Flight of the Eagle: The Birthing and Life of a Super-Minicomputer.

John Faughnan
Sanja Stevanovic
March 16, 1996
Project Management
john@umnhcs.labmed.umn.edu

TABLE OF CONTENTS

EXECUTIVE SUMMARY 2

INTRODUCTION 3

METHODOLOGY 3

PROJECT DESCRIPTION AND HISTORY 4

PROJECT ANALYSIS 7

CRITICAL EVALUATION AND RECOMMENDATIONS 14

CONCLUSION 17

REFERENCES 19

Executive Summary

In April of 1978 the Eagle Project began at Data General, a Massachusetts based minicomputer company. Two years later, the first Eclipse MV/8000 super-minicomputer was introduced to the world. The Eclipse line would continue at Data General until 1988, prematurely terminated by the forces of commodity computing and the rise of UNIX workstations and personal computers.

The Eagle project, under the direction of Tom West, forged a small number of experienced managers and about 20 recent engineering graduates into a highly effective engineering team. A journalist was accepted into the team, and he documented their successful effort in a Pulitzer prize winning novel — The Soul of a New Machine.

Based on Kidder's work, trade journal accounts, technical articles, and electronic interviews with knowledgeable observers, we've reconstructed the Eagle project in the context of Data General and the burgeoning computer industry of the early 80s. We review the tactical strengths and weaknesses of this project, which probably saved the company from extinction. We examine the effective team building process, and the impact of Data General's deteriorating economic status. We examine project delays, and consider whether they might have been averted by the use of modern planning methodologies.

The principle lessons of the Eagle project may be strategic lessons. This high-technology project's success was cut short by a technological discontinuity: the rise of the commodity workstation and personal computer. We examine the implications for modern high-technology products, and review the lessons Data General learned from the Eagle project and its aftermath.

Introduction

In most histories of computing, Data General (DG) is barely mentioned. Data General was one of many casualties in the Great Wars of computing. IBM's mainframes ruled for decades — an eternity in computer years. Then large scale integrated circuits gave rise to the 1970s minicomputer industry; along with the Digital Equipment Corporation came Data General and some 50 other minicomputer companies. These proud companies assaulted the mainframe and grew with dizzying speed; they were the Microsoft and Netscape of their day. Triumph seemed at hand, then came the workstation and the microcomputer. Disaster! Intel, Sun, and Silicon Graphics rose up. Data General almost died, to recently reemerge as an evangelist for non-proprietary "Open Systems".

Data General made a fortune, and it lost a fortune. Its zenith came on April 29th 1980, the day it introduced a 32 bit super-minicomputer, the Eclipse MV/8000. Eight years later the Eclipse line ended, along with all of Data General's proprietary systems. The Eclipse is now part of history, with no direct descendants.

Why should we be interested in a project that led to a now extinct product in a now little known computer company? There are several good reasons. Tracy Kidder wrote a Pulitzer prize winning novel about the Eclipse MV/8000 project (code-named Eagle), titled *The Soul of a New Machine* [1]. No other information system project has been so well documented from the human perspective. Secondly, the Eagle project did produce a new, high technology product, on a very tight timeline. Thirdly, we can compare the Eagle project to the yet more obscure Fountainhead Project, which took place simultaneously at Data General. Lastly, the project, and what followed, provides a window into a fascinating time in industrial history, and an important example of the strategic management of technology.

Methodology

Tracy Kidder's book provides a gold-mine of product related information, particularly when interpreted in the context of our course material. Unfortunately, not much else has been written about the Eagle project, which, after all, is only one of hundreds of similar projects that were underway at the time.

Traditional database searches produced little of value, as most databases do not extend beyond 1986. Locating relevant articles required manual review of available trade publications over the period 1979-1982. The annual Datamation "Top 100" allowed us to trace the history of the Eclipse Project, and the course of its product, over a nine year interval [ref 10-17, see also Attachment A — Timeline].

We used web search tools (Info Search, Alta Vista) to search web space for references to Eagle, Eclipse, and key project participants. That search guided us to Data General's web site. The last two years of position papers and annual reports allowed us to complete our timeline, analyze the longer term impact of the project, and identify some of the lessons learned during the Eclipse Project and thereafter [19-22].

We solicited input by posting on two newsgroups, the IEEE and ACM (Association of Computing Machinery) groups [Attachment D.1]. We received two very informative replies, one from a systems engineer who worked at DG at the same time (Brown), and another from an architecture specialist (Cline) who worked on related projects and knew the principles quite well [Attachment D.3]. The specific questions answered by Cline and Brown are found in Attachment D.2.

We also contacted Data General public relations by phone, but repeated voice messages went unanswered.

The varied data sources were used to construct the timelines [Attachment A], and data elements were aggregated by focal topic. There was surprisingly little overlap in information, and no real areas of conflict to resolve.

Project Description and History

For ten years Data General had been on the ascent. The company was born when Edward De Castro, one of DEC's preeminent engineers, walked out the door. De Castro started his new company, Data General, in a Westborough, Massachusetts' beauty parlor. For ten years it enjoyed annual growth rates of well over 20%, based on 16 bit minicomputers sold largely to a scientific and technical community that needed little hand-holding. Then, in 1978, the Data General engine began to sputter. Litigation and product delivery delays had alienated many customers. Worst of all, Digital Equipment Corporation (DEC), the leading minicomputer manufacturer, had created a leading edge minicomputer. The DEC VAX 11/780 was a 32-bit machine; it established a new category of high-end minicomputer — the "super-minicomputer". The super-minis appealed to technical users who needed power for simulations and graphical interfaces, and to business users considering alternatives to expensive mainframes for office automation. These profitable machines typically sold for about $100,000 to $300,000, but cost only $25,000 or so to make (1978 dollars).

Data General had no alternative to the VAX 11/780. They knew their technical customers were capable of changing vendors very quickly; Data General needed a promising solution quickly. Unfortunately, they were headed down the wrong track. The Fountainhead Project (FHP), begun secretly in late 1976, was nowhere near producing a product [4]. The majority of DG's R&D resources had been diverted to the FHP, located in North Carolina's Research Park Triangle. FHP was going to create the ultimate "no-compromises" minicomputer. This technically beautiful, elegant, longest-path project would burn scarce resources for almost 6 years, before being completely abandoned.

