38%
	Least Important	
XF11	Control Issues	25%
XF12	Have done a similar Project	25%
XF13	Excess staff	13%
XF14	Team breakdown	13%
XF15	Why / justification	13%

4.1.5 DEFINITIONS OF FACTORS AND SELECTED QUOTES.

The factors that were taken from the interview which did not fit into Critical Failure Factors (Flowers, 1996,pp 158) ***Table 15*** and had new factors created are described below. Actual quotes have been included in bold typeface to give an example of some of dialogue of the interviews, it was decided not to identify which interviewee said what.

F1 Hosile Culture.

"The culture of the organisations that will be using these things, so that would be the biggest problem I would say."

F2 Poor reporting.

"Until I had a proper IRD (information requirements determination***) it could not be planned, I don't know who wants what in the organisations."***

F4 Politics.

"***Ummmm, the government is wanting this run, it will most likely be a political minefield, projects and politics don't mix the best.***"

F5 Technology focused.

"All I see in this project is technical requirements, what about the business rules, processes that both currently exist and perhaps need to be altered or created?"

F7 Complexity.

"We often find that what the customer actually wants is far more complex than is stated in the requirements, or similarly we read it wrong and assume it simpler than it actually is."

F8 Poor consultation.

"They should buy rather than build – they lack the experienced people to build it"

F9 Design by committee.

"As long as they [the committee] *knows what it wants, has some technical understanding of the requirements few problems arise, its when users change what they want, or non-technical people become involved that I find problems arise."*

F10 Technical fix for management issue.

"Depending on location, we may need .... [lists many technological options for fixing the project]"

F11 Poor procurement.

"I don't see procurement being an issue here, while we are using off-the-shelf components I would hope that I had investigated the vendors before deciding on their product, if I had such a problem I would consider not using them in the future again."

F12 Turnover (staff).

"Personnel is always a problem in any project, you just have to be ready to hire more people and train them up as needed, and over 7 years, you would want to ensure that the skill/knowledge base is still there and is passed on"

F14 Communication.

"Get all the interested parties in a round the table discussion, iron out as many problems as you can as early as possible."

F15 Deadlines.

"all very well to have a rough set of milestones or time frame, but 7 years for anything is ridiculous, it should be broken down into smaller bits – deliverables"

F17 Training.

"They will be able to get a new system in, but then training people to use it or getting their heads around what such a system can do. "

XF1 Duration / Size. This factor was in regards to the overall size and or duration of the project given. It was agreed by all participants in the interviews that seven years was far too long for a IS project.

"Seven years is a ridiculous amount of time, in the sense that if you had a project so long, it may never finish and if it did, what you delivered would be totally irrelevant"

XF2 Sponsorship. This factor was in regards to the overall sponsorship issues which can arise within a project. These range from a champion for the project in high management to users supporting the proposed system.

"You need someone in high management to be behind your project, a champion as such, someone who is willing to force the changes on reluctant parties"

"This project would need to be owned by somebody within the very highest level of government within the country."

XF3 **Documentation of Plan.** This factor is a very broad one, it includes feasibility studies, system requirements, project plans and estimations, budgets etc. In summary an documentation of the project.

"where are the information requirements, the feasibility study, this document is inadequate."

XF4 **Business Rules and Processes.** This factor it could be argued could also come under *XF3*, I have included it as a separate factor. The designers need to understand the variety of different rules and processes that each and every business unit / department has. In some cases these may need to be re-engineered or common rules / processes established, this would also effect *XF2*.

"You have 100 odd departments, probably each with their own way of doing things [rules and procedures] this would be a major problem that would need to be addressed, wither by standardising or programming each entities rules as they use them."

"All I see in this project is technical requirements, what about the business rules, processes that both currently exist and perhaps need to be altered or created?"

XF5 **Response Time**. This factor is concerned the expected response times that where stated in the pre-requests, "The system must take no longer than it would take to manually look up the record in the old system, estimated at some 3 - 7 seconds.". It was widely believed this was unrealistic "[Said with obvious sarcasm] ***Each person must have been poised over their card box awaiting a query, it sounds too unrealistic.*** "

XF6 **Geography.** This factor is concerned with the geography of the project, i.e in this case it is "*A developing, geographically large*". The vast distances of Australia could prove some problems in technologically immature setting as defined in the Scenario.

"Geographically large, that is going to present some problems as it is, travel of staff, systems may be spread over that geographical location also"

XF7 **Better System Knowledge.** This factor concerns the knowledge of the systems, both better knowledge of:

a) The required system, i.e. sufficient technical details.

b) The existing systems, what is in place, what needs to be replaced, or can used.

"Before going any further I would need much more information about what systems they expect, what is currently in place, how do they intend to use the new system etc."

XF8 **Development Type.** This factor is one considering what type of development will be undertaken, is it staged – one department at a time, cold turkey – i.e. the entire system is changed in the one instance, parallel running – the new system and old run side by side until the old one is phased out.

"Would probably need to build everything from the ground up, no computers are there, no infrastructure..

XF9 **Change Management.** This factor generally considers all aspects of change management that may be required. This includes Business Processes Re-engineering.

"Many issues regarding change management will likely come into this project, we would advice our clients to talk to our change management group about some issues."

XF10 **Existing infrastructure.** This factor is concerning what the existing infrastructure with the country is. This is more of the physical infrastructure, i.e cabling, networks, computing facilities etc.

"They should be able to build their infrastructure, they can hire people to do that, it really depends on how much money they have to spend"

"Developing country, this probably means little or no existing infrastructure."

XF11 **Control Issues.** This factor is one, which covers control issues in general, it could be ownership (bordering on Sponsorship), general ownership problems with some components of perth.

XF12 **Have done a similar project.** This factor is one which was raised exclusively within the large multi-national corporations interviewed, as they have a wide range of clients who have wanted many projects done, they first look for similar projects done by their company.

"We have an extensive repository of all projects our company has done, I would first search that for similar projects, our projects that have similar factors,"

XF13 **Excess Staff.** This factor is concerning the excess staff both in the project teams and in the organisations being worked on.

"Quite frightening, when you look at the numbers of people employed in doing thing that are all done automatically now"

XF14 **Team breakdown.** This factor is concerning the make-up or structure of the team that works on the project, this could also include internal politics within your own project team.

XF15 **Why / justification.** This factor addresses what is the reason for the project? Why do we need it? What benefits will come out of it? Who decided we needed it? Are some of the possible questions that could be raised here, some of this could be overlapping with ***XF3***, though the below quote prompted me to separate it.

