Managing large projects: a special case

"Large projects are not just bigger small projects. They are completely different beasts"

Large projects - a definition

There are three major criteria and two secondary criteria involved in defining project[1] size. The major criteria are development effort, development time-scale and, for software projects, application size. The secondary criteria are technology risk and organization impact.

Development Effort

Development effort is typically defined measured in people effort, capital equipment investment and organization on-costs (accommodation, consumables, etc.).

Few major business projects have kept detailed metrics and measures of costs so many of the examples in this paper will be drawn from IT-related projects, which traditionally have been more focused on measuring costs. However, the sizing data from IT projects is relevant for all projects. Similar sizing data is available from the construction, defence and aerospace areas.

Boehm (1984) and Capers Jones (1991, 1995) provide estimating heuristics that put the development effort for large IT-systems at 50,000 work-hours plus or approximately 40 - 50 person years (based on 5 work-hours/day and 210 days/year). In addition, the actual development effort would require additional business professional effort not usually measured by computer-oriented project tracking systems. These people would be in-directly involved in education, physical office re-organization, testing, file conversions, etc.

Assuming no direct capital investment for the project, the project would also consume computer-time and resources for development design, coding, file creation and so on. A reasonable rule-of-thumb could use the

[1] For the purpose of this paper, a programme involving multiple projects is an example of a large project.

standard metric that the Information System budget is split roughly 50/50 between people and equipment, software and services. In which case, each dollar spent on people would be matched by a dollar of computer, software expenditure. Of course, many large projects such as Westpac's CS90, the US IRS system, the Millennium Dome and the Australian Taxation Office Re-development project involve direct capital investment of many hundreds of millions of dollars.

> **Analysis Paralysis**
>
> In 1969, the Australian Department of Defence commenced a super-large project to automate and integrate the supply systems of the three armed services. It used Mil Spec 2617A which demanded that a full requirements analysis must be completed before any design and development could commence. The project was terminated in the mid-1980's and despite hundreds of people working for over 15 years on the project, it was still at the Analysis phase! It had become mired in continually changing requirements to reflect the changed environment.

Combining these figures and using a generally accepted costing of $100.00 per hour for a typical project member, the development cost of large project would exceed $5,000,000. In addition, using the general rule that people/salary costs are around 50% of a typical Information Systems Group budget, another $5,000,000 approximately could be added for capital investment and equipment costs. Finally, most large projects would require substantial effort from various business-professionals which could, in the cases of high organizational impact (see later), equal the direct project effort and costs (that is an other $5,000,000).

In summary, development costs for large projects would be of the order of $15,000,000 and greater.

Development time-scale

Given the 40-50 person year effort required for large projects to build the system, the minimum development time-scale would be around 12-18 months.

However, as indicated by Boehm (op cit), Putman (1976), Brooks (op cit) and others, the scheduling of 50 person-years effort on one single product in a year can lead to inefficient staffing peaks. Therefore, as described later in Project Development Strategies, large projects would be divided into sub-projects (alternatively, projects would be grouped into a single programme). These sub-projects would be scheduled and staffed relatively independently and as a result the development time-scale is generally staggered over a longer period. Staggered development schedules over 2 years or greater is common for large projects.

In a period of unprecedented economic, political and technological turbulence, the longer the project exists over time, the higher the probability that the project will have to accommodate external requirements changes, technology changes and internal changes such as staff attrition and turn-over. In other words, the very requirement for long development time-scales requires large projects to accommodate a high rate of change leading to a further expansion of the time-scale (see box).

Application/software size

Application size can be defined using three related groups of software metric: (1) lines of delivered source code, (2) input, output and file volumes and, (3) IBM's Function Point.

Capers Jones (op cit) defines large projects as those projects which deliver a system of 64,000 LOC or greater. Barry Boehm (op cit) defines large systems as 128,000 LOC or greater. Other sources define a large system as one with more than 40-50 logical inputs, 40-60 outputs, 25-30 inquiry screens and 20 plus logical database views (files). Function Point sizing which is also based on inputs, outputs and so on would suggest that a large system would produce around 1500 Function Points or greater.

Both Jones and Boehm agree that there is a special category of information system - the super-large system which would produce 512,000 LOC or greater.

A large system (for example, one with more than 128,000 LOC or greater than 1500 Function Points) may also be considered as a super-large system should it require complex input, output, file considerations or require complex algorithmic processing.

Technology Risk

The use of leading-edge technology and/or the use of stable technology to the limit of its processing capability is another feature of large projects. In many cases, technology risk is also sufficient for a medium system (64-128K LOC) to be treated as a large system from the management perspective.

Current examples of technology risk would include the use of object-oriented techniques, complex internet technologies, leading-edge programming languages and communications networks for large projects involving high data-volumes or transaction rates.

The stretching of current technology would be typical for large projects. The size of these applications would require transaction rates, database volumes, communications network and systems design which introduce the exponential scale-of-effect multipliers first documented by Brooks in *The Mythical Man-Month* (1975, 1995).

Organizational impact

The final criterion for large projects would be the impact of the project on the organization's strategic growth path and the essential processing units in the organization. Typical large projects would be vital to the survival of the organization and would be a major element in the development of new strategic product.

As a result, large projects require substantial re-organization and rationalization of many sections of the organization. This raises issues of job re-design, union involvement, client service facilities and products, physical office re-design and so on. Peter Keen (1981) is one of the few people who have documented the complex issues associated with the organizational change and it's impact on people associated with large computing projects. In particular, he addresses the difficult political issues that will be raised by the high degree of change. It would be common, as mentioned earlier, for higher costs and impact to be incurred outside the project than directly within the project.

In addition, the development of large projects would require a higher level of intra-organization coordination. For example, because of the wide organization impact, many separate business groups and technical support groups (data-base, network engineering, operations, etc) would be involved in reviewing and approving planning, development and implementation of the various new products and deliverables associated with the project. In many cases, outside organizations such as vendors, contractors, unions and other companies or government departments would also be involved with the project.