Back in Massachusetts, engineers were working on modifications to DG's aging 16-bit Eclipse line of minicomputers. They'd been left behind when the company's focus shifted to North Carolina, and they wanted a greater challenge to meet. It is not clear that they recognized that the FHP was doomed; Kidder's account suggests they were envious of its ambitious "clean page" approach. Cline, however, states that he declined an offer to join the FHP because "I didn't think it would work given the management structure" [4]. Thomas West, who is today the VP for Advanced Engineering at DG, wanted a project to tackle, and he wanted to occupy his engineers. He had the strong support of

Carl Carman, then DG's VP of engineering. Together they proposed a "fast, Eclipse-like machine", which West marketed as "insurance" [1, p. 47].

The project was accepted in April of 1978, but resources were scarce. DG was really not big enough to run two full-scale development projects, and the FHP had strong support from De Castro. Thomas West, who became Project Manager, and Carl Carman the VP for engineering, would have to fight for support. They made some Faustian bargains to get the project launched. Their first such bargain was to set a deadline of April 1979. The actual product, the Eclipse MV/8000, would be presented to the world on April 29, 1980, exactly one year late (100% over-run). The project was Christened "Eagle".

The project timeline is given in Attachments A and B (table form for this project and outline for overall DG timeline). Attachment C presents the Eclipse Group structure. An initial team began work in the spring of 1978. Several fundamental decisions were made very early in the project, by West with input from De Castro and team members. The new machine would be backwards compatible with 16-bit Eclipses and DG peripherals; it would run existing software without modification. The Eagle would not use a "bit mode"; it must not require restarting to run older applications. It must run faster than the equivalent DEC, the CPU should need far fewer than VAX's 27 boards, and each major element should fit on one board [1, p. 119]. Lastly, it would use "programmable array logic" (PAL) chips. The PAL chips greatly reduced the cost of production, and increased design flexibility, but they made manufacturing dependent upon a very new single-source technology from an outside company [1, 5].

Two important decisions were made largely because of resource constraints, but, in retrospect, they may have been the "right" decisions for the project. Firstly, West (PM), and Alsing (assistant PM) decided to hire "kids". New graduates would be much less expensive, would work longer hours, and would be more adaptable. Secondly, the new graduates would operate in a new work arrangement. In 1978 machine design was typically done by a group of architects, who worked separately from the implementation team. For the Eagle project, as at DEC, "implementers and architects [would] interact in a top-down, goal driven design effort at the earliest phase of development" [5, 4]. This method, with its improved interaction and communication and deferred design freeze, has become the industry standard.

The Eagle project was divided into a software implementation teams, lead by Alsing and Holland, and a hardware implementation team, led by Rasala and Holberger [Attachment C]. The teams called themselves the "Micro-Kids" and the "Hardy Boys". A veteran system architect, Steve Wallach would operate semi-independently, and also manage the critical liaison to the systems software groups (a separate functional group). The groups worked in close proximity in limited windowless quarters, closed off from the rest of the company. Team members described this as "mushroom management" (keep them in the dark and feed them ___). Intense collaboration and communication were almost inevitable.

The project was in full swing by the fall of 1978. The architectural design proceeded fairly smoothly, using work that had been done on MULTICS (UNIX precursor), and borrowing from two informal projects that had been terminated (EGO and EAGLE-1) [1, 4]. The machine architecture used "protection rings", which had been pioneered by

MULTICS but were not common in commercial machines [1, 4]. Kidder details several technical challenges over the next 6 months that had to be overcome, largely dealing with the coordination of cache architectures with memory sub-systems. These required intense work, but they were the expected challenges of developing a new system. At this time Data General as a whole was running into problems. DG was preoccupied with litigation, which would be a giant resource drain during the 80s. Customer support and management were having trouble dealing with the rapid growth of the 70s. [10]. Earnings and share prices began to fall.

In early 1979 a prototype system was operational, only a month or so behind schedule. About the same time West learned that the FHP would definitely not produce a machine within the year (it would never produce a production machine). Data General's finances worsened, and the struggle for resources became increasingly bitter. West felt that the future of DG depended on the Eagle.

Debugging began, and the project ran into serious problems. Debugging proved much more difficult than had been anticipated, even with the use of a sophisticated simulator which emulated the Eagle in software. The schedules continued to slip, and deadlines became mobile. In mid-1979 the sole supplier of the critical PAL components was teetering on bankruptcy — without these components the project was doomed. Around that time Carl Carmen, VP for engineering and a key project supporter, left Data General [10]. Data General's upper levels were at war, but West and project managers shielded team members.

In October of 1979 the prototype Eagles passed the Multiprogram Reliability Test, and went "down the hall" to software. It was a computer. A month later, West, the PM, left the project for a Data General position in Japan, allegedly because he'd made too many enemies [5]. Rasala, Wallach, and Alsing would steer the project to completion. The machine would be delayed another five months, apparently because the PAL components were in short supply. These were five months that DG could not spare, but there was little choice.

On April 29, 1980, the Eagle debuted at the Roosevelt Hotel in New York City. It was rechristened the Eclipse MV/8000. Articles in the Wall Street Journal and New York Times lauded the new machine, which had arrived just in time [6,7]. Team members came up from the basement, only to find their company in disarray. Their was little time or energy for celebration. In September of 1980, 2 years and five months after its formation, the remainder of the Eagle team disbanded. Alsing and Rasala went out to Silicon valley. Wallach, the architect, would co-found Convex supercomputers (now HP-Convex), where he is now VP for Advanced Technology. West returned from Japan, and continued to work at DG. He is now also VP for Advanced Technology.

The Eagle did what it had to do: "The Eagle project created a successful and highly profitable product line, that kept DG customers buying DG machines, but it did probably not increase their market share at any point -- and probably did not have that goal." [3]. The next 16 years would be a very rough ride for Data General. In 1981 DG's stock crashed from $87 to $20. The Eclipse MV/8000 produced 10% of revenue, and was their only area of improvement. Late in 1981 IBM released the first IBM PC, emphatically affirming the power of the micro-computer. A year later Sun began producing commodity UNIX workstations, running a commodity operating system.

DG crawled out of the 80s, alive only because of the Eclipse MV line, which sold very well in 1983 and 1984. Data General grew to 17,700 people on the strength of those years. Then came the crunch, as micro-computers ended DG's office automation strategy, and workstations took over the technical and engineering marketplace [9]. Government contracts, and business purchasers, began to demand non-proprietary systems [3].

DG did very badly in 1985-1988, and by 1988 it had ceased production of new minicomputers. Over the past eight years DG has shrunk to about 5000 people. It has become a producer of commodity servers (AViiON) and disk arrays (CLARiiON, and a source of open systems integration and services. Only in the past few months has Data General returned a profit [19, 21, 22]. Data General's mantra is now "commodity economics overwhelm proprietary economics". The "... challenge is to integrate the latest commodities into a product line ahead of competitors." [20]

Project Analysis

Project Classification: A Type D Project?