"What is the reason for this project, what do they hope to achieve, the government just wanting it is not reason enough"

4.1.6 ALL PART A FACTORS AS FOUND.

The following ***Table 16*** represents all of the factors, both the ones identified by Flowers and the ones identified by the interviewees in this study. The responses have been ranked on the percentage of respondents.

TABLE 16. ALL PART A ANSWERS IN ORDER

ID	Action	Percentage
	Most Important	
XF1	Duration / size	100%
F15	Deadlines	88%
F4	Politics	88%
XF2	Sponsorship	88%
XF4	Business rules / processes	75%
XF3	Documentation of plan	75%
F1	Hostile Culture	75%
F10	Technical fix.	75%
F5	Technology focused.	75%
XF6	Geography	63%
XF5	Response time	63%
	Important	
XF7	Better system knowledge	50%
F7	Complexity underestimated	50%
F8	Poor consultation	50%
F17	Training	50%
XF9	Change management	38%
F14	Communication	38%
XF8	Development type	38%
XF10	Existing infrastructure	38%
F6	Lure of leading edge	38%
F11	Poor procurement	38%
F2	Poor Reporting	38%
	Least Important	
XF11	Control Issues	25%
XF12	Have done a similar Project	25%
F12	Turnover (staff)	25%
F13	Competency	13%
F9	Design by committee	13%
XF13	Excess staff	13%
XF14	Team breakdown	13%
F16	Testing	13%
XF15	Why / justification	13%
F3	Over Commitment	0%

4.2 PART B. THE EXPEDITING PROCESS.

This section reports the results as found for Part B (Expediting) of this study.

4.2.1 COLES ACTIONS IDENTIFIED IN WESTERN AUSTRALIA

Table 17 below displays the factors which Cole (1995) identified as the actions which project managers in the Information Systems industry are likely to undertake to rectify a late project. These were assigned percentage in the text, which were placed in the ***Expected***, and the ***Findings*** columns

TABLE 17. COMPARISON OF COLE AGAINST MARTIN

Ranking	Action	Expected	Findings
C1	Extending the Schedule	85%	100%
C2	Better Project Management Procedures	54%	100%
C3	More People	53%	100%
C4	More Funds	43%	100%
C5	Pressure on Suppliers Payment	38%	63%
C6	Reduction in Project Scope	28%	88%
C7	New Outside Help	27%	63%
C8	Better Development Methodologies	25%	75%
C9	Pressure Suppliers by Litigation	20%	25%
C10	Change of Technology	13%	25%
C11	Abandoning the Project	13%	63%
C12	Other	9%	N/A

4.2.2 FACTORS COLE DID NOT ADDRESS

This section contains the expediting actions identified by the interviewees which were not addressed by Cole (1995). This data is displayed below in ***Table 18***.

TABLE 18. FACTORS NOT ADDRESSED BY COLE

Ranking	Action	Findings
	Most Important.	
XC1	Staff Issues	100%
XC2	Stockholder issues	75%
XC3	Slippage concern	75%
XC4	Adjust / redo plan	63%
	Important	
XC5	Management issues	50%
XC6	Contract reviews	50%
XC7	Risk assessment	38%
	Least Important	
XC8	Never abandon project	25%
XC9	Resign	25%
XC10	better initial planning	25%

4.2.3 ALL PART B FACTORS IDENTIFIED IN WESTERN AUSTRALIA.

This section details all the expediting actions identified ordered on percentage of respondents who identified that action. ***Table 19*** displays this information in tabular format.

TABLE 19. MARTINS COMPLETE ORDERING COLES QUESTION

Ranking	Action	Percentage
	Most Important	
C2	Better Project Management Procedures	100%
C1	Extending the Schedule	100%
C4	More Funds	100%
C3	More People	100%
XC1	Staff Issues	100%
C6	Reduction in Project Scope	88%
C8	Better Development Methodologies	75%
XC3	Slippage concern	75%
XC2	Stockholder issues	75%
C11	Abandoning the Project	63%
XC4	Adjust / redo plan	63%
C7	New Outside Help	63%
C5	Pressure on Suppliers Payment	63%
	Important	
XC6	Contract reviews	50%
XC5	Management issues	50%
XC7	Risk assessment	38%
	Least Important	
XC10	Better initial planning	25%
C10	Change of Technology	25%
XC8	Never abandon project	25%
C9	Pressure Suppliers by Litigation	25%
XC9	Resign	25%

4.2.4 DEFINITIONS AND ELECTED QUOTES

The factors that were created from the interviews which did not fit into the existing rectifying actions (Cole, 1995) are described below; as well as quotes from the interviews.

C1 **Extending the Schedule. .**

"We can always allocate more time to completion date if we have to, most projects I have been in have done it."

C2 **Better Project Management Procedures.**

*"I would examine the current way the project /*project in trouble*/ is managed, is it too lengthy on paperwork, too laidback with little supervision etc. I may need to change some practices now, or in my next project."*

C3 **More People.**

"We can always hire some temporary people to do some work later in the project if need be"

C5 **Pressure on Suppliers Payment.**

"Look at changing suppliers if the don't perform, look for others who have a similar product"

C6 **Reduction in Project Scope.**

"Change the scope of the project to fit into the time allocated."

"If a project appears like it will run for over 6 months, we change its scope so that it fits into a 6 month time frame."

C7 New Outside Help.

"We can always hire some temporary people to do some work later in the project if need be, especially if the have some skills our present team lacks."

C8 Better Development Methodolgies.

"Our initial requirements were way out, investigate this so as we don't do it again"

C9 Pressure Suppliers by Litigation.

"I would avoid threatening with legal action at this stage, if I don't get a response then its an avenue I have that I can take."

C11 Abandoning the Project.

"With the government being un-supportive, we are pushing shit uphill [. . .] we may as well abandon it now, it all depends on the government"

XC1 **Staff Issues.** This factor is concerned all staffing issues of the project team, recruitment, incentives, moral and staff turnover.

"Its no use trying to pay them [your staff] more or give them a holiday to get your slippage under control, this would only be a short term solution"

"Need to find out why the moral of the team is low.. "

"Find out why staff moral is low, is it work, personal, the project, etc. "

XC2 **Stockholder Issues**. This factor is concerning the stockholders of the projects, and any concerns or issues that they may have, or any issues that the project manager may have with them.

"They may be losing their nerve a little bit, there is no reason panic by single entities. (in response to depts. Going their own way) That's what this sounds like to me."

XC3 **Slippage Concerns.** This factor concerns all slippage matters, are they concerned about them, at what point to they worry, analysis of how they got hat far behind in the first place.

