COVER SHEET

The Dag-Brücken ASRS Case Study

Tony Jewels
Centre for Information Technology Innovation
Queensland University of Technology
t.jewels@qut.edu.au

ABSTRACT

In 1996 an agreement was made between a well-known beverage manufacturer, Super-Cola Taiwan, (SCT) and a small Australian electrical engineering company, Dag-Brücken ASRS Pty Ltd, (DB), to provide an automated storage and retrieval system (ASRS) facility as part of SCT's production facilities in Asia. Recognising the potential of their innovative and technically advanced design, DB was awarded a State Premiers Export Award and was a finalist in that year's National Export Awards. The case tracks the development and subsequent implementation of the SCT ASRS project, setting out to highlight how the lack of appropriate IT development processes contributed to the ultimate failure of the project and the subsequent winding up of DB only one year after being honoured with these prestigious awards. The case provides compelling evidence of the types of project management incompetency that, from the literature, appears to contribute to the high failure rate in IT projects. For confidentiality reasons, the names of the principal parties are changed, but the case covers actual events documented by one of the project team members as part of his postgraduate studies, providing an example of the special mode of evidence collection that Yin (1994) calls 'participant-observation'.

Keywords: IT Project Management, Software Development, Organisational Design, ASRS, Materials Handling

1. INTRODUCTION

1.1 Scope of Case Study

For a comprehensive assessment of the DB failure it would be necessary to discuss all the business functionality of DB including marketing, administration (finance) and human resource management. It is intended to discuss these issues only where it can be seen to impact on information systems development. This case study will concentrate specifically on the IT component and is therefore not intended to provide definitive answers as to the reasons for the ultimate demise of DB. It concentrates on examining how the DB IT development processes, and the environment in which the processes were conducted, might have contributed to the ultimate failure of the project and to the eventual failure of the organization.

1.2 Logistics Overview

Australian warehouse storage and retrieval of product is still predominantly a manual or semi-automatic process employing a variety of materials handling equipment such as conveyors, elevators and fork lift trucks.

The relatively low cost of land in Australia, compared to high density population centres in Asia, generally limits that country's use of high rise storage facilities to special situations that might include hazardous storage conditions or where desired throughout cannot be maintained with a manual system.

DB believed that a market could be developed in Asia providing high rise automated warehouse solutions at more competitive prices than that demanded by the major suppliers, using a variation to the normally used ASRS configuration that involved automated ASRS robots (cranes) that were able to drive around corners.

Globally, most high rise warehouses use ASRS cranes that are only capable of travelling in a straight line (referred to as straight-aisle cranes). The limitation of a straight aisle crane is that one crane is required to service each storage aisle in a warehouse. As cranes are a major part of the cost of this type of warehouse solution, by reducing the numbers of cranes there are significant savings. It is theoretically possible for a single aisle-changing crane to service a whole multi-aisled warehouse.

1.3 DB ASRS Background

DB had a history of providing electrical control systems using programmable logic controllers (PLCs) for a number of process control applications. The entrepreneurial DB managing director (MD) was an electrical engineer who had developed knowledge of PLC software and believed that the development of PC control software to be used with this new type of ASRS held few technical challenges.

By 1996, the MD had secured a number of orders for their innovative and competitively priced aisle changing ASRS system. In addition to SCT in Taiwan the contracts included a milk supplier in Thailand, and a beverage manufacturer and two archival warehouse facilities in Singapore. Contracts had specified that the completion dates were to be within 12 months.

1.4 DB Project Scope
In order to successfully market the innovative DB aisle changing cranes it was believed necessary to offer a complete warehouse management solution to clients. The various components included all materials handling equipment, storage racking, crane control software and a PC based control system.

The SCT ASRS was designed to automatically move pallets of beverages from the end of multiple production lines for storage into a high rise warehouse; and when required for sale, to automatically move them to locations where they could be accessed for manual distribution. In what was a profound and accurate interpretation of the system it was philosophically described by one individual as *"Emptying full holes that need to be empty and filling empty holes that need to be full", (Kedge 1996)*

1.5 DB Project Rationale
For SCT a high-rise ASRS solution was selected because in order to minimise stock outs the amount of inventory required could not be stored in a traditional warehouse configuration with the amount of surface area available for that purpose. Although expensive off-site storage was being used as a temporary measure this had resulted in an unacceptably high level of 'out-of-date' stock by not adequately allowing a first in first out (FIFO) stock usage policy. With no personnel actually working inside the storage facility itself, the storage and retrieval system was designed to identify all pallets being stored, to optimise their storage locations and to retrieve the pallets in a FIFO pattern.

1.6 Equipment Used
The ASRS used two, twin fork, double sided, double reach cranes in a 5 aisle, 16 row warehouse configuration. The warehouse layout consisted of long aisles of up to 40 horizontal pallet locations and short aisles of up to 12 horizontal positions all of up to 10 pallets high (Figure 1).

With their multiple reach configuration each crane was capable in any long aisle, of reaching up to 8 pallet locations at any one static position i.e. each set of forks to the right, single deep and double deep, and to the left, single deep and double deep, (Figure 2). The vertical range of 10 pallets high effectively allowed any crane to access 80 pallet positions from any static horizontal location. The principal advantage of the DB system was of course that the cranes could travel around corners and could access any pallet position of any aisle.