Shenhar has developed a 3-dimensional model for categorizing projects in terms of technology, scope, and pace [2, 23, 24]. The Eagle project can be analyzed along each dimension.

The pace of the Eagle project was very intense, involving many long days and nights for engineers. However, even if the importance of the project had been recognized, deadlines were not "do or die". They could, and did, slip frequently. Arguably they slipped *too* much: if Eagle had arrived six months earlier, Data General might have been far better off. Nonetheless, a "fast" paced project cannot afford even a month's slippage, and it diverts the attention and resources of the entire company. The Eagle project pace, in comparison, was only "regular".

Assessing the scope of the Eagle project is more challenging. It does not have "Array" scope, it lacks the interaction of large numbers of semi-autonomous units across time and space. However, a new computer system involves a formidable number of integration problems and complex interfaces: multiple hardware units, peripheral devices, and software at multiple levels — micro-code, operating system, compiler, application, and network. Even a relatively modest minicomputer project, such as later extensions to the Eclipse MV line, could never be managed along functional lines with a "family-like" atmosphere, as an "Assembly" scope project [24]. The Eagle super-minicomputer must be considered a "System" scope project. Even if its technologic challenges were relatively routine, it would require formal systems, extensive planning, and a matrix or project organization.

Shenhar has used the Eagle project as an example of type D, "super high-tech" project. He compares it to the Lockheed SR-71 or the Space Shuttle. In contrast, the building of the Macintosh is considered a type C, "high-tech" project. Is this type D classification justified? This is debatable. The SR-71 and Space Shuttle were far beyond anything that had been done before. The Eagle was a significant challenge, but DEC, Prime, and even

IBM had 32 bit minicomputers on the market when it came out. Shenhar describes the protection rings as a technological innovation, but others debate their novelty and point to the public MULTICS design [4]. Manufacturing constraints, such as the single-board CPU design, were important but incremental improvements also used at other companies. The combined architect-implementer teams may have been in use at DEC about the same time [4]. IBM had produced backwards compatibility, without a bit mode, in their 360 line. The PAL chips were new to the industry, but their main challenge was not technical, but rather marketing and manufacturing. From the beginning the Eagle was marketed internally as a natural, evolutionary, extension to the 16-bit Eclipse line. It would run the same operating system, and leverage DG's existing expertise.

Lastly, if we consider the Eagle as a type D project, then what would we call DG's (failed) FHP? Consider Cline's descriptions [4]:

> It [FHP] was wrong because it required too many breakthroughs at one time. Whole new compiler technology. Demand-paged uCode. Whole new OS. Early user-friendly (almost object oriented) command interface (actually a nice piece of work). Domain oriented system with some capability hacks. Tens of thousands of lines of microcode. Many TLB/cache like things to handle the many different flavors of pointers (the most general of which was something like 160 bits long). All this to be delivered in big-bang integration mode.
>
> ... FHP was the anti-RISC; the whole idea was that a higher level architecture would make things easier for compiler people. The MV/8000 had opposite goals; it was to be a Nova on steroids with few sops thrown to the compiler types.

On the other hand, the Macintosh, considered a type C project, involved a radically different software interface, a radically new operating system, and a design philosophy ten years ahead of the competition, all implemented on basic commodity hardware.

In the context of the competition, both external to Data General (DEC) and internal to Data General (FHP), we would not classify the Eagle as a ype D project.We consider it to have type C+: the integration of very advanced technology with many technical implementation challenges involving debugging and integration. It is noteworthy that the major project delays were due to manufacturing problems with the PAL chips, and debugging problems (weak debugging technology and resource limitations). It was the combination of high-technology, system-type scope, and a low-resource time-pressured development process that made the Eagle project particularly challenging.

Management And Leadership: Style And Attitude

Tracy Kidder gives us an extraordinary perspective on the management and human interactions during the Eclipse project. This perspective is well complemented by contributions from Cline and Brown [3,4]. De Castro set the flavor for Data General management, and West appears to have emulated De Castro, whom he regarded very highly. De Castro favored competition amongst his engineering teams. He fostered initiative, and provided direction while allowing creative people to run with their ideas. He told Wallach and West "no bit mode" — but he allowed the project to proceed, even though he seemed unenthused even at mid-project [1]. De Castro did not provide much positive feedback to employees, he favored a "macho" type of rough and ready culture.

In West's words: "No one ever pats anyone on the back around here. If De Castro patted me on the back, I'd probably quit." [1. p. 179]

West's management style was similar. He chose his people carefully, then gave them latitude to work. His deputies (Alsing and Rasala), in turn delegated to their deputies, who in turn allowed the "kids" to work things out on their own [1 p. 120]. Like De Castro, West did not give praise, though he would vigorously defend his team from any outside criticism. Kidder records several comments from junior team members who clearly missed hearing words of praise from their leader. West praised his team externally and to his peers, but he deliberately cut himself off from personal contact with team members. He seemed to fear the emotional burdens [1, p. 110].

West's "one-on-one" personal skills were not exceptional. Cline and Kidder document several instances of West's awkwardness and insensitivity [4, 1]. He tended to see the world in binary extremes; designs were right or wrong, engineers were winners or losers. He could write people off. He created enemies.

Leavitt divides leadership into 3 phases: Set Direction, Planning and Decision Making, and Implementing through People [23]. West did not excel at the third phase, but he could choose his deputies well. Alsing, Rasala, Holland and Holberger did better with communications, they had superior people skills. Carl Carman, VP for engineering, would greet novice engineers that West seemed to ignore.

West's real skills were in phases 1 and 2, *setting direction* and *planning and decision making*. These skills did extend to creating the "social architecture" for the team [25]. He created a space in which team members would operate (see below). He also had a gift for drama:

> "West made life in their corner of the basement more dramatic ... than it usually had to be" [1, p. 117].
>
> " ... West and other managers gave them enough freedom to invent, while at the same time guiding them towards success... He never passed up an opportunity to add flavor to the project. He helped to transform a dispute among engineers into a virtual War of the Roses. He created ... a seemingly endless series of "brushfires", and got his staff charged up about putting them out. He was always finding romance and excitement in the seemingly ordinary. He welcomed a journalist to observe his team; and how it did delight him when one of the so-called kids remarked to me "What we're doing must be important, if there's a writer covering it." [p. 275]

West's flair for the dramatic was an important aspect of his leadership and management style. It brought a special flavor to the project.