"Its no use trying to pay them [your staff] more or give them a holiday to get your slippage under control, this would only be a short term solution"

"We have been going for 2 years out of 7, a little behind, not an issue yet. We have 4 ½ years to catch up."

XC4 **Adjust / redo plan.** This factor is in regards to the project plan, and typically is closely tied to XC3 as a result of slippage the plan may need to be altered / adjusted. If the project is 25% late at present, does that mean we assume the rest of the project will be at least 25% late also and adjust the plan accordingly.

"If you got to that stage and you were that far out, I think you would definitely need a complete review of the time frame"

XC5 **Management issues.** This factor typically was used by the project managers when the decision was too difficult for them (the contractual issues were mentioned in here), they would pass the entire issue onto their management, or if a particular department was being difficult consult with that departments management.

"It needs to be managed so that staff can leave the project and the entire project wont be effected, people should be able to be replaced, its good management structure."

XC6 **Contract reviews.** This factor was when the project manager did not pass the contractual obligations onto their supervisors. It was typically used for future reference, i.e. if Company A is late delivering components then look into them in more detail in future, incorporate penalty clauses etc.

"The project is far more difficult than at first thought, we may need to renegotiate the requirements."

"Review the contract carefully, can they change the requirements as the want, or is it fixed, if they can change it freely, plan on 14 years"

XC7 **Risk assessment.** This factor concerns risk assessment of this and future projects, this area was deliberately left lacking in this study. By identifying what can go wrong, and the likely hood of that happening before the project some plans can be made to plan the "what if" contingencies.

"The change of government is a big factor, it is probably contributing to the other problems, we should have identified that as a risk early on."

XC8 **Never abandon project**. This factor is concerned with the "We will never quit" mentality, this can be for a variety of reasons from a project manager naive belief he can complete the project to a corporate policy of never abandoning a project.

XC9 **Resign**. This factor addresses what some project managers would do when faced with such a project as the one used in the scenario, upon running a test interview with an engineering project manager, it was his only response.

"think I will hand in my resignation."

XC10 **Better initial planning**. This factor addresses more of what could be done following projects rather than this one, while it may be to late now to plan the overall project more carefully, in future the lessons learnt from this experience may be beneficial.

"Something, a plan or schedule to say what we would deliver and when."

5 DISCUSSION.

While percentages are used in this section, they are used only as a guide, no direct comparison between the studies is possible. Only generalised statements on the factors that the participants identified can be made.

Flowers (1996; 1997) study was done post project and examined the reasons why the project failed, while this study attempted to capture what project managers considered to be potential problems which could arise during the project.

The Cole (1995) study was conducted using a different format from this study, one that was not well described in the article. It consisted of rectifying actions which had taken place within that organisation, whilst this study asks "What will you do?"

5.1. Question 1.

"What are the factors that project managers consider being the most important when they are evaluating or planning Information Systems project with large software composition?"

Table 16 in the preceding chapter, has for readability been reproduced here and broken down into ***Tables 20, 21*** and ***22*** which represent the factors which project managers in Western Australia consider to be important, critical or relevant.

5.1.1. Most Important Factors.

From the study, 11 factors were categorized in the "***Most Important***" (***Table 20***) type. All of the project managers identified that the size or duration of the project (*Duration / size* and *Geography*) were of major concern to them, that is, the longer it is expected to run, the greater the chance of failure.

TABLE 20. MOST IMPORTANT FACTORS IDENTIFIED.

Most Important	
Duration / size	100%
Deadlines	88%
Politics	88%
Sponsorship	88%
Business rules / processes	75%
Documentation of plan	75%
Hostile Culture	75%
Technical fix.	75%
Technology focused.	75%
Geography	63%
Response time	63%

Many issues in regards to the organization were found, these being the understanding / misunderstanding of what organization requires, and how it presently works (*Business rules / processes*), the need for strong support from both within the organization and a powerful sponsor (*Sponsorship*), with this instrument in place the difficult political and organizational culture factors (*Politics* and *Hostile Culture*) may be lessened.

One issue that was raised was to do with project management techniques primarily, this was the factor concerning documentation of the plan (Documentation of plan), this aimed at ensuring the required system is understood and well planned. (*Business rules / processes)* also overlaps with this project management area, i.e. they need to understand the current system before the tinker with it.

The last area of issues of most importance to the project managers were the technical aspects, those associated with high-technology, how will we do it, how fast can we do it etc (*Technical fix*, *Technology focused* and *Response time*). They tend to get concentrate on the technology, and forget the reason why the technology was being used.

5.1.2. Important Factors.

The 11 factors that were categorized as those belonging to the "Important" type of factors are displayed below in ***Table 21***.

Table 21. Important factors identified.

Important	
Better system knowledge	50%
Complexity underestimated	50%
Poor consultation	50%
Training	50%
Change management	38%
Communication	38%
Development type	38%
Existing infrastructure	38%
Lure of leading edge	38%
Poor procurement	38%
Poor Reporting	38%

The majority of the factors identified in the important category can be directly associated to the organization which the project will be conducted. More information is often needed (*Better system knowledge*) in regards to both the requirements for the new system and the existing system, including hardware / infrastructure *(Existing infrastructure* and *Complexity underestimated*). The type of development (*Development type*) issue was also raised, that is, is it parallel, immediate switchover to new system. this also took into account any methodologies the project managers mentioned.

The communications area was also covered in some detail, while some of these factors could also be in the proceeding organizational paragraph. Communication inside both the organisation and the project team were mentioned, although not as many project managers mentioned this as was expected (*Communication*). Other communication issues raised were the need for training, both on a team and organisational level (*Training*) and the probable need for some support if re-engineering or process change is required (*Change management*). The last factor in this group is the lack of feedback or lax reporting (*Poor Reporting*) within both the organisation and the team.

The factors that are left that do not fit into the above areas are those which involve new or untried technology (Lure of leading edge), inept or insufficient consultation by the project staff (Poor consultation) and finally issues about getting off the shelf components purchased and delivered (Poor procurement).

5.1.3 LEAST IMPORTANT FACTORS.

The following 10 factors as displayed in ***Table 22*** represent what was categorised as the "Least Important" factors that project mangers considered.

TABLE 22. LEAST IMPORTANT FACTORS.

Least Important	
Control Issues	25%
Have done a similar Project	25%
Turnover (staff)	25%
Competency	13%
Design by committee	13%
Excess staff	13%
Team breakdown	13%
Testing	13%
Why / justification	13%
Over Commitment	0%

Many of these mentioned factors are concerning the project team itself, one question asked by 2 of the interviewees was "Have we done a similar project in the past?" (*Have done a similar Project*) if they had, they would look up the particulars on that project. Staffing problems were addressed (*Competency, Turnover (staff), Excess staff* and *Team breakdown*) by a very few project managers, they seemed to assume that they would always have the required staff as needed.