In addition, medium-size projects with leading edge technology and/or high organizational impact would tend to behave as large projects and exhibit the same issues. For the purpose of this paper, they will be considered as large projects.

Management Issues

The management of large projects centers on a set of concepts relevant to all projects but which require special focus and variation for large projects:

These concepts are:

- project success;
- project management technique;
- real-time planning and control;
- project team structure and organization;
- project development strategies;
- quality assurance;

- support tools and technology;
- change-control procedures;
- stakeholder management;
- team building; and
- project agreements or contracts.

The Large System Effect

However, before exploring these concepts, we must understand the Large System Effect. In the late 1950's work undertaken in Operations Research during World War II evolved into a broader body of knowledge known as General Systems Theory. Lead by Ross Ashby, Ludwig Von Bertalanfy, Geoffrey Bateson and Stafford Beer and others, General Systems Theory explored the nature of complex systems such as eco-systems, large organizations and the human body. One of the most profound learnings from this research was a simple principle:

The larger the system the harder it is to predict its behavior.

As shown in Figure 1, as systems and/or projects get larger, the number of potential inter-relations and connections between the components (i.e. sub-projects) grows in a non-linear fashion. In projects, the result of this dynamic is that the productivity of the teams drops dramatically as the project size increases. For example, imagine a small project with 2 critical stakeholders. For the project manager to get both those stakeholders into the same room for a meeting may take a couple of phone-calls. Now imagine the same process for a large project with 20 critical stakeholders in a number of different organizational areas. Capers Jones has data drawn from over 3,000 projects that shows order of magnitude differences in the same activity in small and large projects.

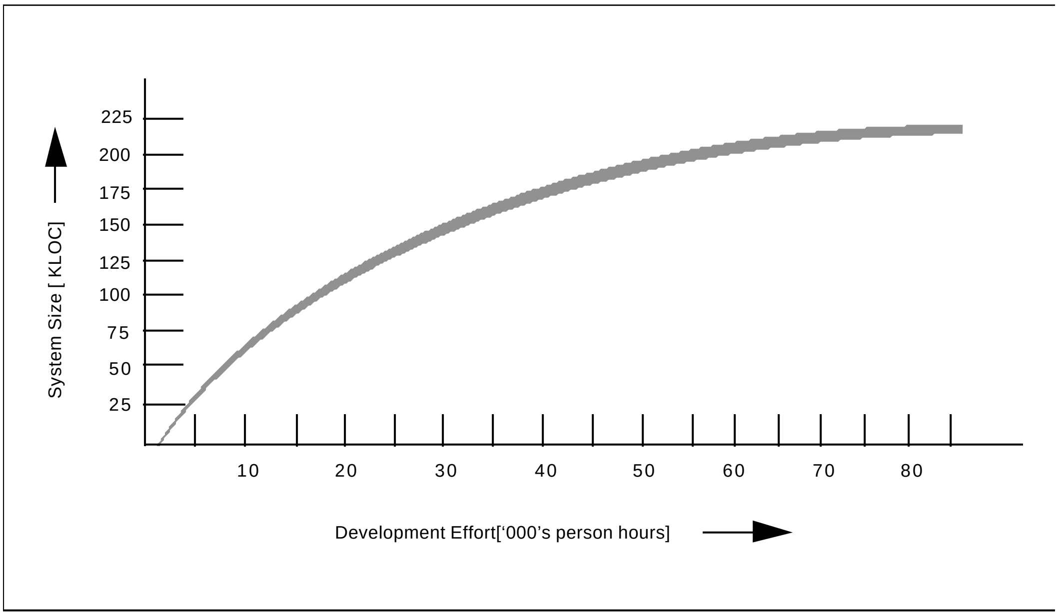

Fig. 1 – Large system effect (from Jones, Thomsett, et al)

Project Success

It has long been considered that a project is successful when it: (a) meets agreed requirements; (b) meets agreed deadlines and; (c) meets budget as measured in people, equipment and so on. However, our group has extended the concept of project success as follows:

- satisfies stakeholders and clients;
- meets project objectives;
- meets budgets;
- meets deadlines;
- adds value (R.O.I.) for the organization;
- meets quality requirements; and
- meets/satisfies teams professional requirements.

For large project, the overwhelming evidence is that the concept of success must be re-considered. As detailed in Appendix A, every development profession (engineering, construction, defense, manufacturing) has its own version of large systems. For example, currently in the Australian construction industry, the new Parliament House, the Sydney Olympics, Darling Harbour projects are clear examples of large projects.

In all cases, the publicly available material indicates substantial blowout of costs (10:1 for the Parliament House, 10:1 for the Olympics, and 5:1 for the Darling Harbour) and missing of deadlines and slippage of project time frames. Further, the Darling Harbour costs exclude the Casino which in January 1988 (the original deadline) was not completed until 10 years later. The Sydney Olympics was originally estimated at $600,000,000 and eventually cost over $6,000,000,000,000. While these may appear to be isolated cases, any detailed research into large manufacturing, defence and construction projects reveals the same pattern of project cost-blow-out and deadline slippage. For example, the RAND Corporation examined the cost over-runs of new Processing Plants in 1981. One of their conclusions was:

Severe underestimation of capital costs is the norm for all advanced technologies[2].

An examination of large computing projects within Australia and America reveals the same pattern. The Australian Public Service's MANDATA, Job-Seeker and Stratplan projects, Bank-America's Masternet (1988), Allstate Insurance (1988), the I.R.S. Migration Project and the Commonwealth Bank's Hogan project all failed to meet the three success criteria. In the case of MANDATA, Masternet and the Hogan project, the projects were abandoned once it became clear that the

A 2000 update

Since this paper was written a few years ago, the incident of super-large project failure has continued. In Australia, Telstra's Jindalee and Flexcab projects, the Mainstream Project undertaken by the Commonwealth Bank, and the CASMAC project for the Australian Universities all failed or are critically over budget and behind schedule. Each project is over $100,000,000. In the U.S.A., the I.R.S. has serious problems with another $400,000,000 attempt project to modernize the tax system, Apple abandoned its COPELAND operating system at around $200,000,000. The list goes on....