Organization

An organization chart for the Eagle project is presented in Attachment C. The principle form of organization was pure-project. Kidder suggests that some persons joined the team for a limited time, but many team members worked full-time, and over-time, on the project for the full two years. Most of the project members were hired specifically for this project, by the project manager (West) and his aide (Alsing).

Some external resources, such as manufacturing and marketing, were organized along functional lines. Most importantly, Software was a functional group who's resources were critical to project success. This appears to have been a source of significant friction, as Software was torn between serving the needs of the 16-bit Eclipse and the new 32-bit Eagle (Eclipse MV/8000). Wallach, who was a manager working under West, managed liaisons with the functional software group. It does not appear that software experts reported to West.

Resources

The Eagle project was provided with an adequate, if spartan, set of resources. Kidder ascribes this to a corporate culture in which project managers were expected to fight for project resources. "... De Castro liked to see a little competition ... you had to persuade such groups [Software] that your idea had merit and would get out the door ..." [1, p. 112].

West was a skilled fighter for project resources, with a strong sense of both strategies and tactics [1,3,4]. The main competition for resources came from North Carolina and the Fountainhead Project (FHP). DG had made a tremendous commitment to the FHP, and West recognized that a direct resource grab would fail. Instead he marketed the Eagle as a "fall back" or "insurance" strategy. Since FHP was accurately perceived as a high risk endeavor, he was able to persuade functional groups, like Software, to lend support.

Kidder suggests that West's strategy of preannouncing milestones, and of "cowboy scheduling" [4] had a tactical purpose. Arguably the Eclipse Group had to "show quick and constant progress in order to get the various arms of the company increasingly interested in helping out" [1, p. 118]. West fostered an atmosphere of excitement and drama, and marketed this internally to promote support and resources. The contention for resources could be bitter. Brown notes "in project management terms I'd have to say there was a lot of waste due to the internal warfare that Kidder doesn't begin to describe adequately" [3]. Following the Eclipse/Eagle project West temporarily moved to DG's Japanese Business Development; allegedly he was "forced to go because he had ruffled too many feathers" [4].

PM Tools Used

In the late 1970s project management tools were being used at Boeing, NASA, and some military contractors. They had not yet come into wide use in minicomputer and software companies. Cline and Brown report a mixed picture of strong documentation and weak project management and scheduling [2,4]. There were initially no tools for code management, but the project did adopt a tool developed by Cline (TCS) [3]. There were no automated CPM/PERT tools available during the project lifespan, though in the early 80s Data General adopted some early versions of project scheduling software. Kidder comments "The ... group ...seemed to be operating on instinct ... they kept no charts and graphs or organizational tables that meant anything ..." [1, p. 120].

Planning, Monitoring, Control

Kidder reports a series of schedule slippages, and of unrealistic timelines created for political purposes. The project's initial one year timeline appears to have been determined by the need to beat the rival Fountainhead Project's timeline: "... West said that Eagle would come to life in a year ... he felt he had to pursue "what's-the-earliest-date-by-which-you-can't-you-won't-be-finished" scheduling" ..." [1. p.113]. The project actually took 18 months, a slippage of about 50%.

Cline refers to the scheduling style as “cowboy scheduling” [4]. Alsing himself calls it crazy scheduling: “If you say you’re gonna do it in a year and you don’t take it seriously, then it’ll take three years. The game of crazy scheduling is in the category of games that you play on yourself, in order to get yourself to move” [1. p. 131].

The scheduling of milestones, for the DG Eclipse project, appears to have been a game which almost everyone played. It does not appear likely that any knowledgeable executive ever believed that the project would be done within a year. Carl Carlman, the VP for West’s division, would say “that no one upstairs believed they would finish Eagle that soon” [1. p.131]. On the other hand, the junior engineers appeared to take the deadlines quite seriously, and the project managers appeared to use them as motivational tools. When the FHP was known to be far behind schedule, West apparently used the April deadline to drive himself and his team [1, p. 134].

In contrast to the scheduling of major milestones, both Cline and Brown report that DG did an excellent job of documentation: “Requirements, Functional, and Design docs were under pretty good control and there was a good review process” [4], “written ... proposals and design documentation were as good as any I’ve seen in the industry” [3]. Formal reviews, typically associated with milestones, were common, and “usually involved manufacturing, documentation, and support staffers as well as engineering and marketing” [3].

West appears to have followed the course of the project through his managers. In addition to formal reports, he met every Friday afternoon for an informal project review with Wallach (architecture), Rasala (hardware) and Alsing (micro-code).

We know little of West’s formal planning. Kidder refers to West’s intense planning sessions, but provides no additional details. He provides examples of West being one step ahead of the project, so that at each milestone he’d already thought ahead to the next one. West does not appear to have anticipated the extensive problems that arose during debugging, however he did allow Alsing’s team to implement a system simulator. He’d declined simulator use in prior projects.

We have one example of managing uncertainty, albeit an unsuccessful example. West decided at the beginning of the project that the team would use the new programmable array logic chips (PAL), then available from only a single vendor. PAL circuits had a number of technical advantages. They greatly speeded development, allowing the hardware team to replace complex wiring with programmable chips. During debugging the PAL chips could be revised much more cheaply than a physical circuit. PAL chips lowered the cost of manufacturing. Wallach felt that they were an integral part of project success [5, 1 p. 188].

The Eclipse MV/6000 would be one of the first computers to use these chips, and West recognized the risk he was assuming. His worst fears came true in mid-project when the sole PAL vendor was close to bankruptcy. Release of the Eclipse would be delayed 6 months by PAL shortages. In this case the project risk factor was anticipated, but there were are not told of any resulting use of isolation or absorption strategies. West had left the project by the time the entire PAL issue was resolved.

One way to manage uncertainty is to defer decisions until appropriate information is available, and to produce consistent and reliable decisions. West appears to have done both. He had a good sense of when a project is “good enough” [1, p. 119-120], and could make quick judgments that a imperfect product was good enough to use.

Project termination does not appear to have been managed smoothly. West left the project months before the MV/8000 was released, and well before termination. Carl Carman, VP for engineering, left at the mid-point. Project members felt unappreciated.

Human Resources Management

Human resources management includes hiring, termination, and support. West was skilled at hiring and assembling a team, and he also relied upon trusted colleagues for the critical hiring process. Kidder records West's comments on the hiring process: “I hire Wests [hardware types], Alsing hires Alsings [software types]” [1, p. 156].