1.7 IT Components
Because the system was automatic, all 'jobs' which, in their simplest form were merely movements from a specific pallet pick up point to a specific pallet delivery point, were required to be scheduled without manual intervention of any type. To operate fully 'automatically' it was necessary to embed a great deal of functionality within a number of the control software modules. At an early stage in the software design process the following functional modules were identified as being required for the system to run successfully:

Figure 1: ASRS Racking Layout

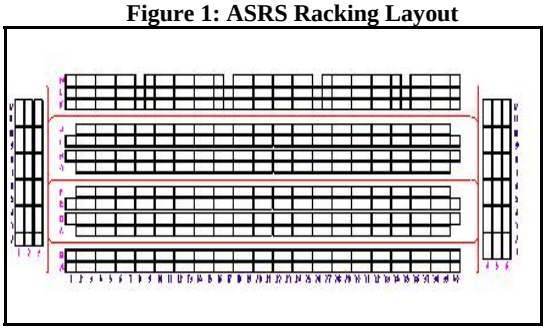
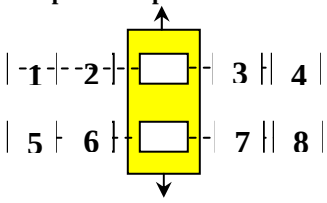


Figure 2: Multiple reach positions of each ASRS crane



- A graphical user interface (GUI) displaying the status of the cranes and storage status.
- A database to store information of storage and transactions.
- A "product mover" (PM) application to determine optimum storage and movements.
- A "traffic controller" (TC) application used to select routes, job priorities and crane separation.
- A communication package transferring data between PLC and PC.
- PLC programming of each crane.

Requirements for each functional module are covered in more detail in the IT specifications section.

1.8 Operational Requirements
The SCT ASRS was at the end of multiple production lines, that manufactured an assortment of beverages in P.E.T. bottles, tetrapak containers and aluminium cans and in-feed into the ASRS warehouse from the production lines took place on three levels.

Level 1 used a palletiser to feed an automatic guided vehicle (AGV), (referred to as the shuttle car), filling 9 double gravity roller locations, used exclusively for P.E.T. bottles. Level 2 used 2 third party supplied robots to palletise stock that were then fed by another AGV into 7 double conveyor infeed locations used exclusively for tetrapak containers. Level 3 used a single robot palletising for a single, double deep conveyor in-feed location used exclusively for canned products. Empty pallets were required at each palletiser for product to be packed onto. Outfeed production was dependent on external demands for stock and requirements for these movements more difficult to estimate. An hourly rate figure equivalent to the

number of pallets entering the ASRS was considered appropriate, i.e 50 in and 50 out. Outfeed locations were via 75 triple deep gravity roller locations situated in one of the long aisles. When any of the outfeed locations was emptied, sensors fitted to each crane were able to detect it and a job created to fill this *empty hole that needed to be filled.* The crane was required to perform an aisle scan every 30 minutes.

Normal production line speeds demanded that to avoid hold ups the ASRS had to be capable of consistently performing all the tasks detailed in Table 1.

Table 1: Performance Requirements for pallet movements

Job Type	Pallets per Hour	Pallet cycle time (minutes)
Clear Level 1 prodn.	30	2 min
Clear Level 2 prodn.	8	7.5 min
Clear Level 3 prodn	12	5 min
To Outfeed locations	50	1.2 min
Scanning Outfeed aisle	2	30 min
Empty pallet stacks (10)	5	12 min
TOTAL PALLET MOVEMENTS	**107**	**0.56 min**

2. THE PEOPLE

The MD and principal owner of DB was the entrepreneurial force behind the organisation and responsible for the basic design of the mechanical and electrical components as well as the software design overview and the overall project concept. He was described by one qualified graduate electrical engineer working on the project as a *'truly brilliant electrical engineering designer'*, although he was never able to confirm whether the MD had ever actually graduated from his university studies. The MD had excellent PLC programming skills, and had 'dabbled' with some Visual Basic code for which he had produced his own simple GUI presentation unlinked to a database which he had used as a sales aid.

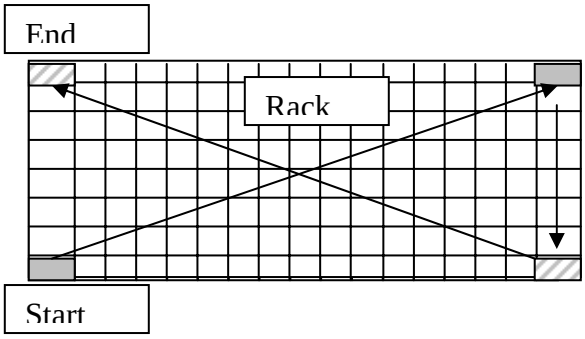
The General Manager (GM) with an extensive background in civil engineering project management had no IT project management experience. A further two project managers were employed for eventual on-site commissioning. One was a qualified mechanical engineer with extensive experience with mobile cranes who acted as the mechanical engineering manager in the production stage. The other had no formal management qualifications but was experienced in contract management and had an understanding of small projects. Somewhat surprisingly, nobody on the management team had any experience of the type of IT development that was being attempted.

There was at the commencement of the development a tactical decision taken to segregate the required software components into an IT based area, subsequently referred to as the 'PC' team and an electrical engineering based area that developed the 'PLC' software. The PC team was responsible for developing software components which included the GUI, database, communications package, traffic controller and product mover, and the PLC group provided code to operate the functions of the material handling equipment through electrical motors and sensors. To reinforce the distinctiveness of the two groups they were physically located in separate offices 40 metres apart, within the industrial complex that was being leased by DB.

