Wile E. Coyote
BSc (Hons)

Matric No: 2005xxxxx

Honours Project Final Report

Project Title: Evaluation of Tools to Support Large Scale Batch Scheduling
Project Supervisor: Richard Foley

Second Marker: Eddie Gray

"Except where explicitly stated all work in this document is my own"

Signed: _____ **Date:**

Abstract:

Batch scheduling is a method of automating many previous manual tasks in organisations via introducing software to help manage these underlying and vital processes. The tool has evolved from a traditional mainframe technology into a critical tool which often runs many businesses, running multi process and cross platform systems globally. Batch scheduling has become a tool of choice for many organisations looking to expand and depend on IT improving the workings of these day to day tasks and processes.

These tools are now becoming essential especially within the financial services sector with its size and importance growing both nationally and globally. Research shows that batch scheduling is being sought more and more to improve the computing processing within many organisations, particularly within the financial services. Helping to automate and streamline many previously manual dated processes. However, with a choice of over 50 vendors delivering batch scheduling tools it can be difficult to know the tool of choice. It therefore requires a method of analysing and assessing the tools to identify those effective solutions which can improve the automation process within a large organisation looking towards batch scheduling as a solution.

This research takes the form of a case study of a large global financial organisation aiming to evaluate 4 batch scheduling solutions (Control-M, AutoSys, Dollar Universe and CRON) to identify the most appropriate solution. This research outlines a number of methods and processes which must be followed through to obtain a reasonable outcome.

In order to analyse these tools a number of generic requirements and characteristics have been identified for which a good batch scheduling tool should adhere too. By analysing these generic requirements key categories and sub requirements specific to each section have been identified which allow for a requirements matching exercise to be undertaken upon batch scheduling software. Further to this many software evaluations take into account various other aspects, mainly that of user interface, thus a secondary evaluation method to assess the HCI of batch scheduling tools has bee created. This takes the form of an HCI QUIS to obtain expert user opinions from within company X.

This research outcomes show there are 3 categories of batch scheduling solutions that of native, basic and advanced. The techniques developed allow for the tools to be categorised using these methods and the results also show the gap between each category of tool. Alongside this he relationship with the user interface of these tools and the satisfaction of the user groups is also highlighted where by those tools which are more advanced often satisfy the user best due to their highly interactive GUI systems. It also highlights the similarities between the advanced batch scheduling tools of the 4 used within company X and gives company X an outcome which leaves 2 possible options for the tool of choice. These being Control-M or AutoSys, while at the same time highlighting the lack of ability in CRON and Dollar Universe as batch scheduling solutions to a large organisation.

The study therefore identifies how to perform an in depth batch scheduling software evaluation which can compare tools for the conclusion of “which is best” according to the developed techniques and requirements list. In conclusion to this evaluation the tools ability and performance can be assessed and compared with that of similar tools.

Contents

Acknowledgments

I would like to thank some important people who have helped me throughout this final year project. Firstly I would like to that my supervisor Dr. Richard Foley for all his guidance, support and encouragement throughout the year; it has been appreciated greatly throughout. I would also like to thank all my family and friends for their help and support they offered over the course of the project. This final project could not have been completed without this help.

1 Introduction

1.1 Background

Batch scheduling has evolved with computing over the years and is now an integral but often forgotten area of IT. Batch scheduling has always been in existence from manual paper based systems, requiring user input throughout each stage of a batch, all the way to automated tools which can run multiple jobs across a network of systems and platforms globally. It is these tools which have become an integral part of most large organisations and the choice of which tool to use can often be a sticking point. This evolution has ultimately changed the way an organisation can operate and with that comes benefits.

The term batch scheduling relates to the process of automating tasks, jobs or commands which traditionally a user would perform on a networked machine. These jobs are usually linked and dependant on the completion of other tasks or processes and so often it became a complex system. Nowadays this so complex it is a global operation which many organisations look towards tools to help manage the tasks. Reports show (Ferrarini, E. 2003, and Gartner, 2007) that batch scheduling is being sought more and more to improve the computing processing within many organisations, particularly within the financial services. Helping to automate and streamline many previously manual dated processes.