In the organisation, issues as to ownership, control of information, hardware, procedures etc was addressed (*Control Issues*). In this particular scenario the presented case was a government agency with complex requirements which had been decided by committees and sub-committees, this factor (*Design by committee*) was only addressed by one participant. The question of why is the project being conducted (*Why / justification*) was also raised, with the argument that just because the government wants it is not reason enough.

The commitment (*Over Commitment*) factor (Flowers, 1996, 1997) was not addressed at all by any of the participants, could this be proof that Western Australian workers are not prone to working more than necessary or simply that none of the participants considered the issue important.

5.2. *QUESTION 2.*

"How accurate are the existing literatures on this topic as contrasted against the data collected from project managers within the Western Australian industry on what they consider to be important?"

The results of this study will be compared against those critical factors as identified in Flowers (1996; 1997)

Flowers.

While as previously mentioned, the original quotation of this study (Flowers, 1996) contained no percentages, on a paper the following year (Flowers, 1997) he identified which case studies had what failure factors, using this data percentages for Flowers Critical factors were produced ***Table 34. Flowers (1997) latest factors breakdown*** (Page 6-134 of this dissertation). While some discrepancies are evident between the two sources with factors missing and others added it was decided to use the figures produced.

Table 23 below displays the Flowers Critical Failure factors, the percentage of Flowers (1996; 1997) studies and the actual figures identified by this study. Only the factors addressed by Flowers are in the table.

TABLE 23. FLOWERS (1996)(1997) VS MARTIN

ID	Flowers (1996) data.	Flowers (1997) data	Flowers	Martin
F1	Hostile Cult	Fear Based culture	50%	75%
F2	Poor Reporting	Poor Reporting	83%	38%
F3	Over Commitment	Over Commitment	67%	0%
F4	Politics	Politics	50%	88%
F5	Technology focused.	Technology focused	50%	75%
F6	Lure of leading edge	Leading edge system	100%	38%
F7	Complexity underestimated	Complexity underestimated	0%	50%
F8	Poor consultation	Poor consultation	50%	50%
F9	Design by committee	N/A	0%	13%
F10	Technical fix.	Technical fix sought	33%	75%
F11	Poor procurement	Weak procurement	50%	38%
F12	Turnover (staff)	N/A	0%	25%
F13	Competency	N/A	0%	13%
F14	Communication	N/A	0%	38%
F15	Deadlines	Project timetable slippage	100%	88%
F16	Testing	Testing	33%	13%
F17	Training	Training	50%	50%
F18	N/A	Changing requirements	50%	0%
F19	Development type (XF8)	Development sites split	33%	38%

The first and obvious fact by looking at figures is the vast difference between Flowers and Martin. While the overall average difference is very small it does not represent the difference fairly, and thus was omitted from this table.

5.2.1. Same as Expected.

- **F8.** Poor consultation.
- **F17.** Training.

From the data collected that were the same as expected it can be concluded that project managers within Western Australia are comparable to those in the United Kingdom in the areas of foreseeing problems with training and Poor consultation.

5.2.2. Greater than Expected.

- **F1**. Fear based / Hostile Culture.
- **F4.** Politics.
- **F5.** Technology focused.
- **F7.** Complexity underestimated.
- **F10.** Technical Fix sought.
- **F19**. Development type.

From the data collected that were greater than expected it can be concluded that project managers within Western Australia are more conscious of the culture of the organisation, the political battles that occur, aware that technology is often used to fix managerial problems, systems complexity is often underestimated, and they are more aware of the development type than their counterparts in the United Kingdom.

5.2.3. Less than Expected.

- **F2.** Poor Reporting.
- **F3.** Over Commitment.
- **F6.** Lure of Leading Edge.
- **F11.** Poor / weak procurement.
- **F15.** Deadlines / Project timetable slippage.
- **F16.** Testing.
- **F18.** Changing requirements.

From the data collected that were less than expected it can be concluded that project managers within Western Australia are more used to companies who have better reporting structures, whose workers are not known for their over commitment, project components are not state of the art or leading edge technology, they suffer less problems with procurement of hardware when needed, less prone to deadline and timetable slippage, do less testing, and have the system requirements changed less frequently than do those project managers in the United Kingdom.

5.2.4. FLOWERS, NO FIGURE AVAILABLE.

- **F9.** Design by committee.
- **F12.** Staff turnover.
- **F13.** Competency.
- **F14.** Communication.

No comparisons can be done on these four factors, nor any of factors identified which Flowers(1996; 1997) did not address in his Critical Failure Factors.

This set of factors, it is believed by the author is not a good reference point to start from when looking at what makes or breaks a project. He (Flowers) missed 5 of the 11 factors (***Table 20***) identified in this study as Most Important ones within Western Australia (***Table 24***).

TABLE 24 MOST IMPORTANT FACTORS FLOWERS MISSED

Most Important	
Duration / size	100%
Business rules / processes	75%
Documentation of plan	75%
Geography	63%
Response time	63%

Most of Flowers factors were ranked in the ***Important*** (***Table 21***) and ***Least Important (Table 22)*** categories in this study. Even in the Important category there were 4 factors identified that he failed to mention.

While many factors were shown, it is the belief of the author that the existing critical failure / success factors in circulation as inadequate for the Western Australian industry. The factors addressed by Flowers (1996) are by far the best that could be found for this study, it is far closer than any other collection of factors that were found.

5.3. QUESTION 3.

"What are the most commonly used rectifying actions that a project manager is likely to introduce to expedite a late or troubled project within the Information Systems industry?"

Table 19 in the preceding chapter, has for readability been reproduced here and broken down into ***Table 25, Table 26,*** and ***Table 27*** which represent the most commonly used expediting actions which project managers in Western Australia use to rectify troubled projects.

5.3.1. MOST COMMON EXPEDITING ACTIONS.

From the study, 13 actions were categorised in the "***Most Common***" (***Table 25***) type. All of the project managers identified the need to put in additional resources to the project (Extending the Schedule, More Funds, More People), these three actions could ultimately be broken down to simply "More Funds" as adding people or time ultimately comes down to spending more money. They also identified the need for refinement of their project management procedures / policies for both this project and future projects (Better Project Management Procedures). All participants also recognised that staffing issues were critical to their projects (Staff Issues), these include recruitment, moral, stress, etc,.

TABLE 25. MOST COMMONLY USED EXPEDITING ACTIONS.