Prior to appointing anyone in the PC team, a management decision had been taken to use a Windows NT based operating system with a Visual FoxPro development package for the GUI and database and C++ for integration with the PLC code via the 'traffic control' module.

The PC team was initially staffed with only two systems analyst/programmers (SA/P), who were appointed at the same time. Only one SA/P had any commercial development experience having successfully operated his privately owned IT inventory management business for 10 years, although his development experience had been in a UNIX environment with character based Foxbase+ application. Although his extensive background in inventory and warehouse management systems and formal qualifications in business management provided a business perspective for the project he had no experience of integrating technically advanced hardware communication features into software applications. His primary role was to design a GUI, PM and database that could be used with the DB ASRS system.

Figure 3: Crane Performance Contract Requirements



The other SA/P had recently completed an undergraduate software engineering degree and had experience in hardware communication systems with a telecommunications provider. His primary role was to design and write a communication system between the PLC and the PC and to design and write the TC module.

The PLC software design was to be carried out by the MD himself, two recently graduated electrical engineers and a more senior PLC electrical engineering programmer. These then, were the people that were initially involved in the

early stages of the actual development (as distinct from the pre-planning) stage.

3. THE CONTRACT

The contracts established between DB and its four ASRS clients prior to the commencement of development all contained provisions for staged payments throughout the project life cycle. Payments were to be made at confirmation of completion of milestones, with the actual payment amounts roughly equivalent to DB's proportional outlay for that stage. One of the stage payments for SCT was to be made at completion of the design and development of the control software that was to be used with the system.

Other than the occasional reference to a 'PC controlled GUI' and some arbitrary PC performance specifications there was no contractual obligation to design and develop the software in any particular way. The open nature of the PC software contract specifications reflected the fact that neither contract party clearly understood what a critical role the software component would ultimately play. Clients had entered into these contracts without involving their own IT/IS departments in any negotiation and it was not until after all contracts were signed that the SA/Ps first met with clients to produce the initial software specification. These meetings were also the first instances of client IT/IS personnel having direct contact with DB, although they had already been briefed by their own management.

The contracts contained required performance levels for the completed systems in terms of crane movements per hour and how these performance levels were to be measured, but the formula used was one used for measuring the cycle times of a straight aisle crane. Aisle changing performance, constituting the principal difference of the DB system was not included in the contract requirements. Though the system involved the use of two cranes designed to carry two pallets at once, contract specifications only included performance figures for the time to deliver a single pallet on a single crane to locations in the same aisle (Figure 3).

DB were offering a system that had yet to be completely designed, other than in broad conceptual terms, yet clients had entered into contracts that had guaranteed almost 80% of the total price before DB were required to deliver any working system. Stage payments were all at verifiable stages of the design/production process yet actual performance requirements were restricted to performances of the crane that could only be measured in the final stages of the project. Performance requirements for the 'working system' were based on single crane, single pallet, single aisle measurement formulae and did not take into account SCTs production line demands detailed in table 1. At an operational level there was evidence of 'implied' performance levels, with SCT erecting a prominent sign suggesting performance figures of 384 pallets/hour. The contract certainly did not offer this fictitious figure, yet this figure could have been calculated by extrapolating dual crane/fork figures from a single fork performance in a test environment. Yet, as poorly worded and constructed as it was, the contract still remained the principal source of reference for system design and performance matters.

Strict adherence to the letter of the contract by DB provided little real opportunities for clients to amend their contract terms and conditions, clearly biased towards the vendor, without accepting large price variations. Compared to other types of ASRS the original contract price was very competitive, yet any variation, (however small), from the terms of the original contract was either rejected outright, or was priced at an exorbitant rate by DB.

An example of a subsequent contract dispute related to the GUI produced by DB. Implicitly, SCT had assumed because the plant was in Taiwan, and the warehouse staff spoke and read Mandarin, that the GUI labels would be supplied in Mandarin. It was not until the system was being tested on-site that SCT realised that the GUI had been produced only in English, with the contract simply not covering this issue. Although the non-Mandarin speaking system designer had in fact developed a method of producing multiple language versions of the GUI this multiple language version was ultimately not adopted because no agreement could be reached on who was responsible for the additional costs involved in rectifying this contract oversight.

The absence of any operational performance requirements throughout the project life cycle resulted in not only the client being unable to evaluate progress but DB themselves having no real performance goals to work towards. Although a mature, experienced company may have been more successful in using a contract with this level of vagueness, by supplementing the contract terms with their own internal quality control procedures, DB had neither formal, nor informal quality control procedures in place. DB's processes were typical of the Software Engineering Institutes (SEI), capability maturity model for software (CMM), 'initial' or 'ad-hoc' stage of development, (Paulk, Curtis, Chrissis and Weber 1993). There may have been a number of reasons why DB did not have a formal QA policy, but high on that list might have been that their clients did not demand it. As Oskarsson and Glass (1996) suggest, "ISO 9000 is a tool for customers buying software more than for developers building it", (pxxi).

4. THE IS/IT COMPONENTS

Although DB was, as principal contractor, responsible for delivery of a complete working system, which included storage racking (supplied by a third party contractor) and the mechanical and electrical components of the system, this case study will confine itself to the various IS/IT components contained within the system.

4.1 IT Environment

The system operated through a Microsoft NT Server operating system using an IBM PC as the main server. Fitted to the cranes were PLCs that were controlled by

sensors and encoders, providing signals to electrical motors (eg hoist and horizontal travel motors). The PLCs also had two-way communication with the IBM Server, accepting commands from it (eg new job details), and providing confirmatory evidence of what it was actually doing (eg position of the crane or job completion). Communication between the static PC (through its COM1 and COM2 ports) and the mobile PLC was conducted using power modems that worked through the main power supply feeds to the cranes.