In Elizabeth Ferrarini's article (Ferrarini, E. 2003.) she indicates that since 2001 all industries have been looking increasingly towards IT to improve the efficiency and throughput of the business and that "Nowhere is this more apparent in financial services where batch job scheduling has become the critical component to IT success"

The need specifically within the financial services for this batch scheduling is mainly due to the boom in the economy and this sector within the last decade. As we are now seeing to date with the financial "crisis" or "Credit Crunch" (BBC News Website, 2008) we are faced with, the financial services are perhaps the most important sector for many countries economies. Many governments and countries are relying more heavily on the financial services in terms of banking and the stock market than traditional exports such as the manufacturing industry.

The latest figures from the CIA world fact book (Central Intelligence Agency, 2008) shows that only 18.2% of the UK workforce works in the industry sector, which once accounted for more that 40% of the UK's output and has been falling steadily for the past 30 years (Clark, E. 2002). The fact book reflects this slump in industry with a figure of 80.4% of the UK workforce working in the services sector, as estimated in 2006, these services being made up from many of the financial services including banking and insurance. These figures highlight the size and importance of the financial services currently and support the need to improve the sector via any means possible. Thus with such a large workforce and portion the economy relying on this sector, the need for automation, where possible, in terms of IT running and controlling the day to day processes is also growing in importance.

The UC4 Software research (UC4 Software, 2008) shows how Batch scheduling, sometimes referred to as Job Scheduling, is a very effective method of improving an organisations daily processes. Batch or Job scheduling is not a new concept in the world of computing. It has a long history in computing which was initially twinned with the traditional mainframe computer. The history of batch scheduling as highlighted in A.D. Kallal's work (D. Kallal, A. 2007) shows that the term Batch scheduling refers to the traditional punched cards, which were often stacked and used in original systems. This method of batch scheduling was critical to running the start up, close down and overnight processes for businesses. It is the evolution within IT, in particular the mainframe architecture, which has lead to the importance of batch scheduling playing a larger role within many organisations. From the punched card system through to mainframe and now open systems batch scheduling has played a part in each stage of this evolution and has a large future within IT. Many of the tools available including BMC Control M (BMC Website, 2008), Dollar universe (ORSYP, 2009) and AutoSys (CA, 2007) were created originally to handle the processes for mainframe systems but has now evolved into desktop scheduling software running applications, jobs and processes across networks often running thousands of jobs per day.

The mainframe scheduling focused around the IBM mainframes particularly from the early 1980's and Job Control Language, often referred to as JCL. JCL is a scripting language used to initiate and control the system in terms of how to run the batch (Crosby, R. W., & Greiner, C. 2006). This mainframe scheduling solution was used in the early development of systems to allow for job automation without human interaction. However with the evolution of computer systems and infrastructure it eventually became a very dated and often unworkable solution to a large scale problem as shown in Patrick Weavers research into scheduling history (Patrick Weaver, 2006). Weavers explains how the mainframe has changed to an integrated system of desktop machines used on a network thus the traditional stand alone mainframe and the scheduling software has had to evolve with it.

It was the evolution of the mainframe explained by Crosby, R.W. and Greiner, C. (2006) from monolithic, running a single operating system/environment, to complex multi platform multinational systems on Geographically Dispersed Parallel networks running thousands of complex systems and processes each day which increased the need for automation. This was previously not possible until the mainframe technology evolved which introduced a greater need for batch scheduling. This has in turn changed the mainframe from the traditional isolated stand alone computer into a well integrated part of many organisations IT infrastructure.

This evolution of mainframe management and architectures has lead to a need for automation across systems and platforms within organisations due to processes too complex for manual user intervention. Thousands of jobs interacting globally across different platforms and applications have led to batch scheduling tools being developed to help aid the control process.