[2] Tony Collins is his expose of major computer project disasters (see Appendix A) suggests that the average estimation error is large projects is 3:1. Based on our research, we feel the margin of estimation error is closer to 5:1.

requirements were facing compromise following budget and deadline blowout. Stratplan, Allstate and Job-Seeker have both faced massive cost and schedule over-runs but are still being developed as the requirements for the projects are still achievable.

The conclusion must be drawn that for large projects, inability to meet all the success criteria is not the exception but the rule.

Simply, whereas normal projects could be expected to meet all success criteria, large projects will meet only one or perhaps two of the success criteria i.e. meet requirements or meet deadline or meet budget.

This revised concept of success for large and super-large projects does not imply that budgets and deadlines are simply sacrificed in order to meet requirements. Indeed, as argued later, the application of budget and schedule control is essential to minimize additional blowouts. However, for large projects, it is inevitable that meeting requirements (original and revised) will not be possible within the initial estimates and budget. When this occurs, the budget and schedule are expanded in a controlled process (see Change-control later).

It is critical that the success expectations for large projects are discussed and negotiated openly before the project is commenced.

For example, in implementing the Goods and Services Tax in Australia, most organizations realized that the success criteria of quality and deadline were mandatory. As a result, there was an ability to negotiate whether all systems had to be compliant by the deadline and the budget, team satisfaction and level of stakeholder satisfaction were compromised. A similar pattern existed in most organizations for the Y2K project..

Project Management Technique

Formal and highly disciplined project management techniques are mandatory for large projects.

The use of RAPP's (Rapid Project Planning) developed by our group are mandatory. These structured planning sessions involving the team, stakeholders and related project managers in planning projects and developing the project Business Case have been proven to improve the quality of and the stakeholder buy-in to projects. They also dramatically reduce the elapsed time for planning.

Because of the need for communication across multiple sub-teams and specialist teams within and many stakeholders outside the project team's organization, the detailed scheduling of tasks and documentation of critical project management information - risks, assumptions, key decisions, agreements and costs is essential to provide a total project view.

Further, the use of organizational standards such as development methodologies, naming conventions, data dictionaries and formal Quality Assurance techniques can minimize the communication complexity. For example, it is typical that large projects will involve multiple project development approaches such as prototyping, quick-wins, fast tracking and so on. These various strategies should be formally documented and common amongst all sub-projects.

The role of senior management is also vital in large projects. Whereas many Steering Committees simply act as review groups, because of the broad organizational impact of large projects, the Steering Committee must act as a problem-solving group particularly in the area of cross-boundary or inter-group disputes. Given the strategic importance of large projects, the senior management Steering Committees must become involved at a more detailed level than is typical for Steering Committees. As discussed later, changes are inevitable in large projects and changes to requirements, costs and schedules must be reviewed by the Steering Committee and specific levels of delegation and authority must be in place to ensure that the typical process of gradual and incremental alteration of the project does not occur without Steering Committee approval.

In addition, it is mandatory that a large project must have a powerful and totally focused Project Sponsor.

The author's research indicates that large projects involve additional project management effort than that required for normal projects. It is generally accepted that 10% of the total development effort should be allocated to the processes of project management for normal projects. For large projects, this allocation could rise to the range of 20-30% of the total development effort. In other words, on a large project of 50 people, 10 would be full-time on running the project management system – communicating with stakeholder, planning and re-planning, selecting tasks, estimating, scheduling, tracking effort and deliverables, reviewing and re-scheduling as required. Since much of the project management process involves clerical tasks such as recording, summarizing and aggregating data and driving the automated project scheduling tools (see later), the 10 people allocated to project management would include a number of specialist and clerical support people (a Project or Programme Office). As covered later in this paper, super-large projects also involve significant additional project management effort in managing the external groups involved with the project.

Real-time Planning and Control

This technique is counter-intuitive. Many organizations approach large projects with a belief that the more levels of management involved in the project and the more rigorous the reporting system, the more likely the project will succeed.

Bureaucracy is the cancer of large projects.

Our company and organizations with a proven track record of delivering super-large projects such as Microsoft[3] has proven that to

[3] Microsoft delivered Windows 2000 involving over 35,000,000 lines of code only 18 months late. It is the largest single computer system ever developed.

effectively manage a large project radical management techniques are required.

A specific example of this is the concept of real-time planning.

In a large project, there will be many inter-related sub-projects and multiple related projects in other areas of the organization. Even a small schedule slip or scope change on one sub-project could have complex "ripple-effects" for other projects.

It is mandatory that any change in one project be communicated as quickly as possible to the related projects. A delay of one week could be catastrophic. Our group has seen organizations where the time for a message to move between one project and another project or a project manager and his or her sponsor can exceed 4 weeks. Imagine having a human body where the time between the eyes seeing an approaching car and the brain registering the situation and issuing commands for muscles to begin the process of running took one day!

Fast and open feedback loops between all levels is a critical management issue for large projects.

Real-time planning involves *daily* reviews of all projects and project status by all project managers involved in the project. Our group deployed this technique on a project involved in integrating two large banks. For the initial planning and quick-win phase of 3 months, 26 project managers, the Project Director and the Sponsor met every day at 8.30 am to ensure a maximum of cross-project communication. Microsoft deploys a similar technique including daily builds and integration of the software written by over 500 people.

The War Room

We have seen successful large projects create a central area where all related project plans and other critical project information is kept on walls in a room where all people involved in the project and critical stakeholders can quickly get an overview of the project. The War Room was also used for the daily meetings with project managers and was close to the Programme Office. Coffee, muffins and so on were always on hand.