Most Common	
Better Project Management Procedures	100%
Extending the Schedule	100%
More Funds	100%
More People	100%
Staff Issues	100%
Staff Issues	88%
Better Development Methodologies	75%
Slippage concern	75%
Stockholder issues	75%
Abandoning the Project	63%
Adjust / redo plan	63%
New Outside Help	63%
Pressure on Suppliers Payment	63%

Some actions which concern the running of the project were discovered, reducing the expectations / requirements / deliverables of the project (Better Development Methodologies) was addressed by several participants, delays in the project (slippage concern) time-line was discussed with some project managers and some managers brought up the need to adjust the project plan to reflect the changes in the project (Adjust / redo plan), some issues over wither to change the entire project time-line to reflect the current delays, or simply to replan those late activities were brought up in this stage. The decision to cut ones losses and abandon the project (Abandoning the Project) was also discussed.

Some identified actions were categorised as management issues, that is, issues that the project manager himself may not have to deal with (though he may have to), rather his manager / management team. While the project manager may interact with the stockholders (Stockholder issues), sometimes higher management may need to become involved, particularly if the project manager is the problem, this category also included normal stockholder reports and meetings.

The need to find both experienced replacement staff and consultants to the project who can perform any desired task may fall upon upper management (New Outside Help), an example could be an auditor to check the progress of the project. If suppliers fail to supply the deliver their orders on time, some pressures, contractual, legal, financial and promises to cancel contracts (Pressure on Suppliers Payment) may be required, of the project managers interviewed, those who brought this up said they would personally not do this, their management would.

5.3.2. COMMON EXPEDITING ACTIONS.

The 3 actions that were categorised as belonging in the "Common" type of actions are displayed below in ***Table 26***.

TABLE 26. COMMONLY USED EXPEDITING ACTIONS

Common	
Contract reviews	50%
Management issues	50%
Risk assessment	38%

The Common actions are all in the management roles, the area of general overall management comes up (Management issues), while covered in the "Most Common actions" it comes up here with different management issues more suited to the project managers level. The scenario brought some problems in procurement and supply, this was addressed by some participants and they mentioned a process of reviewing contracts to see exactly what the term were (Contract reviews), this applied to both vendor and themselves. The last action in this category is continuous risk assessment (Risk assessment), while risk assessment should happen before the project, it should also be an on-going procedure.

5.3.3. Least Common expediting actions.

The following 5 actions were categorized "**Least Common**" from the study are displayed below in ***Table 27***.

Table 27. Least commonly used expediting actions.

Least Common	
Better initial planning	25%
Change of Technology	25%
Never abandon project	25%
Pressure Suppliers by Litigation	25%
Resign	25%

Most of these actions are to do directly with the project, for instance better planning for any project from the start in future (Better initial planning), this won't help this project, but will assist in better management of future ones. Changing they selected technologies of a project (Change of Technology) to use some new, often untested technology. While previously "Abandoning the project" was discussed, this action (Never abandon project) was addressed by project managers whose companies often take on troubled projects and never abandon a project or resign from difficult projects (Resign). While pressure on suppliers and contract review have be discussed previously, this action (Pressure Suppliers by Litigation) is pressure on them by litigation or threat of it.

5.4. .QUESTION 4.

"How accurate are the remedies identified in the existing literature for late or troubled projects as contrasted against the likely remedies as identified by project managers within the Western Australian industry?"

Cole.

While Coles (1995) study is significantly different than this study in several ways, size, collection technique, location, type of participant etc it was decided that they could be compared. On this assumption it was decided that (in absence of evidence to prove otherwise) the percentages of Cole were the percentage of participants who listed that action as important. Using this idea the figures of this study should be generally representative of a greater sample. The figures used are as a guide only, no statistical analysis is possible.

Table 28 below displays the 11 (12 including other) actions which Cole (Cole, 1995) states are actions project managers are likely to take to bring a run-away project into control again and the percentage of participants who address those concerns (***Expected***), compared against what this study found (***Finding***).

While the percentages are not directly comparable statistically (as stated previously) they will compared against each other to indicate if this studies results are similar to that of Cole.

TABLE 28. COLES (1995) VS MARTIN

Ranking	Action	Expected	Finding
C1	Extending the Schedule	85%	100%
C2	Better Project Management Procedures	54%	100%
C3	More People	53%	100%
C4	More Funds	43%	100%
C5	Pressure on Suppliers Payment	38%	63%
C6	Reduction in Project Scope	28%	88%
C7	New Outside Help	27%	63%
C8	Better Development Methodologies	25%	75%
C9	Pressure Suppliers by Litigation	20%	25%
C10	Change of Technology	13%	25%
C11	Abandoning the Project	13%	63%
C12	Other	9%	N/A

From this data, as well as the 10 actions not identified by Cole (1995) the actions will be placed into the below four headings:

5.4.1. SAME AS EXPECTED

No actions fitted into this section.

5.4.2. Greater than Expected.

All of the actions matched against Cole (1995) in this study fell into this section.

From this fact one could say that project managers in Western Australia are better at handling run-away project than those of their United Kingdom colleagues. This could also be explained away as the fact that this study surveyed project managers only, where as Cole did a variety of high management positions.

One fact that is predominant in this study is that all project managers still pour more resources, man-power in particular into run-away projects. This conflicts greatly with Brook's Law (Brooks, 1975) which states that you don't add man-power to a late software project.

5.4.3. Less than Expected

No actions fitted into this section.

5.4.4. Not Expected at all.

Table 29 below is a reproduction of ***Table 18*** which was brought here for readability. The list has been broken down into Importance levels as it was for Coles' (1995) actions.

TABLE 29. ACTIONS NOT IDENTIFIED BY COLE.

Ranking	Action	Coles	Actua l
	Most Important.		
XC1	Staff Issues	N/A	100%
XC2	Stockholder issues	N/A	75%
XC3	Slippage concern	N/A	75%
XC4	Adjust / redo plan	N/A	63%
	Important		
XC5	Management issues	N/A	50%
XC6	Contract reviews	N/A	50%
XC7	Risk assessment	N/A	38%
	Least Important		
XC8	Never abandon project	N/A	25%
XC9	Resign	N/A	25%
XC10	Better initial planning	N/A	25%

This information was previously been covered, it is surprising XC1, XC2, and XC3 were omitted from Coles work.

6 CONCLUSIONS.

The results of this study show that experienced Project Managers readily identify most of the likely foreseeable problems that could arise when given the task of developing a large Information Systems project. Those factors raised closely mirror those factors addressed by Flowers'(1996; 1997) General factors underlying Information Systems Failures, as well as the specific factors identified in a post hoc analysis of the Mandata scenario as reported in Sauer (1993).