Many of the scheduling solutions available offer the same basic features with variations between vendors. The complexity of the tools may often leave organisations overlooking the benefits they offer if not cause the organisation to avoid to tools and technology completely. The infrastructure, costs, user interface and

requirements which each tool aims to meet all differ and this can make choosing a tool all the more difficult. It is finding the correct tool to meet the needs of the organisation and ensure that it is useable for the staff to improve productivity which is important. This research in particular will take the form of a case study of a large financial organisation of global scale which is currently using a number of batch scheduling solutions. This will allow for an evaluation of batch scheduling solutions to identify the tool(s) which best meet the business and user needs for the company's specific requirements.

1.2 Aims and Objectives

For the purpose of the research the company, which the case study is based upon, will be known as company X for the purpose of anonymity. The aim of the project is to identify the best batch scheduling solution for company X. To achieve this an investigation, in the form of a case study, of company X will be undertaken to identify the batch scheduling tools in use and evaluate them. Company X is a large global financial organisation which plays a major role in the financial markets around the world with thousands of clients to which they offer a vast array of financial solutions and advice. It is a market leader which is using cutting edge technology to help lead the way and stay in touch with competitors and changing global conditions. Batch scheduling is used on a large scale within company X; on average they run approximately 750,000 jobs a day globally across 4 tools, meaning the choice of tool and its performance is extremely important to their operations. The author of this research has undertook a 3 month summer placement within the batch scheduling team of company X during the summer of 2008 hence the particular interest and knowledge within this field of research.

The research will look to evaluate the 4 main tools in use within company X these being Control M (BMC Website, 2008), Dollar universe[1] (ORSYP, 2009) AutoSys (CA, 2007) and Crontabs (Whil Hentzen, 2004). This evaluation will take part in the form of an HCI and requirement analysis of these four tools. This will be to identify the differences and features between tools to help identify the best choice of tool for company X. The HCI evaluation will identify the tool which allows users to use it best to their advantage via methods researched through a literature review. The requirements analysis will look to develop a set of standard requirements for batch scheduling solutions and analyse the tools against these. These requirements will be developed via reviewing the current software solutions and manuals along with research into schedulers found in literature.

Each of the products which are to be evaluated are COTS (Commercial off-the-shelf) products used within many organisations which company X either currently have in operation throughout the IT infrastructure or could adopt as an alternative tool for their scheduling should the findings prove its value. Of these 4 tools 2 are seen as possible options to use company wide. The tools in use at the moment are Dollar Universe ($U) currently being phased out within the European region for

[1] Pre Version 5 before graphical monitoring and development was introduced in future versions. The evaluation will take place on the command based Dollar Universe system which is used in company X.

the preferred choice of Control-M, who's vendor is the Israeli company BMC and supported by company X's US operations team. The other tools are Cron tabs which are used simply to control many of the Linux/ UNIX jobs which run across the applications and do not require specific scheduling, the other is CA AutoSys used within the US arm of company X to control the US applications which are developed there. The reason for the company having these 4 batch scheduling tools in place can be traced to historical records of company X not standardising the batch scheduling processes. Different tools were introduced at different time periods to on an ad hoc basis to run the various applications and systems at the time of development. This standardisation has only recently been sought after across all regions where batch scheduling is in place. The introduction of standardisation would allow for a central operations and training process for the company on a global scale.

A number of tasks must be worked through to reach a final outcome of the research these include:

- Identify general batch scheduling requirements via an in depth review of literature to analyse and identify the typical requirements of batch scheduling. This will allow for a general overview of what batch scheduling solutions should offer to a client and which features are more important or useful than others. This will then allow the development of a checklist of standard and advanced requirements which a tool should aim to meet. This will be important to the project as it gives a landscape view of the tools and features of batch scheduling.