The hiring process for the project seems to have been quite successful. There was, however, substantial turnover during the project. In at least one case, a valuable team member left because of a personality conflict that might have been better managed. West and his managers felt that their engineers would be rewarded for their work with new projects: “another game of pinball”. The financial state of Data General meant these promises were not realized for many of the engineers.

Subsequently almost all of the project’s managers left Data General. Only West remained, albeit after a sojourn to Japan. Many of the other engineers formed new companies, or joined high-tech startups. Some commented that they felt “unappreciated” by Data General. They would have liked stock options, but mostly they wanted recognition for a hard job well done. Data General’s culture of macho independence was not oriented to positive feedback even at the best of times, and in late 1981 praise was hard to come by.

Teamwork And Simultaneous Leadership

West could not be considered a “people person”. Kidder’s account, however, demonstrates that he knew how to build a team. “He would bind his team with mutual trust, he had decided. When a person signed up to do a job for him, he would in turn trust that person to accomplish it ...” [1. p. 131] West knew that “mutual dependence required and recognized” was fundamental to team formation [26]. The process of team formation began with hiring, in which highly desirable applicants were told “we accept only the best ... we’ll get back to you” [1]. When the formal acceptance arrived the applicant knew they were joining a hot team.

Team building occurred in two stages. In April of 1978 West assembled his core managers. He chose them very carefully, recruiting intensively when his favored engineers were reluctant to join up. His persistence wore them down. Steve Wallach was the greatest challenge. A gifted system architect, he might be considered a “Prima Donna”, – a brilliant and talented person who’s value mandated unusual tolerance and special handling. Kidder describes him operating outside of the team organization, yet his contribution was fundamental. West’s recruitment of Wallach demonstrated patience, determination, and planning.

West sealed his team off from the chaos enveloping parts of Data General. “The managers had sealed off the team right form the start, ... the worst administrative problems never touched the MicroKids and Hardy Boys” [1, p. 226]. Down in the Westborough basement, team members would work with limited resources but no distractions. Formal and informal communication was facilitated between the Eagle sub-teams, and among managers. Communications between the engineers and West was filtered through the sub-team managers.

The Eagle managers created a culture for their team. It began with “signing up” — a ritual process of commitment [1, p. 63]. The senior team members were “the left-overs”, unwanted by North Carolina — but they would show they were superior engineers. They had to fight for resources, but they would get what they needed. The young engineers, with little prior work experience, readily joined this culture and adopted its private language [1. p. 46]. Rivalry with the North Carolina teams was a powerful glue, but it could get out of control. Brown comments that the “... warfare between development groups down there was extraordinary; I've never seen anything like it since.” [3].

West and his managers divided the development team into sub-teams, and encouraged friendly inter-team rivalry [Attachment C]. Alsing and others provided the social glue that West could not — outings, “award ceremonies”, personalized work areas, regular informal gatherings.

West ensured that the engineers would feel that this was “their machine”. He stayed in the background, making fundamental decisions and setting policy. He preferred his contributions to be “invisible” to team members: “There’s 30 guys out there who think it’s their machine ...” [1, p. 225]. Despite powerful temptations, he did not attempt to take over the technical responsibilities of team members. He respected team integrity. For team members to give their all, they had to feel ownership.

Thamhain lists six drivers that have positive associations with project team performance: professionally interesting work, recognition of accomplishment, experienced managers, proper technical leadership, qualified project team personnel, and professional growth potential [25]. The Eagle project had all of these, although recognition of accomplishment could have been better done. It also suffered from a few of the barriers to team formation that Thamhain mentions: power struggle and conflict and possibly insufficient resources. The positive factors were more numerous and stronger than the negative factors, and Kidder’s account shows that the Eagle team performed at a very high level. Team members were motivated by the desire to build a very fine computer — they wanted to win at “pinball”: “You win one game, you get to play another”. [p. 228]

Evaluation: Was the Eagle Project a Success?

Yes. Eagle did not revolutionize the world of computing. It didn’t guarantee Data General’s future. But in the words of Brown and Cline:

> The 32-bit MV Eagle was a natural extension of the Eclipse machines, and was designed primarily to make it possible to quickly reproduce the extensive software facilities of AOS/VS and its applications in a 32-bit address space. ... The project was essentially a shortest-path effort to keep up with a rapidly changing marketplace; the competing DG "FHP" megalith project in North Carolina was a technically beautiful but longest-path effort to achieve goals never met by the MULTICS projects of the 70s. The Eagle project created a successful and highly profitable product line, that kept DG customers buying DG machines, but it did probably not increase their market share at any point -- and probably did not have that goal. [3]

> The MV series kept DG alive for years, but much of that was in spite of the hardware and the architecture. Software did a lot to cover up for hardware deficiencies. [4]

The MV series was Data General's main revenue source from 1983 to 1990. It produced two excellent years, without which DG would almost certainly have disappeared. DG stopped making proprietary systems in 1988, but eight years later MV parts and services still produced 7% of operating revenue.

The Data General might have had a much bigger impact, but the Eagle was launched into a changed world. Workstations would seize the technical, scientific, and financial markets that DG has previously targeted. Micro-computers, with their less costly design and implementation methods, would handle the office automation market that DG wanted. On every platform commodity hardware would overrun proprietary systems. The Eagle was a solid product, but it was a dinosaur in the age of mammals.

How does this intuitive perspective mesh with the four dimensional model of project success [27]? Shenhar et al use four dimensions to analyze a project: meeting resource constraints, benefit to the consumer, business success, and preparing for the future. We consider each in turn.

The Eagle project ran 12 months beyond an impossibly short timeline. We do not know what formal budget was set for the project, but we are told that the Eagle project had to scrounge resources, particularly functional support (software), unused by the FHP. Considering the great technological uncertainty involved, the complexity of integration, and the struggle for resources, the Eagle project should be considered relatively successful in meeting resource constraints.

The MV line did meet customer expectations. It significantly improved on past performance, while preserving their investment in software and peripherals. The MV line did not greatly expand DG's base, but for eight years it appeared to satisfy existing customers in an extremely competitive market. We consider the MV line to have been a Type C technology project. Shenhar's criteria for success along the customer impact dimension for a Type C project is "significantly improved capabilities". The Eclipse meets that criteria. [27, see Table 5]

Was the Eclipse MV line a type type C "business success"? Did it produce high profits and market share? For two years it did: 1983 and 1984. Had that success continued, we would call the MV line great success. Alas, by 1985 the PC revolution was in full force. Even might DEC was reeling. Data General's office-oriented strategy was unraveling. On this dimension we would give the MV a mixed score.