Most of the factors (89%) addressed by Flowers' (1996; 1997) in the planning stage were identified by the project managers within this study, ***Table 23*** on page 5-99 of this dissertation shows the difference between the studies. Within the Western Australian sample the trend shows that there are less testing, reporting, requirements changes, deadline issues than those in the United Kingdom. Alternatively the Western Australian sample had more emphasis on technology, politics, hostile cultures, under-estimation of complexity and more prone to looking for technical fixes for management issues.

This studies participants also raised some issues which Flowers(1996; 1997) failed to mention, some of those are;

- Duration / Size of project
- Sponsorship from upper management.
- Documentation of plan.
- Business rules / Business processes of existing business.
- Geography, the vast distance involved.
- Better knowledge of existing and desired systems.

The project managers in this study had all their responses listed and make into a table so that it could be seen which factors they thought most important. The results of this are in ***Table 16*** on page 4-82 of this dissertation. In summary they considered the duration most important (100% of respondents) followed by Deadlines, Politics and Sponsorship (88%), then Business rules / processes, Documentation, Hostile culture, Technical fix and Technologically focused (75%)

From this study it shows that project managers have no problem forecasting possible problems which may interrupt, hamper or disrupt their projects.

Similarly, for remedial expediting actions that a project manager could use when a project is in difficulty, the study showed that the project managers considered a wide range of measures to both manage the current problem as well as contain any future problems which could arise because of it. All of the factors cited in Cole (1995) were addressed by the project managers withing this study, infact in all categories the findings exceeded the expected ***Table 17*** on page 4-83 of this dissertation shows the comparisons. While the studies cannot be directly compared due to the nature and size of each study, the results can be used as a general indicator.

Other issues were raised that could not be classified into the Cole (1995) factors, these were factors such as;

- ➢ Staffing Issues
- ➢ Stock Holder Issues
- ➢ Concerns over Slippage of deadlines
- ➢ Adjustment / redoing the project plan

The 'mythical man-month' fallacy as written by Brooks (1975) was not found to be true in this study. All eight experienced project managers didn't hesitate to add more fuel to the fire by adding more man power to the late project.

Returning to the original question: Why do systems failures in the Information Systems industry continue to occur with the same recurring factors? These present results were compiled by analyzing the knowledge of a few, yet highly experienced Information Systems project managers. From the data collected it tentatively suggests that the reason the Information Systems projects fail is not because the project managers cannot identify the likely problems in advance. This suggests that the problem lies elsewhere, a lack of experienced project managers, the project factors are not known before hand or the factors involved are non-controllable by the project manager.

7. REFERENCES.

Ackoff, R. L. (1994). It's a Mistake. *Systems, 7*, 3-7.

Ambaye, D., & Hayman, A. (1995). *Causes of IT Failure in Teams.* Paper presented at the Third European Conference on Information Systems, Athens, Greexe.

Baker, B., Murphy, D., & Fisher, D. (1983). Factors affecting project success, *Project Management Handbook.* New York: Van Nostrand Reinhold Co.

Baker, F. T. (1972). Chief programmer team management of production programming. *IBM Systems Journal, 1*, 53 - 73.

Belassi, W., & Tukel, O. I. (1996). A new framework for determining critical success/failure factors in projects. *Int. Journal of Project Management, 14*(3), 141 - 141.

Bennatan, E. M. (1995). *Software project management : a practitioner's approach* (2nd ed.). London ; New York: McGraw-Hill.

Boehm, B. (2000). Project termination doesn't equal project failure. *Computer, 33 - 9*(Sept 2000), 94 - 96.

Boehm, B., Clark, B., Horowitz, E., Westland, C., Madachy, R., & Selby, R. (1996). The Cocoma 2.0 software cost estimation Model. *American Programmer*(July 1996), 2 - 17.

Boehm, B. W. (1981). *Software engineering economics.* Englewood Cliffs, N.J: Prentice-Hall.

Brooks, F. P. (1975). The mythical man-month : essays on software engineering / Frederick P. Brooks.

Brooks Jr, F. P. (1975). The Mythical Man-Month (essay), *IEEE - Tutorials* (Vol. 1997, pp. 64 - 71). Danver: IEEE.

Bryan, E. G. (1994). Not All Programmers are Created Equal. *IEEE - Tutorial*(1997), 346 - 355.

Bullinger, H.-J., Warschat, J., & Fischer, D. (2000). Rapid product development - An overview. *Computers in Industry;*, *42*(3), 99-108.

Chand, A. (2000). India: Hot spot for IT outsourcing. *Call Center Solutions, 19*(1), 80.

Chaney, L., & Lyden, J. (2000). Making U.S. teams work. *Supervision, 61*(1), 6-8.

Chase, N. (1999). Learning how to lead a virtual team. *Quality, 38*(8), 76.

Cleland, D., & King, W. (1983). *Systems Analysis and Project Management*. New York: McGraw Hill.

Cole, A. (1995). Runaway Projects - Cause and effects. *Software World, 26*(3).

Curtin University. (1999). Lecture notes for Software Engineering.

Cusumano, M. A., & Selby, R. W. (1996). *Microsoft Secrets*. Glascow: Harper Collins.

Cusumano, M. A., & Selby, R. W. (1997). How Microsoft Builds Software. *Communications of the ACM, 40*(6 / June 1997), 53 - 61.

Denies, S. (1988). IT not responsible for project failures: Study. *Computing Canada, 24*(11), 6.

Feibus, A. (1998). Manage your Project's Requirements. *Informationweek*(Oct 19, 1998), 100 - 108.

Flowers, S. (1996). *Software failure, management failure*. Chichester: John Wiley & Sons.

Flowers, S. (1997). Information Systems Failure: Indentifying the Critical Failure Factors. *Failure and lessons learnt in IT management, 1*, 19-29.

Fortune, J., & Peters, G. (1995). *Learning from failure : the systems approach*. Chichester ; New York: Wiley.

Glass, R. L. (1998). *Software runaways*. Upper Saddle River, N.J: Prentice Hall PTR.

Hanks, P., & Wilkes, G. A. (Eds.). (1986). *The Collins English Dictionary* (2nd ed.). Sydney: Collins.

Johnson, J. (1995). Creating CHAOS. *American programmer*(July 1995), 3 - 7.

Jones, C. (1994). *Assessment and control of software risks*.: Prentice Hall, Yourdon Press.

Jones, C. (1995). Risks of Software System Failure or Disaster. *American Programmer*(March 1995), 2 - 9.