- Refine more detailed batch scheduling requirements for businesses such as company X this will be done via a case study of company X and their needs with regards to batch scheduling. This may include interviews with those within the company. This will help to focus the evaluation of the software by identifying the current tools in use and the situation regarding the batch scheduling within the company. The research can then be tailored to try and best cover the needs of the company and offer valid and useful conclusions. This will produce a developed list of requirements which a good batch scheduling tool should try and match. This will be done via collating the results of the literature review and initial case study of company X. This will allow for in depth requirements to be set out for comparison with each tool. This checklist will then be used as a valid evaluation technique via assessing each tool and the requirements which it can or cannot meet. This will help to prove the effect the tool would or does have within company X based on the features and power it has available.

- Identify HCI evaluation methods and techniques for software evaluations via the literature review whereby the author can identify the most appropriate method for evaluating the tools interface to conclude which tool best meets the user's needs. This will help identify the options for performing an HCI evaluation and focus the research on one or more method which will be best effective when performing an HCI evaluation on the case study tools in question.

- Develop a suitable HCI evaluation method for the research via using the results of the literature review to then develop an evaluation which each tool can be evaluated against and conclusions drawn. This will be used as the HCI evaluation

for the project and hopefully provide valid outcomes and conclusions based on the method developed with understanding and knowledge from the literature review being used as the basis for it.

- Evaluate the tools used for batch scheduling in company X both via the requirements checklist and the HCI evaluation to give outcomes and results. This will be the 2 part evaluation of the HCI and requirements which will ultimately allow for the research to gather and develop outcomes and conclusions via these2 main data gathering techniques.

- Final analysis of the evaluations findings will allow for outcomes and conclusions to be drawn based on company X and the tools in question. This will allow for the project research to offer its findings on batch scheduling solutions and that which is best suited to company X. It will also help to eliminate those which are seen as less useful or effective to the company and allow for future plans to be made.

1.3 Research Question

Which of the 4 batch scheduling tools identified would best meet the business needs of company X?

1.4 Hypothesis

The hypothesis drawn from the research in the authors mind is that:

- An appropriate tool will match all or most of the requirements of company X based upon a review of each tool against a developed set of requirements identified via the literature into basic batch scheduling features a tool should offer.

This is simply as the most effective tool, in terms of features, will cover the most requirements from the developed checklist. This should be testable via a developed method and results which the author can collate to help prove or disprove.

- The best suited tool of choice will have a user interface which the staff believes to be the best quality in terms of meeting their wishes as a tool for batch scheduling.

This will be measured via and HCI evaluation of the tool. This method is suitable and relevant to this project as experts will be evaluating the software in their environment to assess which tool would best suit their working needs. Günther Gediga, Kai-Christoph Hamborg & Ivo Düntsch's work (Günther Gediga, Kai-Christoph Hamborg & Ivo Düntsch, 2001) indicates that an expert evaluation is often the best approach to a case study evaluation where the question is "which is better?" and comparing tools relative to each other.

1.5 Remaining Research

The remaining sections of the research and report are aimed at testing the above hypothesis and completing the aims and objectives which have been outlined. Chapter 2 should look to investigate via a literature review the various aspects of batch scheduling and the proposed methods of evaluation, both the requirements of batch scheduling and the evaluation of a user interface. This will be done in order for the author to develop a clear understanding of how best to perform the evaluation of the software. This will then lead into chapter 3, the methods section which will outline how the author will perform the evaluation based on the findings of the literature review in chapter 2. Chapter 4 will present the results and findings of the methods/ evaluations which have been undertaken before final analysis and conclusions are drawn within chapter 5 conclusions.

2 Literature Review

The literature review is research into the surrounding and related topics of batch scheduling and HCI techniques of which the user must be familiar with before conducting the evaluation. This review of current methods and tools will allow for decisions to be made and conclusions drawn helping focus the main project research and validate the findings. The literature review should identify the various user interface evaluation techniques which will allow for a decision to be taken as to the best method for this research. It will also aim to identify the general requirements for batch scheduling tools which will be used to help develop the checklist for use as part of the methods of the research.