Project Team Structure and Organization

Clearly the scenario for large projects is one of a project management team with a number of sub-project teams consisting of business professionals, computer professionals, policy and product specialists, clerical support and computer specialists (data-base, communications, etc).

While each sub-project may have its own project manager, the project management team would be responsible for the total project. The central project management team could also include the project-resident technical specialists and the clerical support staff. In other words, the central project management team would be the project's resource centre for specialist functions. This would be in addition to the normal Computer Group specialist areas.

Each sub-project team would be comprised of business and computer professionals working on the component of the overall system being developed by the sub-project. As is now generally accepted, business professionals from the organization areas involved in the project would be full-time members of the project team and, in many cases, senior business professionals would be performing the project management functions.

What is mandatory in large project is to minimize the hierarchy and maximize the communication. As discussed earlier, this may appear counter-intuitive, the history of large projects (see Attachment) indicates a common pattern of many layers of management being involved in making even the most simple of project decisions. The result of this is massive delays in decision-making, substantial filtering and distortion of project communication and, in effect, collapse of the project management system for which the management hierarchy was formed to control and manage.

In essence, the *larger* the project, the *shorter* the communication lines must be to ensure fast-feedback through the project management system to all sub-project teams.

Ideally, a large project should have no more than **two** levels of management between the sub-project team and the Steering Committee. This can easily be accomplished through the use of Likert's (1961) Linking-pin structure. As shown in Figure 2, this structure involves each sub-project team being represented on the project management team and the project management team being directly involved in the Steering Committee team. In other words, the Steering Committee has the project manager as a full-time member and certain team members are involved in two teams (either horizontally or vertically) as the linking-pin.

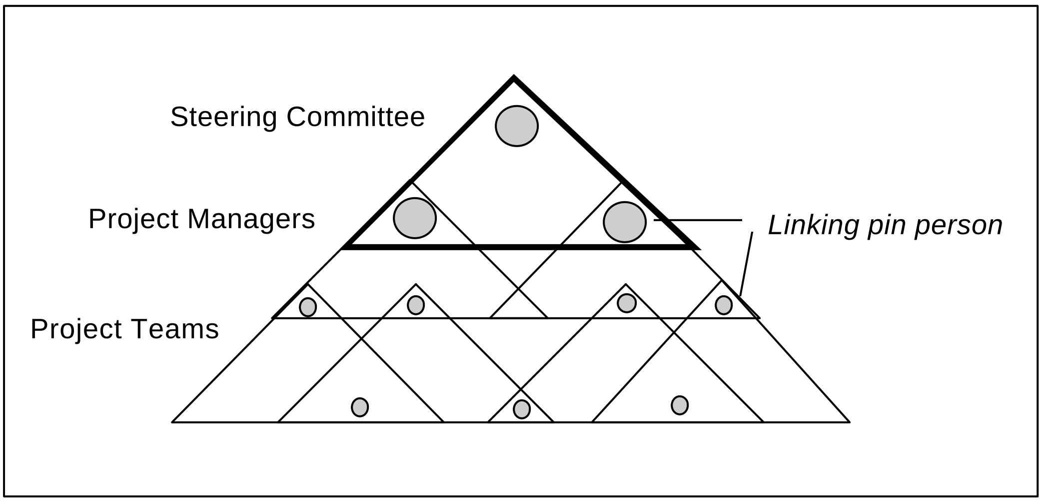

Fig. 3 - Project team structure (option 1)

An alternative model of project organization is to create a totally flat reporting structure. In Option 2, the Project Sponsor, Programme Director and all the Project Managers of each involved project form a virtual team.

Using the real-time planning concept, the Project Management Team meets daily (initially) and weekly (once the projects have been initialized and and in full development. In these meetings, risk management, issues management, project resourcing and other planning concerns are dealt with in a non-bureaucratic structure.

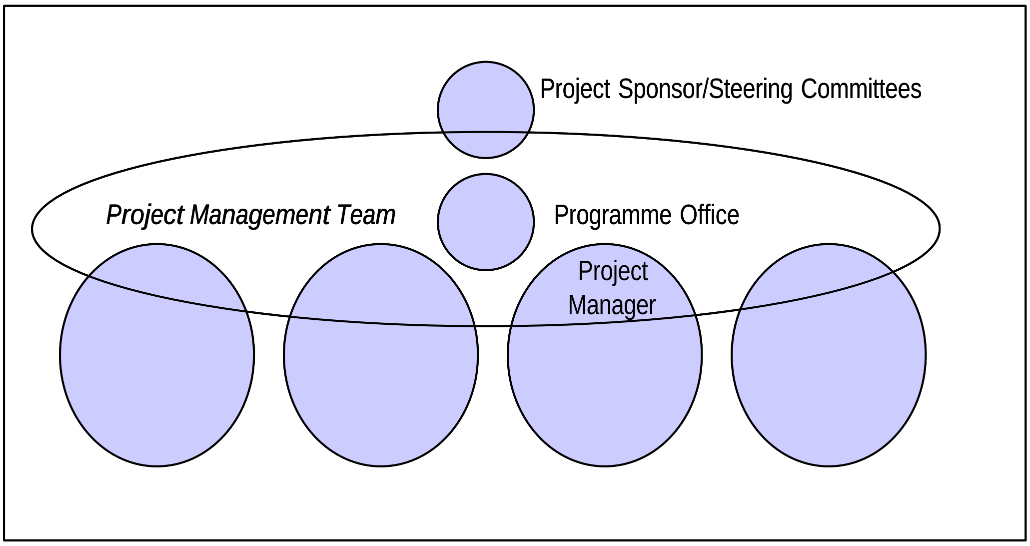

Fig. 4 – Project team structure (Option 2)

A Programme Office must be created to support the Project Director and the Project Managers. It is important to mote the emphasis is on support *not* control. Roles typical of a state-of-the-art Programme Office include:

- Communications Guru - an expert in communication to develop and review all forms of communication between the Programme and stakeholders;
- Product Architect - a person who has extensive product and business expertise to ensure that products involved in the programme are integrated and support the strategy;
- IT Architect - a person who has extensive IT expertise to ensure that technology issues are integrated and that technology issues are co-coordinated;
- Project Management a project management expert who

Guru -	mentors and consults with the Project Managers to ensure professional project management techniques and approaches are being deployed;
• Change Management Expert -	a person expert in change management and other organizational change issues;
• Programme Executive Assistant -	an executive assistant to provide administrative and related support;
• Project Co-coordinator -	a person who integrates the individual project plans and maintains the project status summary.