4.2 Software Modules:

A graphical user interface (GUI) - A screen generated representation of the physical layout of the warehouse in both plan and elevation views showed the real-time positions of the cranes and their statuses and the storage status of each pallet location. Created in an application development system (VB4), the GUI also provided two way communications with the cranes, display and resetting of error states and an emergency stop feature. Within the GUI there were also stock maintenance features, transaction reporting features and visual displays of the locations of product types. Its real-time storage location display was achieved by linking with an on-line database.

A product mover module (PM) - Embedded within the GUI were algorithms that created jobs for the cranes. In what was referred to as the 'job hunter' a continuous loop of database queries looked for *full pallet positions that needed to be emptied, and empty pallet positions that needed to be filled*. A job was initiated by the database query locating either a pick up point or a delivery point and the PM would, based on the attributes of either point, determine the other point to create a job. Various priorities were included within the algorithms that included:

- First in first out recall (FIFO)
- Last in first out recall (LIFO)
- Part Pallet priority
- Returned stock/mixed lot number priority
- Production lot number priority
- Fast moving stock priority and/or special storage
- Optimum storage selections eg front/back matching and height preferences.
- Quarantine status of stock

Communication code, between the PC and PLCs -

The communications module passed information between the traffic controller and the PLC on each crane but did not process any of the information. The TC and the communication module used a common DLL to pass information. The communication module would assemble the required messages for the PLC, which would be transmitted to the crane using a power line modem, connected to a single phase of the three-phase power to the crane as a dedicated communications line.

A traffic control module - The traffic control (TC) module was written as a dynamic link libraries (DLL) function in C++, and activated through the GUI code. Its principal purpose was to optimize the jobs that had been scheduled from the 'job-hunter', while selecting the most appropriate forks (of either crane) to transport the pallet and to select the most appropriate route for the crane to take. The choices that it needed to make in real time were:

- A priority sequence for jobs
- Which crane should be used for each job
- Which set of forks should be used for each job
- The most appropriate route to take from pallet pick up point to delivery point.

Crane use needed to be optimized by selecting jobs that would use both forks whenever possible, minimising situations where cranes were travelling with unallocated forks. An important additional function of the TC module was to retain separation of the cranes to avoid collisions. This function required the input of minimum safe distances between cranes varying according to which aisle the cranes were operating in.

Materials handling control system (PLC code) -The purpose of the PLC code was to accept a series of instructions to move pallets, and to report the crane's current status and alarms. The crane would accept a list of up to 10 movement instructions, a single instruction could be to pick up a pallet at a given coordinate, or to deliver a pallet at a given coordinate, or to move to a specified position. It was up to the crane to verify the coordinates, and to know how to move to them. If the destination was more than one aisle change away, the traffic controller had to include instructions for the intermediate path (so the path was not ambiguous).

The PLC code was written in a version of BASIC, with enhancements for running multiple threads. Each motor on the crane ran in a dedicated thread - the long travel motors, the hoist and the forks. Other functions on the crane also ran in their own thread, including the module for decoding the messages from the traffic controller, and encoder accumulation.

An On-line Database - A Microsoft Access relational database was used to store real-time storage location details and a transaction history of movements.

5. THE SOFTWARE DEVELOPMENT PROCESS

5.1 Designing in House

Although Thomsett (1989, p.24) suggests that "Project Planning is a team-driven process; all team members should be active in planning their project", certain strategic directions and tactical decisions had already been taken by DB management prior to employing any specialist IT personnel. The choice of a Windows NT environment on PC servers and workstations together with Visual C++ and Visual FoxPro application development systems had been already included in each client contract. The ability of IT developers to redefine the application environment and its component parts was therefore somewhat restricted.

5.2 GUI/Database

Client requirements for the GUI/database had been provided to both SA/Ps on a visit to Taiwan shortly after they had joined DB. These requirements together with DB management general overview briefings, was the full extent of the formal specifications for the design.

After approximately six weeks of initial design work using the Visual FoxPro application system, it was obvious to the SA/P working exclusively on it, that there were some serious technical limitations in using this combination of FoxPro combined database and front end with Visual C++. Although his background had been with FoxBase+/FoxPro and he had no professional experience with Visual Basic, the SA/P still advised management that they should reconsider the use of FoxPro in favour of VB with a separate database of either Microsoft Access or SQL. It was agreed to change the application development tool to VB and after further analysis, management decided to adopt MS Access as the database back end in preference to SQL.

Fortunately the original design of the GUI/database provided a high level of 'portability', described by Meyer (1997, p11) as, "… the ease of transferring software products to various hardware and software environments" and the change of application tool was carried out with relative ease. Although a formal inspection process did not exist, the SA/P demonstrated each milestone he reached to anyone in the organisation he thought might be interested. It was however becoming increasingly obvious to the SA/P that, without the communication module, his own work was essentially only of cosmetic value.

5.3 Communication Module

It was clearly evident after three months that development of the communication module and initial design of the traffic control module was proving to be more difficult than originally anticipated. The SA/P developing these modules had failed to deliver even an initial prototype for the communication module. As nobody at DB, (including his fellow SA/P), understood what he was doing the SA/P was unable to get internal help and unlike his fellow SA/P had not developed his own informal network of experts that he could call on for assistance. The SA/P being extremely introverted did not communicate readily with his colleagues, keeping to himself while working feverishly producing copious lines of code, the value of which only he understood.