Jung, C. G. (1933). *Modern man in search of a soul*. London: Kegan Paul Trench Trubner.

King, J. (1997). Project management ills cost businesses plenty. *Computerworld*(22 Sept, 1997), 6.

Lam, W. (1998). *Computer-Aided Software Estimation*. Paper presented at the World Multiconference on Systemics, cybernetics and Informatics, Orlando, Florida.

Locke, D. (1984). *Project Management*. New York: St Martins Press.

Lyytinen, K., & Robey, D. (1999). Learning failure in Information systems development. *Information systems Journal, 9*, 85 - 101.

Mall, R. (1998). Software Project Management. Curtin University: Curtin Uni.

Mantei, M. (1981). The Effect of Programming Team Structures on Programming Tasks. *Communications of the ACM, 24*(No 3 Mar 1981), 106 - 113.

Martin, C. C. (1976). *Project Management.* New York: Amaco.

Martin, I., & Smith, W. (2000, November 2000). *Problems in Planning for multi-party systems: why does history repeat itself.* Paper presented at the Working for E-Business, Perth.

Morris, P., & Hough, G. (1987). *The Anatomy of Major Projects.* New York: John Wiley and Sons.

Park, R. E. (1996). A Manager's Checklist for Validating Software Cost and Schedule Estimates. *American Programmer*(June 1996), 30 - 41.

Pinto, J. K., & Slevin, D. P. (1989). Critical success factorsin R&D projects. *Res Technol Management*(January - Febuary), 31 - 35.

Salt, J. (1999). Planning, Estimating & Project Management [1999.

Sauer, C. (1993). *Why information systems fail : a case study approach.* Henley-on-Thames: Alfred Waller.

Sayles, L., & Chandler, M. (1971). *Managing Large Systems.* New York: Harper and Row.

Stackman, H. (1968). Exploratory studies compating online and offline programming performance. *Communications of the ACM, 11*(1 - January).

Thayer, R. H. (1997). Software Engineering Project Management: The Silver Bullets of Software Engineering. In J. Butler (Ed.), *IEEE - Tutorials* (Vol. 1997, pp. 504 - 505). Danvers: IEEE.

Thomsett, R. (1996). Double Dummy Spit and other estimating games. *American Programmer*(June 1996), 16 - 22.

Weinberg, G. (1971). *The psychology of Computer Programming.* New York: Van Nostrand Reinhold.

Weinberg, G. M. (1991). *Quality software management - Volume 1*. New York, N.Y: Dorset House Pub.

Yourdon, E. (1997). *Death March*. New Jersey: Prentice Hall.

Zawacki, R. A. (1985). How to Pick Eagles. *Datamation*(Sept, 15, 1985), 115 - 116.

8. APPENDIX 1. THE MANDATA PROJECT.

PART A)

A developing, geographically large, newly democratic government has in its wisdom decided to implement a project that will change the way the government administers certain aspects of some departments.

At present the country's main service industries, Telephone, Utilities, Postal Service, Public Transport are all government owned. There are rumours of some privatisation of these services in the near future. As well as these services there is a main government office and each of the country's 8 regions also has a regional government.

At present, the pay records, employee records, job descriptions etc are located in each department's location. Clerks are employed to update staff and pay records, Mostly paper index cards to store the data. This task alone employs many people; and many mistakes are made costing the government vast sums of money.

Using a typical example: the Postal Service has 150,000 staff members. Of these, an estimated 1860 full-time staff, in over 400 separate locations, are dedicated to the keeping of personnel records.

The government has made a new department, the Public Service Management Department to develop a way for the government to centralise all of its administrative functions like pay, vacancies, and employment details, to save money and make the process more efficient.

An initial feasibility study has been done, and some interim planning. An initial estimate of 7 years has been made (total project duration), this being due to the complexity of design, and the time to develop new technologies that at present do not exist.

To summarise some of the main requirements.

The system must be ultra-secure. The government is concerned to avoid 'leaking information'. The specification calls for all transactions to encrypted at a very high bit count. This encryption software needs to be developed.

The system must be usable by clerks located all over the country in different departments of government on a real time basis.

The system must take no longer than it would take to manually look up the record in the old system, estimated at some 3 – 7 seconds.

The system must be flexible enough to add/remove components of the programs as needed. For example, to handle changes in employment law or award rates.

The system must be easily understood by the end user and work under a commonly defined operating system.

PART B)

The project is now been running for two years, a number of problems have arisen:

Looking at the projects plan, you see that at the two year mark you should be much further forward than you are. The level of work done matches what should have been completed by 18 months.

The moral of your key project staff is low, and you have staff leaving the project for other jobs.

Software and hardware components are being delivered late or occasionally not at all.

The system for pay grades of public servants and their leave components is far more complicated than at first thought, this is due to each department personalising their own pay system.

Several departments who were initially involved in the project have changed their minds. They are not willing to wait on the system as it is taking too long and does not meet their needs. They are opting towards developing their own systems. An argument they are using is they are looking at privatisation.

A new government has been elected, one who does not as supportive as the previous one. Government bodies are investigating the project and generally upsetting the staff involved in the project.

As the project manager in charge of this project, what possible actions could you foresee yourself doing to bring the runaway project back under control?

9. APPENDIX 2. MANDATA FAILURE FACTORS

Table 30 Factors affecting the success/failure of the 'Mandata' project during phases of initiation and initial development (1971-1976), adapted from Sauer (1993), ***Table 31*,** shows which factors the interviewees addressed during the study.

TABLE 30. MANDATA FAILURE FACTORS

S1	Commitment of the various departments to the project.
S2	Control over the government departments to carry out tasks for the project.
S3	The potential of withdrawal of departments to build their own system
S4	The existence of a competing system (the Treasury was building a distributed payroll system).
S5	Bureaucratic decision-making caused by the hierarchical structure of the body governing the project.
S6	The need for specific resources to be assigned to the consultation processes between the various parties involved.
S7	Staff shortage caused by lack of available skilled IT professionals.
S8	Early flaws in the design which are inevitable but which need to be corrected.
S9	Physical accommodation of system which was mainframe based
S10	Delay in the supply of components from manufacturers.
S11	Flexibility of contracts with suppliers, so that arrangements can be adjusted to fit progress.
S12	The need for reviews to be built into the project so that early problems can be formally identified.
S13	Economic crisis in Australia in the mid 1970s.
S14	Change of government following the demise of Whitlam.