Further, the use of an electronic mail system dedicated to the project team would be a valuable aid in ensuring that critical messages could be sent to all relevant people. Most electronic mail systems also include an “action-prompt” feature, which would ensure that critical messages were responded to within in an agreed time frame.

Project Development Strategies

A system or project development strategy is the overall partitioning of the project and the high-level sequencing of sub-projects. Robert Melichar of IBM articulated three primary development strategies.

The *monolithic*, waterfall, or classic strategy where all tasks associated with a system development phase (e.g. analysis) are completed before any of the tasks involved with subsequent phases are planned, scheduled or undertaken; the *version* or *release* strategy where after initial systems analysis or modeling, the product being developed is divided into sub-products and each sub-product is developed as a quasi-independent sub-project. There are two "sub" strategies associated with the release strategy. The first is sequential release where one sub-project is completed before the next sub-project is started. The second sub-strategy is concurrent release where the sub-projects are developed by separate teams often being scheduled in parallel.

The final strategy is the evolutionary or *fast track* strategy where the approach is to develop a full production version of the product as quickly as possible. The result would be an undocumented, inefficient and "dirty" production system which could be used while the fast-track development cycle is repeated to stabilize, add or delete features based on the use of the production system. This strategy is the most commonly used in other industries for large or high-risk projects (e.g. the new Parliament House and the Darling Harbour are using this strategy).

However, it is probably more practical to use a combination of all strategies for large projects. For example, the overall system development strategy for large project would be a concurrent release strategy with some of the high-risk sub-projects being developed via fast-tracking while more low-risk sub-projects could be broken into sub-releases and developed using the monolithic strategy.

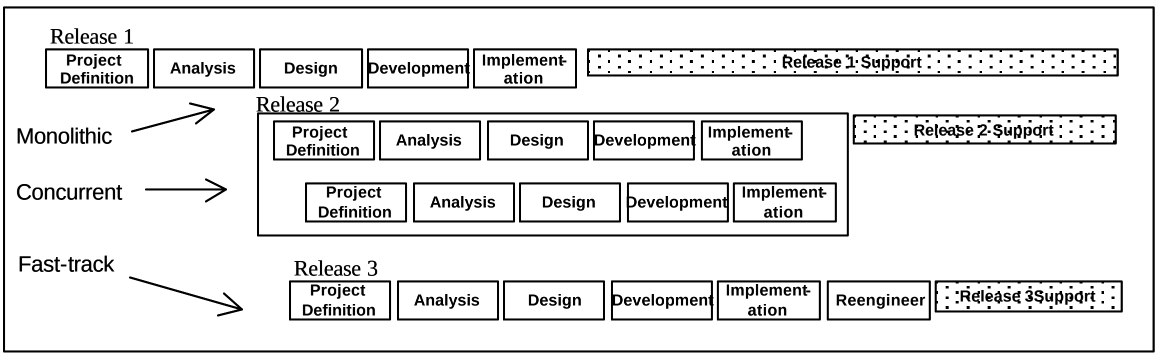

Fig. 4 - Typical large project development strategy

The use of multiple strategies within multiple releases adds further support to the requirement for vigorous and formal project management techniques. Apart from the sheer complexity of scheduling many people to many tasks, the project management system is vital in ensuring that changes in one sub-project (release) do not impact on another sub-project.

Quick Wins

It is critical in large projects to look for delivery of "quick wins". Incremental delivery of early products or prototypes enables the organization to minimize its exposure to long time-frames. It also provides the teams with a sense of accomplishment. Quick wins should also look to early benefits delivery.

Quality Assurance

While formal quality assurance techniques (Technical Reviews, walkthroughs and QA Groups) are apparently only partially implemented in the computing areas of Australian companies, the need for Total Quality Assurance is essential in large projects.

As Capers Jones' (op cit) research shows, the scale of effect issues discussed earlier in this paper has a significant impact in the area of defects in large projects. Simply, the level of defects per line of code is 200% higher for large projects than for small projects and up to 400% higher than for super-large projects.

This has serious implications for the intrinsic quality of the product and the higher level of defect implicit in large projects adds an additional factor to the potential degradation of the requirements through analysis, data-modeling, design, documentation and coding defects. Capers Jones (op cit) has found that the detection and removal of defects throughout all phases of the system development life-cycle can account for 40% of the total development effort for large projects. In other words, without an effective Quality Assurance component, defects can account for around $2,000,000 of a typical large project's development cost!

Apart from the simple issue of defects, the relatively high incident of defect associated with large projects has a major impact on the project management issues. The "ripple effect" of defects passing between dependant project tasks is one of the biggest causes of poor estimation and project slippage.

The solution to this issue is to enforce formal quality assurance techniques (Technical Reviews, Inspections or walkthroughs) at the end of each task. As documented by Sprouster (1984) and Weinberg and Freedman (op cit), it is cheaper and more effective to perform quality assurance in short, sharp sessions conducted by the development teams rather than the prevailing approach in computing (and many other industries - see Sprouster) of large QA sessions conducted at the end of a number of tasks (sub-phase or phase reviews). However, in super-large projects, quality assurance also needs to involve many additional external groups when compared to smaller projects.