Did the MV line prepare the future? In the computer industry seven years is a long time, and the MV was the centerpiece of DG's operations from 1982 to 1989. It produced a new product line, and for a short time opened new markets in offices and business. The MV and FHP experiences did provide longer term lessons for DG, but they were not happy lessons. Today the DG mantra is "commodity, not proprietary".

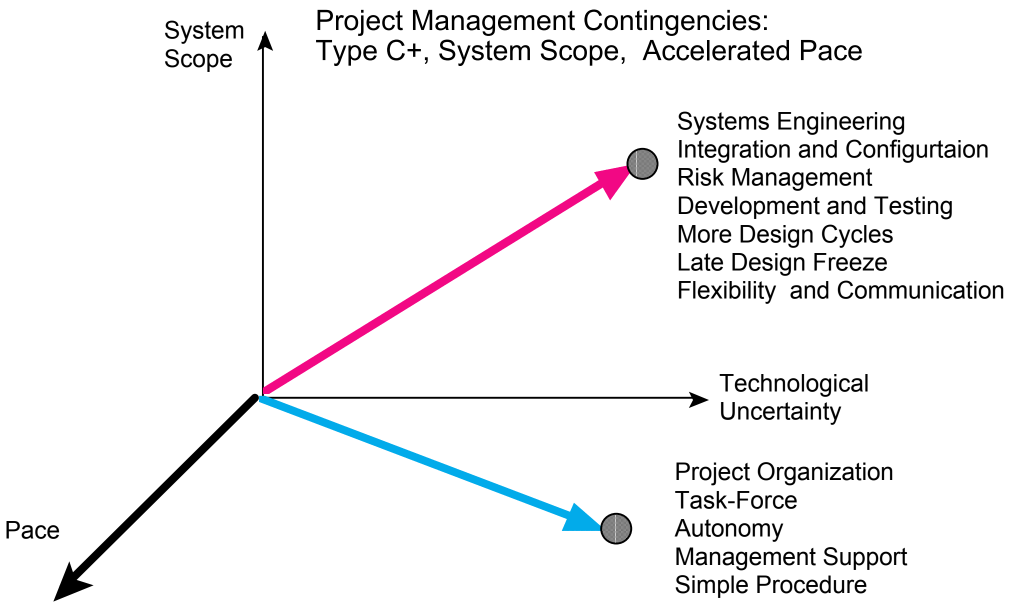
The four dimensional model supports our initial intuitive assessment. The Eclipse was a successful Type C High Tech project. It was not the great success it might have been in a world without PCs and workstations, but it kept the company alive.

Critical Evaluation and Recommendations

The Eagle Project, and its lesser known rival, the Fountainhead Project, were both very high technology projects. The first was moderately successful, the second an abysmal and expensive failure. What lessons can we learn from these Data General projects? How could they have been improved?

Our knowledge of these projects is limited. Little, if anything, was ever formally released about the FHP. Tracy Kidder's superb book documents the human aspects of the Eagle project, but it omits many aspects of project management and may be somewhat diplomatic. Nonetheless, based on private correspondence, literature reviews, Kidder's text, knowledge of the era, and some guesswork, we can venture some preliminary conclusions. We'll consider them in the context of Shenhar's model for Project Management Contingencies, given a type C+ project (high technology) of accelerated pace and System Scope. [Figure 1, 23]. Our representation of this model deals separately with the combination of scope and technology, and pace and technology. We'll discuss first recommendations pertaining to pace and technology, then consider the scope and technology combination.

Figure 1: Project Management Contingencies [23]



The Eclipse project implemented some of the *high-technology and accelerated pace* recommendations of Figure 1. It had a project organization, but the need to access critical software resources through functional lines was a significant weakness. The project had considerable management support, but it was lukewarm at times and probably disrupted by Carman's departure. West appeared to have considerable autonomy however — a positive aspect of the same lack of control which lead to disruptive infighting over limited resources. Procedures, insofar as we can tell, were simple and generally well executed.

Tactically, the Eagle Project had other strengths. Despite West's weaknesses in interpersonal communication, his social architecture work, and his managers, built a very strong team. Resources were limited, but, with the possible exception of the debugging phase, adequate. Formal and informal communications leveraged strong technical skills. Formal planning was not detailed, but considering the great uncertainties of a high tech project, and the "state of the art", the lack of detailed planning may have been appropriate for the project's level of certainty. Overruns of a year or more are still common on today's integrated hardware/software products. Project management, including termination, was disrupted by DG's internal struggles, but this occurred after most of the project work was done.

The Eagle Project also had some clear tactical weaknesses. The Data General culture of "no pats on the back" was unwise. There is abundant evidence to show that almost any human will work more effectively, and have better morale, with earned praise. Appropriate acknowledgements might have retained some very valuable technical staff following project completion. Kidder writes of strife between the development groups, and Brown describes the "warfare" as "extraordinary ... never seen anything like it ..." [3]. Some competition for resources may be valuable, but Data General overdid it. De Castro bears responsibility for the conflict over resources, and he was criticized for this management style in the early 80s.

The project ran into two major tactical problems causing long schedule delays. The PAL chips were unavailable, and debugging took much longer than anticipated. West clearly feared delays of the PAL chips, but was unable to manage the risks by either isolation or absorption. If DG had been more committed to the early project, perhaps they could have intervened early to stabilize the PAL source. Debugging is always a very uncertain activity, and West did authorize Alsing to create a "simulator" to aid debugging. It is not clear that additional resources would have helped, but the management turnover in the late project probably slowed debugging.

Perfect tactical execution might have brought the Eclipse MV/8000 to market 6 months earlier, and it might have retained more skilled engineers and improved morale. Those extra six months would have been very valuable, since the Eclipse market niche would effectively end in 1984. The increased revenue might have eased the inevitable transformation Data General experienced during the 80s.

Strategic problems, however, would have remained. The super-minicomputer market turned out to be quite limited. These strategic problems can be addressed through Scope-Technology contingencies (Figure 1). We are particularly concerned with strategic issues related to technologic discontinuities.

In the 1960s a technologic discontinuity, the replacement of the transistor by the integrated circuit, gave birth to Data General, DEC, and the other minicomputer companies. The entire basis for manufacturing and design changed, and new industries arose. Another discontinuity occurred during the Eagle project. Commodity processors began to replace custom, proprietary, integrated circuits in workstations and personal microcomputers [21]. That discontinuity almost finished Data General, and sharply limited the benefits of the Eagle project.