2.1 Batch Scheduling Overview

The traditional batch scheduling systems which have been developed over the years all have a general and usually equal set of requirements which they aim to match with the use of the software as indicated in AutoSys research (CA, 2007). Vendors such as BMC, Tidal and AutoSys who have been producing software to manage automatic batches for well over 20 years have analysed, constantly re-assessed the customers needs and have researched the area of batch scheduling producing white papers and documents showing the research and benefits of improvements which have been introduced over the years. The Gartner report of 2000 claimed that "Through 2005, more than 50% of an enterprises inter-application interfaces will continue to use batch data exchange design" (Gartner Group, 2000), later proved correct with the Gartner report of 2007 (Gartner, 2007). This initial claim backed the need for improved and advanced batch scheduling solutions across business applications almost a decade ago, clearly indicating the importance of batch scheduling solutions for the modern day throughout 2005 and beyond.

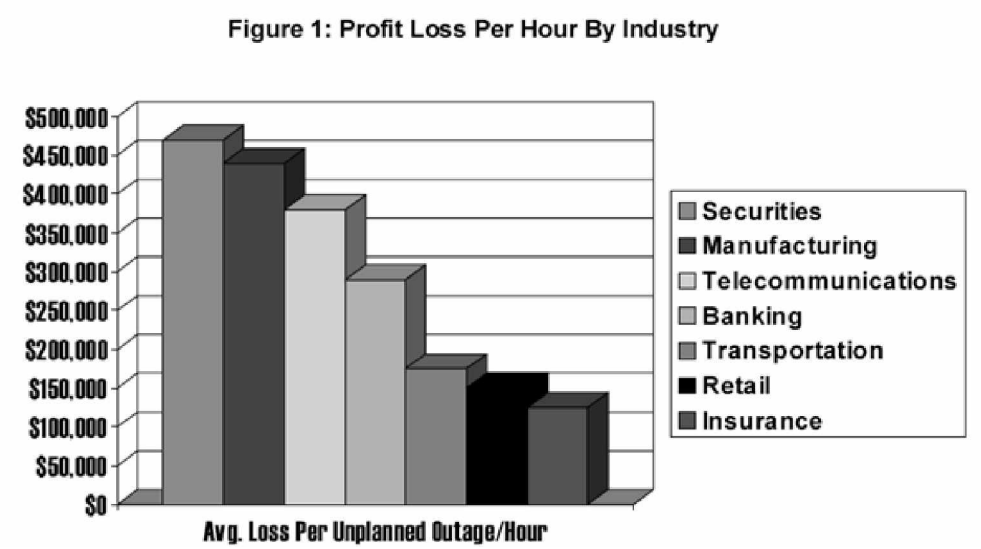
Ultimately all batch scheduling software must try and match the same general needs of the users and perform certain tasks or offer features to help aid the users batch scheduling. Batch scheduling, as outlined in the introduction section, is a tool which is used to help automate jobs on a computer which would usually require manual user interaction. Some examples of these jobs, taken from Patrick Weaver's overview of scheduling history (Patrick Weaver, 2006) include:

- Simple file maintenance jobs which are copying moving or deleting files within the system in relation to the application it is controlling.
- Daily start up processing jobs which are often used by banks and insurance firms to help balance the business figures and stocks for the new day ahead.
- Archiving jobs often used to backup, restore or pull information from historical data in the system in relation to the system request.
- Or ad hoc jobs which can be scheduled or triggered to run on ad hoc basis

These are simply a few examples of tasks which prior to batch scheduling software, required a user to manually operate and process. Tim Grieser (Tim Grieser, 2007)

explains how this often required a large teams of users who would follow a strict timetable and plan work together to ensure that the thousands of jobs required to run per day were all running as and when they should. This was a repetitive dull and mundane method of running the batch however it was the only option and is sometimes, yet rarely used. It was recently still used such as in Hong Kong with the HSBC bank up until 2004 (Arif Mohamed, 2006). This repetitive and important timetable of jobs which must be followed day in day out with only slight variations between days and ad hoc scheduling as required often allowed for human error to interfere with the running of the strict batches. Holidays, sickness and general human error would all play a role in affecting an application or group of applications which were requiring constant user interaction to help them operate as designed (BMC White Paper Publication, 2002). As illustrated below you can see the costs to business per hour for unplanned outage. This unplanned outage could be caused by the error or problem with a critical batch.