Finally, as discussed earlier a feature of large projects is the use of staggered multiple releases of sub-project deliverables into the production environment. While this is a essential feature of the release and fast-track strategies it presents a special problem in the area of production support and defect repair.

It would be typical for the initial production releases of the project to be continually developed or enhanced as well as undergoing the normal post-production defect repair cycle. As a result, changes will be made to production data, data structures, function and documentation. These changes may have a flow-on to the releases still in development. For example, a data structure may be shared between a production release and an in-development release. These changes must be reviewed not only by the production support teams but also by the development teams working on related releases. In other words, production support QA must be shared by both development and production support teams. As a result, it is essential in large projects for the development and production support teams to be fully integrated and co-located to minimize communication breakdowns.

Support tools and technology

Substantial development has occurred over the past few years in the area of automated development support tools (particularly in personal computer- based tools). The development and support of large projects would require dedicated access to the following tools:

- CASE (Computer Aided Software Engineering) - these tools are currently in a period of rapid development. However, all of the widely-implemented products (I.E. Workbench., Excelerator, etc,) provide the ability to record data-models, data-flow diagrams and design diagrams using graphic interfaces. CASE-type products will reduce the effort required to document and maintain the various technical specifications across the various sub-project teams;
- Data Dictionary - it is vital that the data being developed in the project is recorded in a centralized manner. Given that data is shared across the various sub-projects (releases), any person on the team must be able to determine the logical and physical characteristics of the project's data and, in particular, which data is used in which sub-project;
- Desk-top Publishing - Capers Jones (op cit) has data, which shows that documentation and paper-work consumes 30% of a large project's effort (compared with

10% for smaller projects). Just as CASE tools will assist in reducing the effort on documenting the technical diagrams, the use of desk-top publishing tools will assist in the production of business professional and operations manuals, forms production and general system documentation;

- Electronic mail/ Groupware system - as has been discussed earlier, a dedicated electronic mail or Groupware system could facilitate the intra-project communication. Given the need for communication across many sub-teams and related specialist and management areas, this technology could provide a quick, efficient and auditable mechanism for recording and transmitting information;
- Intranet/Internet - these technologies are essential for fast and relatively cheap communication with the large number of stakeholders associated with large projects. The development of a Project Web Site which contains the Business Case, project status reports and so on is a major advantage in stakeholder communications;
- Advanced phone system - the author's research indicates that the typical project person can lose up to 20% of a day in the activities

of answering other project members' phones, taking messages and locating that person. The use of a switchable phone system (or answering machines) where each team-member can switch their phone to a centralized message centre staffed by a clerical support person can lead to substantial gains in productive hours of work per day. In addition, the use of phone-conferencing would assist the electronic mail system in facilitating project team communication; and

- Meeting software - Products such as Zing which enable the gathering and processing of many ideas from project planning sessions.

Other support issues include application development and prototyping tools, automated testing and debugging tools, physical office design, rooms for walkthroughs and team meetings, access to technical libraries and so on.

Stakeholder Management

A stakeholder is a group, organization or person outside the direct organizational control of the project manager. In a small project, the stakeholders would typically include the project sponsor, direct business users of the project's deliverables, operations areas and perhaps some internal or external software development and data administration consultants. In super-large projects, the stakeholder community exhibits the same dis-economy of scale as in the effort required for development.

In a super-large project, the stakeholders would include would include groups both within the organization such as Data Administration, Network, Operations, multiple client groups and other dependant or interdependent projects and, in most cases, areas outside the organization such as vendors, other companies, unions, government and other major groups. For example, in one project with which the author was involved over 60 external stakeholder groups were identified. This means that much of the project management effort (in excess of 50%) will be devoted to managing the stakeholder relationships in these types of projects. Further, the quality assurance processes such as reviews would need to accommodate the requirement for multiple stakeholder reviews. For example, in a small project, a review of functional requirements may involve 5 -7 people and take 2 hours. In a super-large project, a review of functional requirements may involve 20 - 30 people over a period of 2 days. Clearly, review techniques designed for small group review need to be modified for super-large projects.

The use of the linking-pin as discussed earlier is another vital part of stakeholder management. In particular, linking-pins between the project and related projects ensure that the detailed technical issues such as common function, data, design impacts and so on are managed at the appropriate level.

Team building and maintenance

Because of the long duration of super-large projects, team dynamics are considerably different than those in smaller projects. In three super-large projects with which the author has been involved in over the past three years, the team membership changed completely over the time for development. In one project, the only original team member left upon

> **Maintaining the rage**
>
> Project fatigue is common on large projects. After months of consistent deadline pressure and the “chaos” of changes and so on, many people simply start to “burn out”. It is essential to schedule regular “time-outs” where the entire large project team can get together and celebrate, play, team-build and recharge their batteries.

final delivery was a contractor!

Because of the inevitable turnover in super-large project and other concerns covered in this paper such as the need for highly-disciplined project management, multiple external relations and large team size, the issue of team building and maintenance is one of the key issues in super-large projects. As discussed by Constantine (1995) and Thomsett (op cit), most computing organizations have failed to develop and implement effective models of teams and the processes of team-formation and building are left to random selection based on technical skills and resource availability only. In the case of small projects, a randomly-formed team may manage to produce the required system without having to face the morale and motivation crises that are a documented part of large projects. However, the team dynamics of super-large projects are such that there is a need to ensure that the requisite team roles as described by Thomsett and Constantine are present and that the appropriate team maintenance and support mechanisms are build into the project.

Super-large projects require the deliberate and structured formation of a project culture that provides a common vision for the team members. The project vision is a vehicle for ensuring that the project's objectives and values are linked into the corporate objectives while providing a micro-climate or culture that will sustain the project as team members leave and join throughout its life. In the super-large projects with which the author has been consulting with over the past three years this project culture is manifest in the project management documentation (scope, objectives, quality and so on) as well as in "softer" concepts such as a project logo, project marketing material (brochures, portfolios), formal project induction programmes and regular (usually every 3 months) "time-outs" which combine formal planning and team building exercises.