As shown in Figure 1, several strategies may be used to manage Scope-Technology contingencies, including technologic discontinuity. Sixteen years ago, the Eagle managers did use some of these techniques, such as flexible planning and extensive communications. For example, though some aspects of the Eagle were specified early, the close and early collaboration between the architects and implementers allowed a relatively late design freeze.

Systems engineering and risk management are newer components of high-tech project management. Risk management includes monitoring the technologic environment — looking for both quantitative changes and the less frequent technical discontinuity. The replacement of custom circuits by commodity microprocessors had very important economic implications; commodity microprocessors were so much cheaper than custom circuits that custom benefits couldn't compensate for the custom cost. The new technology had design and implementation implications as well ...

> "... computers had become so small and cheap, so suddenly, that most of the design and implementation methods that were appropriate for "mainframe" type computer systems became obsolete, because much more casual approaches to engineering and software were "acceptable" for single-user machines linked with LANs." [3]

If the impact of this new mode of production had been recognized, the rational response would have been to fully support the Eclipse effort and terminate the FHP. Knowing that the small office and technical super-microcomputer markets had only a limited lifespan, the FHP would be superfluous. Even the most extraordinary super-microcomputers would be unable to compete in these markets, hence all efforts would need to directed towards a transitional product that afforded backwards compatibility, such as the MV/8000. Technology development would then shift towards commodity products that would supersede the minicomputer line.

This is the strategy that Hewlett-Packard took. Of all the minicomputer manufacturers, only they emerged from the 80s relatively unscathed. DEC did as badly as Data General, only on a much larger scale. Instead, Data General did not terminate the FHP, but continued to pour money and talent into a dead-end. When they finally did follow HP's course, they'd been severely weakened.

Data General has learned from the last war. All of their quarterly reports and press releases restate the new party line: "commodity, not proprietary ... value added through integration and software ... customer support". [19-22]. Data General built commodity servers based on Motorola processors in the late 80s, but when Intel processors became cheaper commodities, they switched to Intel. They've built their servers using a customized version of UNIX (DG/UX), but recently they've moved to Windows NT.

Data General is on the lookout for another technologic discontinuity. They think they've found it in Intel's Pentium Pro commodity symmetric multiprocessor building blocks, machines built around processor blocks rather than chips [21, 22]. They aim to leap to the latest commodity architecture before their competitors. This time, DG doesn't want to be breeding dinosaurs.

Conclusion

The Eagle was an imperfect project in an imperfect company. Perfection is not common in the human world. Imperfect, it was still a marvelous project, the focus of a team of skilled engineers and managers for almost two years. They gave a part of themselves. Some were well rewarded, and others less recognized. Some were immortalized in a fine story, and others have their own memories. Many went on to other projects and founded companies, teaching others lessons they'd learned on the Eagle.

We've learned strategic lessons about managing uncertainty in a project, and uncertainty in the marketplace. Even with newer techniques, such as risk management and systems engineering, Eagle-class projects are risky. High tech companies have to steel themselves to dramatic changes in strategy. Microsoft's adaptation to the Internet is an outstanding example of a company shredding dozens of projects and years of work, changing direction so dramatically that its competitors were momentarily stunned.

Increasingly companies may have to adopt a Darwinian approach to risk management and high tech projects. They may launch many Eagle-class projects more or less simultaneously, and then be willing to terminate them in early stages as the market evolves. Commodity components, management structures that support project transitions, and systems for capturing and reusing knowledge, may be the foundations for high-tech project management management in the 21st century.

References

1. Kidder T. Soul of a New Machine. 1981. Little Brown and Company.
2. Shenhar AJ. From low to high-tech project management. R&D Mgt 1993 ;23(3):199-214.
3. Alex Brown, former Data General Systems Engineer. Personal communication.
4. David Cline, former Data General architecture developer (early Fountainhead, EGO, EAGLE-1). Personal communication.
5. Wallach S., Holland C. 32-bit Minicomputer Achieves Full 16-bit compatibility. Computer Design. 1981 Jan: p. 111-20.
6. Schuyten PJ. Data General Offers 'Super' Minicomputer. New York Times 1980 April 30.
7. Bulkeley WM. 'Super' Minincomputer Market Blossoms, Spurred by Growing Nonscientific Use. Wall Street Journal 1980 April 30.
8. Annual Microcomputer Survey. Datamation 1980 Nov:145-54.
9. Annual Microcomputer Survey. Datamation 1983 Nov:44.
10. The Top 100 Ranking: Company Profiles. Datamation. 1980 June: p. 108.
11. The Top 100 Ranking: Company Profiles. Datamation. 1981 June: p. 114.
12. The Top 100 Ranking: Company Profiles. Datamation. 1982 June: p. 136
13. The Top 100 Ranking: Company Profiles. Datamation. 1983 June: p. 116.
14. The Top 100 Ranking: Company Profiles. Datamation. 1984 June: p. 76.
15. The Top 100 Ranking: Company Profiles. Datamation. 1985 June: p. 72.
16. The Top 100 Ranking: Company Profiles. Datamation. 1986 June: p. 90.
17. The Top 100 Ranking: Company Profiles. Datamation. 1988 June: p. 90.
18. W. David Gardner. Route 128: Boston's Hot Bed of Technology. Datamation. 1981 Nov: p. 118.
19. Data General Corporation. 1994 Annual Report. http://www.dg.com/info/4581a.html.
20. Data General Corporation. Bringing Common Sense to Computing. http://www.dg.com/info/4581b.html.
21. Data General Corporation. 1995 Annual Report. http://www.dg.com/info/
22. Data General Corporation. Annual Meeting of Stockholders: January 31, 1996. http://www.dg.com/info/am96.html
23. Shenhar. Project Management. Class notes.
24. Shenhar AJ, Dvir D. Managing Technology Projects: A Contingent Exploratory Approach. In press. 1996.
25. Thamhain HJ. Team Building in Project Management. Course Packet.

26. Meredith JR, Mantel SJ. Project Management: A Managerial Approach. 3rd ed. John Wiley and Sons. 1995.

27. Shenhar AJ, Dvir D, Levy O. Project Success: A Multidimensional, Strategic Concept. In press. 1996.

Attachments

A. **Time Lines: The Eclipse Project**

B. Time Lines: Data General 1968 - 1996

1968-1978 (pre-project)

- **1968:** Data General founded. Built the NOVA, the first minicomputer based on integrated circuit technology.
- Successive generations of 16-bit minicomputers; known for high performance and excellent price/performance. Sales largely to technical and engineering customers.
- **1976:** Fountainhead Project begins with great secrecy in the Fountainhead Apartments.