Figure 1: Profit Loss Per Hour By Industry



Taken from BMC White Paper Publication, 2002 - source Forrester Research/ Fletcher Research

2.2 Types of Batch Scheduling and Typical Business Requirements

ORSYP, vendor of Dollar Universe, research (ORSYP resource centre, 2009) shows that Batch scheduling tools or solutions can be distinguished as 1 of 3 types: Native, Basic and Advanced. These can be viewed as 3 levels of functionality:

- Native batch scheduling such as Windows scheduled tasks and Unix Crontab offer limited functionally when defining and running jobs and are usually for local processes to a single machine only.

- Basic batch scheduling usually cover a more comprehensive set of features such as job dependencies, exotic scheduling of jobs for specific days/months, and the reduction of gaps in the batch via the removal of time overruns. This

sort of batch scheduling is often suited to many organisation which are small to medium sized and are not dependant on a global IT infrastructure shown in Gartner and CA research (Gartner Group, 2000, CA,2007)

- Finally there is advanced scheduling which incorporates all that of native and basic and includes event driven scheduling to trigger the start of jobs based on specific events, cross-platform and cross application communication and services to tie a number of applications and processes together, and real time monitoring of all jobs and schedules on the system on all servers shown in ORSYP's white paper (ORSYP, 2008).

This project research will look into a cross selection of the types of schedulers which are relevant to company X in order to provide a fair selection of tools and provide an effective and efficient solution to the problem. Granted the size of company X may suggest that native and even basic are simply not effective the research will aim to clarify what type best suits and offer justification for the tool(s) which have been assessed. Of the four tools only Control-M and AutoSys can be seen as advanced due to the nature of the tool. Cron falls closer to the native and Dollar Universe is best suited to the basic style when comparing the tools ability against the 3 styles and examples outlined by ORSYP (ORSYP resource centre, 2009).

The requirements of batch scheduling are often rather technical and complicated to a non expert user who is new to the software. However there are a number of features offered by most of the vendors which can be seen as typical requirements for batch scheduling including BMC, AutoSys, and Dollar universe. It can be taken from the various features and selling points of each vendor along with the user manuals, research and white papers (BMC Manual, 2005, ORSYP, 2008, Gartner, 2007 et.al) that go with the software that there are a number of requirements which are the typical or norm for batch scheduling. This along with evaluations of batch scheduling software performed by tidal software (tidal, 2008) offer a general basis for what batch scheduling solutions should offer. The research of the tidal paper (tidal, 2008) along with the detailed overview of the 4 products in question has allowed for a deep understanding of the requirements which batch scheduling aims to meet.

This research has also taken into account the 3 month placement spent at company X working within the batch scheduling team. During this placement both training and working with these tools has allowed for an insight into the need for and purpose of the tools and has aided the research of requirements for batch scheduling for this project. This training took place in both Glasgow and London and included one to one training with a BMC and independent batch scheduling consultant.

The Tidal paper (Tidal, 2008) shows there are a number of areas which a batch scheduling tool should look to cover. These areas include Queues: Limits and Prioritization, Event-driven Processing and Dependencies. These have been developed based on the needs of Tidal clients and the software processes which they

have in place to improve their software. It is the needs of business being fed through the software development lifecycle based on the client's needs which has lead to the vendors developing checklists which allow companies to asses their needs against the features the software provides. The UC4 white paper (UC4 software, 2008) also outlines similar areas which a batch scheduling solution should offer, areas such as:

- Workload balancing – Similar to that of prioritization from the tidal checklist
- Dynamic and data driven automation – similar to event-driven processing previously shown within the tidal checklist
- And job execution/ recovery