Development and maintenance of the project culture is a key project management role in super-large projects.

Change-control Procedures

Probably the single most important feature of large projects is the previously mentioned scale-of-effect (Fig. 1). There is a non-linear growth and effort pattern for large projects.

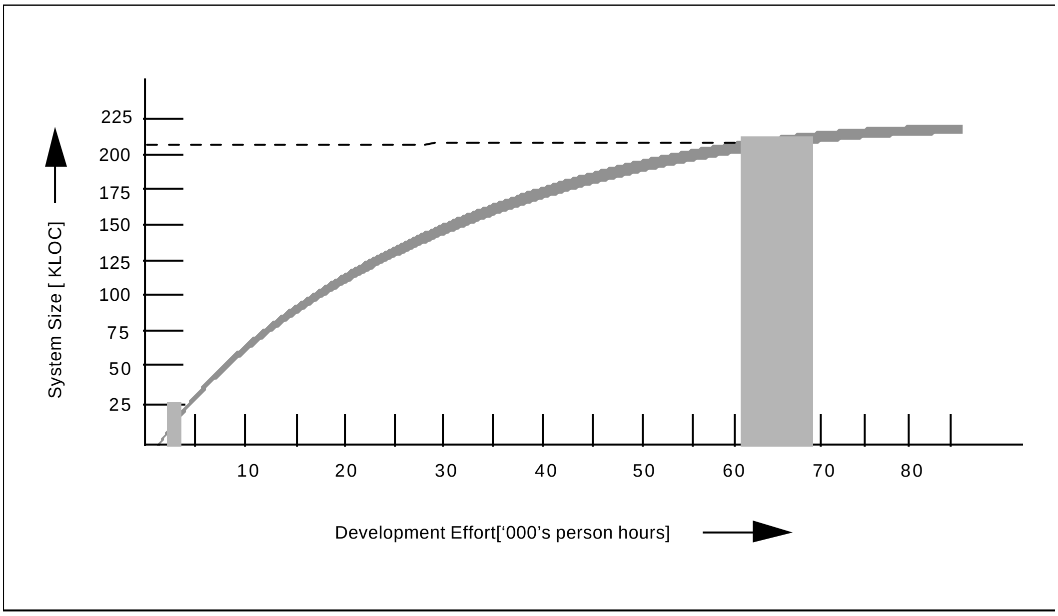

Fig. 6 - Impact of changes

Using the Lines-of-Code in Fig.6, a change in initial requirements resulting in the need to provide an additional 5,000 LOC on a system of base-size of 25,000 LOC would require in the order of an additional 200 - 300 work-hours of effort. To add the same requirement of 5,000 LOC to a large project of 125,000 LOC would require in the order of an additional 6000 work-hours plus!

This leads to the formulation of Rob's Rule of Large Projects:

For large projects, there is no such thing as a small change.

Capers Jones (op cit) refers to another I.B.M. study that indicates that the typical behavior of all information system projects is an expansion of requirements and effort by 50% over the projects development life. Given the scale-of-effect problem, the control of changes in large projects becomes mandatory.

It is normal for a large project to change monthly.

In effect, any change to the project (requirements, costs, effort, staffing and so on) should be treated as a mandatory "freeze" point.

The project or sub-project should be temporarily stopped and a meeting involving business professionals, project leaders and managers, client groups, relevant specialist groups and the team be convened to evaluate the impact on the sub-project and the total project of the change-request. Should the change impact agreed costs, schedules, requirements or impact other sub-projects, then the change should be forwarded to the Steering Committee for approval prior to implementing the change.

Clearly, there will be some changes which can be accommodated within the sub-project manager's scope of control and authority but, all changes to the initial requirements and schedules should be formally documented and reviewed at the regular project manager meetings.

Finally, most change-control procedures are designed to accommodate externally-generated changes. In a typical project there are probably as many internally-generated changes resulting from poor estimation, staff turnover, defect repair and so on. The change-control system must also accommodate these internal changes as well.

Project Agreements or Contracts

As discussed earlier, all large projects necessitate the use of multiple teams that cross traditional inter- and intra-organizational boundaries.

It would be typical for a large project to involve 10 or more separate teams - business professionals, computer professionals, specialist groups, unions, contractors, multiple vendors and so on. In many cases, these teams are not in the direct scope of control of the overall project manager. Often, it would be the case that key tasks on the project's critical path would be under the control of an outside "sub-contracted" area.

As a result, the development and commitment to deliverables, tasks and deadlines involving teams outside the direct control of the project manager must be negotiated and formalized prior to the overall commitment by the project manager and the team to that specific part of the project. For example, should the project manager require consulting and or direct assistance from the Network Design group, the project manager would review the schedule and the level of support required with the Network Design manager. Once an agreed schedule was negotiated, the Network Manager would be required to sign a formal contract to the effect that he or she was prepared to commit to the project's requirements.

The formalizing of what is usually an informal review process is critical in large projects as it ensures that the project manager has considered the broader organizational support required and that the other areas are involved in developing a realistic schedule that recognizes their other on-going work. Of course, all project contracts would be reviewed by the Steering Committee prior to approving the particular component of the project development cycle.

Summary

The management and development of large projects is one of the critical issues in computing over the next decade. The relative lack of published material dealing with this area is symptomatic of the criticality of the issue.

To effectively develop and implement large information systems requires a re-evaluation of traditional management techniques and attitudes. Senior management must become closer to the issues of project management and in their roles as project sponsors and Steering Committees must adopt a "hands-on" approach to the project management and associated techniques and technology. In particular, the management of large projects requires special attention to the following areas:

- project success;
- project management technique;
- real-time planning and control;
- project team structure and organization;
- project development strategies;
- quality assurance;
- support tools and technology;
- change-control procedures;
- stakeholder management;
- team building; and
- project agreements or contracts.