Because there was no formal inspection process in place and no way of interpreting what had been achieved, this stand alone type development relied on an 'all or nothing' tracking approach, commonly referred to as the 'one hundred-zero' approach. Either a finished product emerges at the end or it doesn't. When ultimately a development ends up with 0% rather than 100% it is convenient to allocate blame on developers, yet a fundamental of modern quality assurance systems recognises managements ultimate responsibility for quality, (PMI 2000, p.97).

Unable to assist his fellow SA/P technically, the GUI SA/P was also unable to convince management that the total absence of any deliverable should be urgently addressed. His warnings were misconstrued by management as being simply a 'clash of personalities' between an extraverted developer and his introverted colleague, so management decided to bring in a third to oversee interaction between the SA/Ps.

5.4 The External Consultant

Management decided to engage the services of a part time external consultant to oversee the work being carried out by the two developers. When it was obvious to everyone that the development was not proceeding normally, the reality of software engineering as being more than just programming, (Ho-Stuart, Moraji and Thomas, 2000), was still not clear to DB management. This misunderstanding led DB management to select a university lecturer in electrical engineering with little commercial IT development experience, rather than an IT professional, to oversee the project.

His first task was to perform a walk-through of the two developers' code and his first report to management indicated that there was indeed a serious problem. The problem that he identified was however not in the communication module at all, but in the GUI/database module. Although also not understanding what the communication module code was attempting to do, his remarks on how well the communication module code looked e.g. indented correctly, well documented with proper naming conventions, contrasted with his scathing attack to DB management on how the coding appeared in the GUI/database module.

His advice was that the communication module was proceeding normally but that the GUI/database needed to be totally rewritten using proper naming conventions, code indenting and more documentation. He also became directly involved in redesigning the GUI, requiring features that had neither been discussed nor requested by clients, but which he felt would enhance the GUI's 'look and feel'.

The GUI SA/P complied with the consultant's instructions, as the explicitly stated alternative was his dismissal. However by focussing on what might be construed as 'cosmetic' items or internal quality factors, Meyer (1997), it deflected interest away from the practical consideration of non-delivery of any part of a communication module.

5.5 The Test Rig/PLC

In the early stages of the project DB had constructed at considerable cost a scaled down ASRS test site complete with racking and a single crane. The rationale for constructing this test facility was for both testing components and to show clients a prototype of the equipment that they had contracted for, because as yet, no such operational system existed.

It was used predominantly to test mechanical, electrical and PLC software features associated with the cranes. DB management appeared more at ease with the type of testing where something physical could be seen to be happening, e.g. cranes moved, forks extended and retracted, cradles lifted and importantly the crane drove around corners, rather other types of testing more commonly associated with software developments.

Because there was ever only one crane on the test rig the functionality of the TC module was limited to selections of route rather than its ability to cope with multiple cranes, maintaining separation or selecting the most appropriate crane for a particular job. Therefore most of the traffic control module features could never be physically tested on the test rig. Although this was a limitation of the test facility it did offer the opportunity to produce prototypes of all the other software modules.

A GUI was constructed representing the test rig racking layout, linking each storage location to a database in the same manner in which the operational version at SCT was designed to operate. Yet, the PLC testing was however almost exclusively conducted directly from the PLC code itself and not from the GUI via the communication module. Each module was essentially being produced separately by different groups of developers representing a development methodology that could best be described as a series of stand alone waterfall approaches, (Sommerville 1995).

This type of 'bottom-up' approach requires each of the components to be at an advanced level of completion before any serious attempt at integrating individual components can take place. If any of the development teams are unable to complete their individual components, use of such a monolithic approach, not only prevents integration testing from occurring, but those teams that have completed their own individual tasks are held up from further testing their own work. When for example the GUI/database (as a stand alone component) was in an advanced state of completion, the communication module had not progressed beyond an initial development stage. It was impossible therefore to pass messages from the GUI interface on the PC to the cranes PLC. The effectiveness of the test rig was therefore severely limited by the absence of one small but nevertheless critical component.

5.6 Integration

The initial decision to segregate the PC and PLC teams was an indication that DB management believed the functionality of each role was totally different, whereas in reality both groups were developing software components. The separation was overtly reinforced by segregating the teams in different offices and covertly reinforced by naming the PLC team as the 'A-Team'. Although this naming was done to motivate the team that the MD himself had decided to lead, it failed to acknowledge the demotivating effect on members who weren't part of his 'A-Team'.

It was eventually realised that all these individual parts were not automatically integrable and that in fact some major functionality was missing altogether. The absence of any integrated development plan, which might have resulted in an understanding of the need for a more appropriate software development methodology, such as evolutionary prototyping, limited the development options to more monolithic models such as the waterfall approach. Reliance on the completion of substantial portions of individual modules before any integration could begin, presented both significant technical and strategic risks, (Australian Computer Society Inc 2000). However, as no formal risk assessment was ever undertaken these risks were never explicitly acknowledged nor addressed by management.

Even when it was clear to DB management that the development was not proceeding normally, it still believed belief that software development was little more than programming, and that by increasing the amount of programming effort, solutions would eventuate. Its obvious next step was simply to employ additional programmers, which they did, in the form of more inexperienced graduate programmers. It is unclear, when subsequently faced with further evidence that the strategy of "throwing more programmers at the problem" was still not achieving results, why alternative development strategies were still not considered.