1978

- *Data General begins its downward trend. Clear problems with customer support are emerging. DEC releases the VAX 11/780.*
- April: initial completion deadline set for April 1979.
- spring: preliminary working group, including West (PM), several experienced engineers.
- summer: hiring new graduates, assembling key team members
- late summer: preliminary design begins
- fall: "full" team assembled.

1979

- @Jan: initial design completed.
- Jan-Feb: begin debugging prototypes. News arrives that FHP will not be completed in time.
- March: debugging schedule slips to May.
- April: Instruction Processor identified as a main problem. Software brings in programmers for system software.
- June: Product Board Meeting and presentation to de Castro, who extends a "laconic blessing". Debugging schedule slips to September.
- June-July: Systems Software and Diagnostics. PAL supplier in financial trouble. National Computer Conference.
- @mid-year: Carl Carmen, VP Engineering and project supporter, leaves company.
- August: debugging continues. Prototype Whetstone performance lags behind VAX.
- September: Eagle failing Multiprogram Reliability Test.
- October 6: Debugging largely completed. "It's a computer". Data General releases a disappointing financial report. Data General's stock values will oscillate greatly over the next 18 months. The next 16 years will be quite rough!
- November: West leaves project, Rasala, Wallach and Alsing finish up. West stays with DG. He'll go to Japan for a few years, and eventually rise to be Senior VP for Advanced Engineering.
- Oct - Dec: refinements, diagnostic software
- *Earnings disappointing. Eagle project the subject of industry rumors.*

1980

- Jan - April: waiting for PALS supply, final debugging
- April 29, 1980: official debut at the Roosevelt Hotel in New York City. Eagle rechristened the Eclipse MV/8000. Eagle goes into production.
- spring and summer 1980: team members "surface" to find Data General in disarray. Severe infighting as organization contracts.

- Sept-Oct 1980: Eclipse group disbanded. Alsing and Rasala go west. Wallach will co-found Convex Supercomputers. All feel unappreciated by the company.
- *Data General is losing sales to DEC and Prime. Analysts feel the 32 bit systems are over a year late. Problems identified with sales and marketing. Large increase in corporate R&D (up 26% to 68 million) to develop software for Eclipse MV/8000.*

1981-1996 (post-project)

- **1981:** Eclipse MV/8000 supermini producing @10% of DG orders, only source of gain. DG stock crashes from $87 to $20. DG losing market share. Introduces Comprehensive Electronic Office software, will try to target office automation rather than research and engineering. Fountainhead Project is cancelled sometime around 1981-82. *First IBM PC released.*
- **1982:** *Sun microsystems established, will sell UNIX workstations to technical and engineering customers.* Another dismal year at DG, but *the Eclipse/MV line is expanding and new orders are coming in.* DG is catching up to DEC's technology. Former IBM execs join company.
- **1983:** *Datamation reports minicomputers losing out to microcomputers. Offfice automation goes to micros.* DG is back from the grave. Other former IBM execs join company to plot strategy. Strong updates to Eclipse/MV line, Eclipse accounts for more than 50% of revenues. DG produces a microcomputer system that runs their AOS operating system. De Castro hints DG will need to focus on services.
- **1984:** Superb year, with 40% leap in sales in fourth quarter. Eclipse line more than 50% of sales. Strong international growth. Data General moves away from technical and engineering market towards office automation. Micro line not selling. *Data General attains peak employment of 17,700 people, by 1995 DG will have 5000 employees and will be be continuing to reduce its fixed staffing. IBM AT released.*
- **1985-87:** 1985 is a disaster. Worsening sales, huge losses. Low-end super minis now sell for $40,000. Former IBM executives, who'd plotted strategies, leave. Comprehensive Office Automation software strategy has failed (lost to micros). Data General moves away from producing systems and towards delivering services.
- **1988:** Cease production of new minicomputers. Announced open systems strategy based on industry-standard microprocessors, operating systems, and storage components. Will target commercial market with UNIX systems. Commodity hardware becomes DG's byword.
- **1989:** Delivered first AViiON systems with Data General UNIX (DG/UX) using Motorola processor (competing with HP Unix systems). Ninety percent of revenues still coming from Eclipse MV line. *New products will be based on commodity hardware.*
- **1991-92:** Delivered first RAID storage systems, and introduced CLARiiON disk arrays as second generation of RAID systems for AViiON, ECLIPSE, and other computing platforms.
- **1992-1995:** Years of financial consolidation and downsizing. Company loses $187 million in 1993-95. By 1994 90% of revenues coming from commercial UNIX systems and drives. *Eclipse MV revenues are falling about 50% a year, but are still 7% of revenues in 1995.* May 1994, DG receives ISO 9001 certificate of compliance for US service and support operations.
- **1995-1996:** AViiON systems to use Intel processors and standard SMP (symmetric multiprocessing) motherboards. In early 1996 new systems run Windows NT server.

DG is still supporting about 19,000 Eclipse MV systems deployed worldwide, there are 30,000 AViiON servers worldwide. They are 3rd in market for medium-scale commercial UNIX servers. DG has had two consecutive quarters of profitability.

C. **Eclipse Group Project Structure**
A graphical version of this project structure follows. The number of staff varied, and included several not listed here. In mid-project it involved about 30 engineers.

1. Ed de Castro: CEO

2. Carl Carman: Vice-President

3. Project Manager: Thomas J. West (hiring)
   a) Secretarial Support: Rosemarie Searle
   b) Architecture and external Software Liaison: Steve Wallach
   c) Software Implementation: "Micro-kids" (Microcode and Simulator development)
      - Carl Alsing (team leader, "social director", hiring)
        - Chuck Holland (microcode manager)
          - Jon Blau
          - Dave Epstein
          - Dave Keating
          - Bob Beauchamp
        - Dave Peck (Simulator manager, other software)
          - Neal Firth

   d) Hardware Implementation: "Hardy Boys"
      - Ed Rasala (team leader)
        - Ken Holberger (supervisor)
          - Josh Rosen
          - Dave Epstein
          - Jim Guyer
          - Jim Veres

4. Support Groups (Functional)
   a) Software

D. **Electronic Solicitations and Responses**
(Following this page)

1. JF Newsgroup posting
2. Email questionnaire
3. Responses from Brown and Cline

E. **Sources**
(Following this page)
Sources are ordered by reference number and have the reference number written on the first page. These sources include relevant Data General web pages.