TABLE 31. SAUERS' FACTORS IDENTIFIED IN INTERVIEWS/

	Interviewees								
	I1	I2	I3	I4	I5	I6	I7	I8	proportion
Years of experience in IT/IS field	21	17	10	18	21	14	26	15	considering factor
Sauer's factors Vs interviews									
S1	✓	✓	✓	✓	✓	✓	✓	✓	100%
S2	✓		✓		✓				38%
S3	✓	✓	✓	✓	✓	✓	✓		88%
S4									0%
S5	✓		✓			✓	✓	✓	63%
S6		✓			✓	✓			38%
S7	✓			✓	✓			✓	50%
S8	✓								13%
S9									0%
S10				✓					13%
S11						✓			13%
S12			✓		✓			✓	38%
S13									0%
F14	✓	✓	✓	✓	✓	✓	✓		88%

10. APPENDIX 3. FLOWERS COMPARSONS

TABLE 32. RAW DATA FOR PART A.

		Interviewees								
		I1	I2	I3	I4	I5	I6	I7	I8	proportion
Years of experience in IT/IS field		21	17	10	18	21	14	26	15	considering factor
Critical Failure Factors, FLOWERS (1996)										
F15	Deadlines	✓	✓	✓	✓	✓	✓	✓		88%
F4	Politics	✓	✓	✓		✓	✓	✓	✓	88%
F1	Hostile Cult		✓	✓		✓	✓	✓	✓	75%
F10	Technical fix.	✓	✓	✓	✓	✓			✓	75%
F5	Technology focused.	✓	✓	✓	✓		✓	✓		75%
F7	Complexity underestimated	✓	✓					✓	✓	50%
F8	Poor consultation		✓		✓	✓	✓			50%
F17	Training		✓	✓		✓			✓	50%
F14	Communication					✓		✓	✓	38%
F6	Lure of leading edge		✓				✓		✓	38%
F11	Poor procurement			✓		✓	✓			38%
F2	Poor Reporting	✓	✓						✓	38%
F12	Turnover (staff)				✓			✓		25%
F13	Competency				✓					13%
F9	Design by committee						✓			13%
F16	Testing								✓	13%
F3	Over Commitment									0%
Other factors not in Flowers										
		✓	✓	✓	✓	✓	✓	✓	✓	100%
	Sponsorship	✓	✓		✓	✓	✓	✓	✓	88%
	Documentation of plan	✓		✓	✓	✓	✓	✓		75%
	Business rules / processes	✓	✓	✓			✓	✓	✓	75%
	Response time	✓	✓		✓	✓		✓		63%
	Geography		✓	✓		✓	✓		✓	63%
	Better system knowledge	✓	✓			✓			✓	50%
	Development type	✓						✓	✓	38%
	Change management	✓		✓					✓	38%

	Existing infrastructure		✓				✓	✓		38%
	Control Issues	✓				✓				25%
	Have done a similar Project			✓					✓	25%
	Excess staff							✓		13%
	Team breakdown								✓	13%
	Why / justification								✓	13%

11. APPENDIX 4. COLES COMPARISONS.

TABLE 33. RAW DATA FOR PART B.

		Interviewees								
		I1	I2	I3	I4	I5	I6	I7	I8	proportion
Years of experience in IT/IS field		21	17	10	18	21	14	26	15	considering factor
COLES FACTORS (Cole, 1995)										
C2	Better PM. Procedures	✓	✓	✓	✓	✓	✓	✓	✓	100%
C1	Extending Schedule	✓	✓	✓	✓	✓	✓	✓	✓	100%
C4	More funds	✓	✓	✓	✓	✓	✓	✓	✓	100%
C3	More staff	✓	✓	✓	✓	✓	✓	✓	✓	100%
C6	Scope change	✓	✓	✓		✓	✓	✓	✓	88%
C8	Better Dev. Methodologies	✓			✓	✓	✓	✓	✓	75%
C11	Abandoning Project		✓	✓	✓		✓		✓	63%
C7	New Outside Help	✓	✓	✓			✓	✓		63%
C5	Pressure on Suppliers	✓	✓	✓	✓		✓			63%
C10	Change of technology				✓				✓	25%
C9	Litigation		✓		✓					25%
OTHER FACTORS										
	Staff Issues	✓	✓	✓	✓	✓	✓	✓	✓	100%
	Stockholder issues	✓	✓			✓	✓	✓	✓	75%
	Slippage concern	✓			✓	✓	✓	✓	✓	75%
	Adjust / redo plan	✓	✓				✓	✓	✓	63%
	Management issues	✓		✓	✓	✓				50%
	Contract reviews	✓	✓	✓				✓		50%
	Risk assessment	✓				✓	✓			38%
	Never abandon project					✓	✓			25%
	Resign							✓	✓	25%
	better initial planning							✓	✓	25%

12. APPENDIX 5. FLOWERS FAILURE FACTORS IDENTIFIED BY CASE

TABLE 34. FLOWERS (1997) LATEST FACTORS BREAKDOWN

	LAS	PRS	Taurus	Wessex	TFS	Confirm	%
a. Organisational Context							
Fear Based Culture	Y	-	-	Y	-	Y	50%
Poor reporting Structures	Y	Y	Y	Y	-	Y	83%
b. Management of Project							
Overcommitment	Y	Y	Y	Y	-	-	67%
Political pressures	-	-	Y	Y	Y	-	50%
c. Conduct of Project							
Technology focused	Y	-	Y	Y	-	-	50%
Leading edge system	Y	Y	Y	Y	Y	Y	100%
Complexity underestimated	Y	Y	Y	Y	Y	-	83%
Technical "fix" sought	Y	-	Y	-	-	-	33%
Poor consultation	Y	-	-	Y	Y	-	50%
Changing requirements	-	-	Y	-	Y	Y	50%
Weak procurement	Y	-	-	Y	Y	-	50%
Development sites split	-	-	Y	-	-	Y	33%
Project timetable slippage	Y	Y	Y	Y	Y	Y	100%
Inadequate testing	Y	-	-	-	Y	-	33%
Poor training	Y	-	-	Y	Y	-	50%

- Source (Flowers, 1997, Page 22)

LAS - London Ambulance Service.

PRS - Performing Right Society.

TFS - The Field System.

Information on the above cases can be found in Software failure: management failure (Flowers, 1996) in the following pages:

London Ambulance Service. (Flowers, 1996, Pages 47 - 93)

Performing Right Society. (Flowers, 1996, Pages 7 - 24)

Taurus	(Flowers, 1996, Pages 96 - 123)
Wessex	(Flowers, 1996, Pages 126 - 153)
The Field System	(Flowers, 1996, Pages 126 - 153)
Confirm	(Flowers, 1996, Pages 28 - 45)