The ASRS building in Taiwan had been after some months delay, (not caused by DB), completed and the cranes themselves had been assembled within the storage racking. At this stage the client expected the software components to be added to the system so that operational testing of the cranes could begin. Although there was at yet no effective communication module, and only a rudimentary traffic control module it was believed that on-site commissioning must commence to comply with the terms of the contract.

5.7 Project management

The project manager chosen was the contract management specialist who had already been on-site for some months, supervising the construction of the building and erection of the cranes. In correspondence with the developers in Australia it was evident that he understood little of the nature and importance of the software components and he was unaware just how little of the software had actually been completed. His project plan indicated that using two PLC developers and one SA/P, commissioning (commencing in August 1997) would be completed in three weeks. However, after five weeks of frantic development only a simple screen display of the position of one of the cranes was achieved, indicating that a basic communication link had finally been established. At this point there was no operational TC module, which meant that multiple cranes could not safely operate together. On site development work in Taiwan was in fact to continue

with a team of up to six people up to the official plant opening on April 1st 1998 and beyond.

Although it appeared superficially that there was a separation of responsibility between managerial and technical control in the project, (Thomsett 1989), the reality was that nobody was adequately performing either function in the development process.

- Managerial, as the organisation had undergone a rapid expansion process with little infrastructure to support the expansion. The absence of a detailed risk assessment together with unrealistic scheduling, and a failure to prioritise indicated a deficiency in management control of the development process.
- Technically, as the projects were prototypes that demanded a high degree of research and development style work. The absence of functional specifications, data models, design charts, module specifications and test plans indicated a deficiency in the technical control of the development process.

5.8 Testing

On-site commissioning was supposed to represent only the installation, testing, tuning and user training phases appropriate to the software development yet the commissioning phase became merely an extension of the research and development that was still being undertaken in Australia.

While one SA/P remained in Australia working on the communication and TC modules the other SA/P was on-site, installing and testing those parts of the GUI that did not require any communication link and populating the database with the warehouses specific storage data. The PLC engineers could not properly test their code as the crane's electrical hardware had yet to be completed.

No formal test plans were designed throughout the development project although individual developers, when time permitted, did attempt to test their own work. There was little static testing by any third party source and hence reviews, walk-throughs and inspections were either self generated by the developers themselves or not carried out. The academic appointed to oversee the software development part of the project undertook some initial static testing but appeared to make all seven of the common problems associated with software reviews that Weigers (1998) refers to:

1. Participants don't understand the review process.
2. Reviewers critique the producer, not the product.
3. Reviews are not planned.
4. Review meetings drift into problem solving.
5. Reviewers aren't prepared.
6. The wrong people participate.
7. Reviewers focus on style, not substance.

The level of test plan employed by DB could be best indicated by relating the example of the original on-site project manager being requested by the MD one evening to construct a test plan for the engineers to carry out. Awaiting the engineers the following morning was their test plan, a hand written document that said, in bold letters… "**TEST CRANE 1** then **TEST CRANE 2**." No further test plans were ever requested from that project manager.

The nature of the development process prevented most integration testing until the latter stages of the project. Even when DB's testing uncovered considerable faults and in some cases completely missing functionality, little attempt was made to modify the development plan and little immediate formal action taken to correct software related deficiencies. It was generally left up to the developers themselves to modify their code and to add additional features when necessary. This led to situations where on-site developers were working up to 108 hours per week and regular 16 hours per day for weeks at a time.
When one crane was operational and was being controlled by the GUI it was possible to finally perform a simple stopwatch test in the manner that was detailed in the contract terms. It appeared that the crane could meet the contract specifications for required straight aisle performance. However, the opportunity was taken to also test the performance of the crane in a more realistic operational test, picking up double pallets from one aisle and delivering them both to another aisle. The required pallet movements (table 1) for each fork was therefore (2 x 2 x 33.6 seconds) or approximately 2 mins 14 seconds per crane with 100% efficiency of two crane utilisation. The average recorded time for a single crane pick up and delivery of two pallets was 2 minutes and 12 seconds. This indicated that only with close to 100% crane utilisation efficiency, would the cranes be capable of reaching SCT's operational requirements.

Because on site developers were working such extraordinary long hours there was little time left for static product analysis involving the regular analysis of source code that they had written. The GUI had included 38 different PLC crane errors such as:

- Front Fork Encoder
- Long Travel Vector Drive
- Failed Delivery Location Full
- General PLC Run

When the cranes were operating, capturing and recording these crane errors provided an opportunity to undertake dynamic product analysis by automatically collecting information about the systems own performance, with no additional user effort. By plotting the daily log of summaries of errors it provided DB with a guide to how long the system would take to stabilise while also identifying problems as they occurred and provided a method of prioritising remedial action.

This dynamic testing had been undertaken by the on-site SA/P for his own analysis purposes. When the client became aware of this method of testing they too were keenly interested in the results and what they represented.

An example of the, by that time, strained relationship between the management of DB and SCT and the degree of trust exhibited by the parties, manifested itself in DB's refusal to provide this data to the client without a formally documented request, which ultimately took another three weeks.

When one crane went into any error state both cranes immediately stopped operating, the rationale being that if one crane was for some reason unable to operate, the other crane might not know where it was and a collision might occur. The GUI would display what fault had occurred and it was then necessary to reset all errors from the GUI before either crane would restart. Some (critical) faults also needed to be investigated and rectified by physical entry into the warehouse. The average time for resetting faults was 3 minutes.

Over a one month period of pre-production testing the level of overall critical and non-critical errors did not significantly drop, with the total amount of errors averaging over 200 per day, roughly equivalent to 6 hours of 'downtime' per production day. An algorithmic model predicting when the system would stabilise was unable to compute a date. DB's immediate reaction was to stop producing the daily report.