While organizations can expect to gain major benefits from the new technology and delivered services and products involved with the next wave of information system development, should the implementation of the approaches discussed in this paper be inadequate, rather than a “brave new world”, many organizations will be facing a “new nightmare” of system and organization failure of an unprecedented magnitude.

The hubris associated with large systems is well documented in the texts referred to in Appendix A. Failure of large projects is not inevitable however, but the margin of success is narrow and all levels of management must make the effort in time, education, finance and leadership to ensure success.

References

B.W. Boehm, “Software Engineering Economics”, *IEEE Transactions on Software Engineering,* Vol. SE-10, No.1, Jan. 1984, pp. 4-21.

F.W. Brooks, *The Mythical Man-Month*. New York, McGraw-Hill, 1975, 1995.

L. Constantine, *Constantine on Peopleware*. Englewood Cliffs, Yourdon Press, 1995.

C. Jones, *Programming Productivity : Issues for the Eighties*, L.A., IEEE Computer Society, EHO 186-7, 1981.

C. Jones, *Applied Software Measurement,* New York, McGraw-Hill, 1991 & 1995.

P.F. Drucker, *Innovation and Entrepreneurship*, N.Y., Harper & Row, 1985.

P.G.W. Keen, "Information Systems and Organizational Change", *Communications of the A.C.M.*, Vol. 24, No.1., 1981.

R. Likert "The Principle of Supportive Relationships", *Organization Theory*,

(Pugh, D.S. ed), Middlesex, Penguin Books, 1971.

D.A. Ludlum, "$80M MIS Disaster", *Computerworld*, Vol XXII, Feb. 1, 1988, pp1

L. Putman, "A General Empirical Solution to the Macro Software Sizing and Estimation Problem", *IEEE Transaction of Software Engineering*, Vol.SE-4, 1978.

J. Sprouster, *Total Quality Control - The Australian Experience*, Sydney, N.S.W., Horwitz Grahame Books, 1984.

R. Thomsett, "Effective Project Teams: A Dilemma, a Model, a Solution", *American Programmer*, Vol. 3, nos 7-8, July/August, 1990.

Appendix A: Selected Readings

Two excellent books on the behavior of large projects, systems and high-risk technology are Professor Charles Perrow's *Normal Accidents, Living with High-Risk Technologies* (Basic Books, N.Y., 1984) and Professor John Gall's *Systemantics* (New York Times Book Co, 1977). Perrow's book deals with the failure of large projects in construction, engineering, nuclear power and other areas. Gall's is a look at the behavior of large systems with a serious message but presented as a series of simple laws.

Three other books look at specific industries. Mary Kaldor's *The Baroque Arsenal* (Abacus, 1983) details the massive problems in the projects that are developing new weapons technology and Patrick Tyler's *Running Critical - The Silent War, Rickover and General Dynamics* (Harper & Row, N.Y., 1986) is an essential text as it presents both the political, human and project costs associated with the development of Trident submarines. Malcolm McConnell's *Challenger - A Major Malfunction* (Simon & Schuster, London, 1987) is similar to Tyler's book as it presents a detailed account of the poor project management, political and human concerns and massive communication breakdown typical of large projects under pressure of deadlines.

Management Disasters and how to prevent them by O.P. Kharbanda and E.A. Stallworthy (Gower, Brookfield, Vermont, 1986) provides a detailed examination of a large number of large projects including the Bhopal disaster and develops an excellent set of general diagnostics and preventative measures for large projects. Henry Petroski's *To Engineer Is Human* (MacMillan , London, 1985) is another broad-ranging discussion of the engineering problems associated with large projects and the role of management in preventing potential disaster.

John Stringer while at the Australian Graduate School of Management (University of New South Wales) produced a number of papers on the problems of managing large engineering projects. In particular, *Management Problems of Large Engineering Construction Projects* (Working Paper No. 82-005, May 1982) provides an extremely well-researched approach to large project management which has direct implications for large information system projects. The Auditor-General's *Efficiency Report on the new Parliament House* (A.G.P.S.,1988) also contains many examples of the behavior of large projects.

Two other notable articles dealing with large projects are "Divad" by Greg Easterbrook (*The Atlantic*, October 1982 pp 29-39) which records the massive problems associated with the Divad or Sergeant York automated all-weather gun platform and Jim Mintz's "How the Engineers are sinking Nuclear Power" (*Science 83*, June, pp 78 -82) which contains an account of the 10:1 project cost and schedule blow-outs associated with the second wave nuclear power plants in the U.S.

Capers Jones' *Programming Productivity: Issues for the Eighties* (see References) provides articles written by some of the leading experts in software engineering. A number of these articles refer specifically to the special issues of large projects in computing. Notable articles include "On Understanding Laws, Evolution, and Conservation in the Large-Program Life Cycle" by Lehman and "Management Perspectives on Programs, Programming and Productivity" by Kendall and Lamb.

Software Runaways by Bob Glass (Prentice-Hall, 1996) contains detailed descriptions of the Denver Airport fiasco and the multiple attempts by the Internal Revenue Service to re-write the US Tax systems and well as many other large project failures. *Crash* by Tony Collins with David Bicknell (Simon & Schuster, London, 1998) also focuses on large IT project disasters and the learnings gained from them.

Finally, Tracy Kidder's Pulitzer Prize-winning *The Soul of a New Machine* (Avon Books, N.Y., 1981) details the development of Data-General's Eclipse super-mini and contains many examples of the behavior of large projects written with a journalist's flair. Also in the riveting read category are two wonderful books on how Microsoft develops large and highly-innovative systems. Fred Moody's *I Sing the Body Electric* (Penguin Books, New York,

1995) and G. Pascal Zachary's *Showstopper* (London, Warner Books, 1994) are a revelation.