5.9 Development Performance

The relative simplicity of the GUI/database components and the ease at which the product mover had been integrated into the code might have confirmed DB's misconception of the level of complexity of the other software components

The language used in the PLC coding was itself not that sophisticated or difficult to work with. The individual functions likewise were not that difficult, but the combination of all of the individual components made this software component very awkward to develop. The lack of overall functional specifications, design documents, test plans or formal documentation forced developers to adopt an ad-hoc approach in which there was a reliance on reactive rather than pro-active responses for fixing problems, as they occurred. The combination of the use of the selected types of PLC and motors presented problems that were special to this environment and made the nature of the development more R & D rather than process oriented. The graduate developers working in this area had never worked with this simple type of software, having studied more sophisticated application systems such as C++ and Java at university.

The communication module was the key component in integrating the most basic commands between PC and PLC. Throughout the project the communication module, as a result of the design of the physical communication layout, continued to produce 'noise' which sometimes resulted in phantom calls to and from the PLC.

Although the communication module could be considered a key component for basic operations, the TC module was critical for performance purposes. Development of the TC was key to the success of this type of ASRS where multiple cranes can operate in the same physical space. None of the developers ever admitted the complexity of the tasks that they were being asked to perform or their inability to deliver what was required. Subsequent investigation of the complexity involved in the requirements for the TC module was undertaken by separately asking the opinion of two senior software engineers, both with PhDs in Computer Science and one with a specialisation in robotics. Although both believed it 'might be possible, given enough time' they agreed that a fully optimised version would almost certainly be outside the capabilities of undergraduates or graduates with limited experience.

The versions that were being provided for use at SCT were however far from being optimised. Early versions of the DLL were considered successful if they didn't crash the system yet permitted both cranes to operate concurrently. But at the time of the official plant opening the TC version in use was such that a single crane operating alone could deliver more jobs than both cranes operating together. Attempting to use both cranes in the same general area would result in both cranes spending more time getting out of each others way rather than picking up and delivering pallets. Operators realised this and would send one crane to a remote part of the warehouse, disable it and allow the remaining crane to work on its own.

5.10 Negotiation

The obvious vagueness in the contract terms created a plethora of potential conflicts needing resolution throughout the project yet the absence of formal processes for enabling specification changes and handling conflict resolutions made the tasks of individuals working on site arduous to say the least.

The generally held Western belief that a contract was legally binding contrasted sharply with the more usual Asian interpretation that contracts were more of a 'starting point for negotiation'. The intransigence of DB on some issues thus appeared to SCT simply as uncooperative and recalcitrant behaviour.

This interpretation was reinforced by the original project manager adopting, what was described by client representatives and sub contractors alike as "a dictatorial attitude exhibiting an abrupt and uncompromising communication and negotiating style". His relationships in the various construction phases had predominantly been with local manual labourers and tradesmen; the autocratic style he had adopted he believed to be most appropriate for that situation. Yet when he was joined on-site by professionally qualified, self motivated and technically aware DB staff he failed to modify his leadership style. His almost total lack of understanding of the technical aspects of the IT development, ignorance of IT project

management methodologies coupled with an already fragile client relationship, limited his contribution to the post construction phase of the project. It was not altogether surprising when another project manager was appointed to complete the more technical final phase of the project.

5.11 Leadership

Understanding both his weakness as a team member/leader and his poor overall management skills, the MD had appointed a general manager who had both excellent project management skills and general management abilities, to take a hands-on role managing the business. Yet, even though the MD was not even physically located in the same complex his influence over the rest of the organisation was still pervasive. The MD was an 'ideas man' who could develop concepts in his own mind but had difficulty in passing these ideas on to others. He had little need himself, for formal processes or methodologies as his own work was both innovative and multifarious. The MD also appeared to have had difficulty in coming to terms with the concept of 'separation of responsibility' and his role as 'business owner' often clashed with his role as 'technical director'. Whereas it is of course possible to have the same individual perform more than one role this duality was misunderstood by many staff, suppliers and clients alike and importantly misinterpreted by the MD himself.

5.12 Organizational Structure & Design

The layout of the work environment at DB's head office was itself worthy of mention. The workshop area was separated into offices housing teams working on their various functional tasks. Informal discussions, in work breaks, among employees working in different functional areas, effectively ceased when the groups were required to take separate breaks.

The appointment of the part time academic in a contract role initiated a DB policy change away from employing most staff on a full time basis. The hiring of part time contractors, working with full time employees but earning significantly higher hourly rates was undertaken principally because these contractors had other commitments that prevented them from working full time. Any negative effect from the disproportionate remunerations from this arrangement did not become apparent until the part time contractors were working and claiming the equivalent of full time hours, and subsequently greater than normal full time hours. Although full time staff were unable to claim overtime for any work performed in excess of their 'normal' 40 hour week, on-site they frequently worked well in excess of these hours. Their motivation for working such extended hours was obviously not financial, yet the presence of staff claiming large overtime payments was particularly frustrating to them. The development team(s) that had never worked together as a unified and synergistic unit had yet another reason to work heterogeneously.

5.13 The Contractors

The motivations of the 'contractors'; as they were then described by the full time employees, was being questioned. It is still difficult to determine why full time staff under such extraordinary pressure, retained their motivation, perseverance and dedication to their tasks, yet even though everyone was failing, the contractor's inability to succeed was given special derision. In what appeared to be a case of a self-fulfilling prophecy, (Rosenthal and Jacobson 1968), the perceived 'lack of loyalty' by the contractors ultimately manifested in a covert act that was to have severe implications for DB. Two DB contractors working together on another DB contract in Singapore were approached by that client to 'consult' for a day, prior to returning to Australia. The contractors, as 'independent' consultants, and paid handsomely by the client for only an extra days work, provided the client with inside information about the project that possibly contributed to that client terminating their contract with DB a few days later. This clandestine meeting in early 1998 and the precise nature of the details in it was never made known to DB management, but the contract cancellation impacted on DB's overall cash flow, credibility and possibly stakeholder confidence in successful completion of its other projects.

5.14 The End is Nigh (Epilogue)

Less than one week after DB had been awarded its export award it announced that 'for administrative reasons' it was to be placed in the hands of administrators. Approximately 12 months after this took effect the company was wound up and its assets liquidated. The effect of this liquidation resulted in DB being unable to provide further assistance to SCT, as the result of which:

- The SCT-ASRS was unable to perform at anywhere near the required operational speed.
- The system was unreliable both mechanically and electrically.
- The software components still required substantial modification, debugging and tuning and did not have the expected functionality.

SCT eventually replaced the whole system with a more conventional straight aisle ASRS model. Although many ASRS systems have been successfully completed since, none have ever used the same technology as attempted by DB. As well as all DB employees losing their jobs, a number of client executives that were involved with the various ASRS projects were also dismissed. The administrators were ultimately only able to award DB creditors a few cents in the dollar, and legal action was undertaken against the administrators themselves.

6. CONCLUSIONS

The case study failure has hopefully provided some important lessons, and it is hoped that its messages might help prevent similar occurrences in other projects.

This case study attempts to illustrate why modern IT project managers require a range of multi-disciplinary skill-sets in order to increase the likelihood of project success. It also illustrates how the absence of any one of those skill-sets might contribute towards project failure. Generic project management skills, acquired within one discipline, regardless of the general competence of the individual, may be neither sufficient nor appropriate for application within an alternative discipline.

ACRONYMS

ADS	Application Development System
AGV	Automatic Guided Vehicle
ASRS	Automated Storage and Retrieval System
CMM	Capability Maturity Model
DB	Dag Brücken Pty Ltd
DLL	Dynamic Link Libraries
GM	General Manager
GUI	Graphical User Interface
MD	Managing Director
MIS	Management Information Systems
PIR	Post Implementation Review
PLC	Programmable Logic Controller
PM	Product Mover
SA/P	Systems Analyst/Programmer
SCT	Super Cola Taiwan
TC	Traffic Controller
VB	Visual Basic

7. ACKNOWLEDGEMENTS

Advice, information and support in preparing this case, is gratefully acknowledged. Particular thanks go to Richard Pope, Moira Wilson, Horng-Ming Su, Luke Chieu and those individuals who wished to remain anonymous.

8. REFERENCES

Australian Computer Society Inc (2000) *Project Management 2 Study Guide Version 2*, ACS Professional Development Distance Education Programs, Melbourne, Australia.

Gido, J. and Clements, J. P. (2003) *Successful Project Management 2nd edition*, Thomson Learning, USA.

Ho-Stuart, C., Moraji, H. and Thomas, R. (2000) *Software Engineering and Quality Assurance Work Book*, Faculty of Information Technology, Queensland University of Technology, Brisbane.

Kedge, P. (1996) A casual conversation over lunch with Dag-Brücken Systems Analyst, Taoyuan City, Republic of China, 8th August

Kerzner, H. (1987) In Search of Excellence in Project Management, *Journal of Systems Management*, (February), pp.30-39.

Meyer, B. (1997) *Object-Oriented Software Construction, 2nd edition*, Prentice-Hall PTR, Upper Saddle River, New Jersey.

Oskarsson, O. and Glass, R. L. (1996) *An ISO 9000 Approach to Building Quality Software*, Prentice Hall, New Jersey.

Paulk, M., Curtis, B., Chrissis, M. and Weber, C. (1993) *The Capability Maturity Model for Software version 1.1*, Technical Report Software Engineering Institute, Pittsburgh, 26 pages

PMI (2000) *A Guide to the Project Management Body of Knowledge*, Project Management Institute Inc, White Plains.

Rosenthal, R. and Jacobson, L. (1968) *Pygmalion in the Classroom: Teacher Expectation and Pupils' Intellectual Development*, Holt, Rinehart & Winston, New York NY.

Schwalbe, K. (2002) *Information Technology Project Management 2nd edition*, Course Technology, Canada.

Sommerville, I. (1995) *Software Engineering 5th edition*, Addison-Wesley, USA.

Thomsett, R. (1989) *Third Wave Project Management*, Prentice Hall, Upper Saddle River, NJ.

Weigers, K. (1998) The Seven Deadly Sins of Software Reviews in *Software Development Magazine* August pp

Yin, R. K. (1994) *Case Study Research: Design and Methods 2nd ed.*, Sage Publications, USA.

AUTHOR BIOGRAPHY

Tony Jewels has been an IT practitioner for over 30 years, in an extensive number and types of roles including programmer, systems analyst, hardware engineer, system architect and IT Project Manager operating in Europe, Australia and Asia. He holds an undergraduate degree in business management, a Masters degree in IT and is currently completing his PhD with a topic of increasing IT project success applying knowledge management principles. Currently he lectures and coordinates the IT project management units at graduate and postgraduate levels at the Queensland University of Technology, Australia.