The UC4 paper also gives an evaluation matrix which has 8 categories these being: General, Schedule Creation, Event Management, Job Execution/Recovery, Graphical User Interface, Output Management, Security and Application Integration. These have been identified as the categories of which the main features or requirements of the tool fall under based on UC4 research and software needs of their clients. Similarly ActiveBatch have created a checklist (ActiveBatch, 2006) and MVPSI Systems likewise (ho, 2007) both are experienced batch scheduling vendors who have an understanding of the users needs. These checklists all cover the same similar areas although using different terminology or phrasing but all aim to provide similar features such as alerting. These have therefore been key to help gain an understanding of batch scheduling requirements and have been incorporated in the development of the appendix of requirements.

Along side these checklists which the literature review has identified the authors 3 month placement within company X has helped understand and developed a clear understanding of the sorts of requirements which were required specific to the company. These requirements which the author identified were developed through the training and interviews with clients and managers within the company along side using batch scheduling software on a day to day basis to develop and create schedules and solutions. This time within company X thus gave the author a clear understanding of the areas which batch scheduling solutions must cater. The area such as alerting specific email or communication devices on the completion or failure of a job is something which was important to the company. Many other areas of requirements are also important to the company such as the ability to have data driven and event driven scheduling.

With this in mind the author has developed a list of requirements which fall under 10 categories. The categories have been identified via the research the checklists which vendors have produced along side the authors own knowledge from within company X's batch scheduling team. The 10 categories outlined below are all important to company X in one way or another.

The main areas of requirements for batch scheduling which have been broken down are shown below in no particular order of importance:

- Alerting
- Event Scheduling and Control
- General Software Requirements regarding batch scheduling
- Job Execution and Recovery
- Jobs and Schedules
- Schedule Restraints and Dependencies
- Schedule and Job History
- Schedule Capabilities
- User Interface
- Variable and Parameter passing

Below is a table of all ten categories and the sources, being the vendor checklists, where the author identified similar or relevant categories which have then defined the authors own 10 categories shown above.

	Tidal	**UC4**	**Active Batch**	**MVPSI Systems**
Alerting	Alert Management	Alerts	System Alerting	Reporting
Event Scheduling & Control	Queues/ Event Driven Processing	Dynamic, Data-Driven Automation	Events	Scheduling
General Software Requirements		General	General	
Job Execution & Recovery	Auto Recovery/ Fault Tolerance		Job Execution & Recovery	Exception management
Jobs & Schedules	Queues	Schedule Creation	Job &Job Plans	
Schedule Restraints & Dependencies	Dependencies		Scheduling	Scheduling
Schedule & Job History	Audit Trails		Auditing	Auditing
Schedule Capabilities		Workload Balancing		Workload Management
User Interface	Ease of Use/ Virtual Resources	End-user involvement		
Variable & Parameter Passing	Database Support		Variables & Dynamic Data Passing	Workflow Foundation Support

Each of the 10 categories has been developed via the collation of various sources shown above. Each category is explained in more detail below:

Alerting

Alerting is the mechanism for making aware any problems or issues which a user may wish to monitor. This includes if a schedule has stopped running or has completed which ultimately could affect a business if it is not dealt with in an appropriate manner or timescale. All batch scheduling vendor checklists outlined previous refer to alerting in one form or another as it is a key aspect of batch scheduling. Ultimately the alerting process is what helps to monitor an automated process to ensure it's successful.

Event Scheduling and Control

Event scheduling and control is another category which can be seen as important to batch scheduling outlined in the various sources highlighted previous. This area of batch scheduling allows for the ability to schedule events or jobs and control the activity of these within each schedule based on specific triggers or situations. An example of this would be the ability to trigger an event which would run a clean up script on a database triggered by an alert from another job. This is what event scheduling and control incorporates in relation to batch scheduling. The ability to control the trigger of jobs and schedules is important to company X as there are thousands of jobs per day running on batch scheduling software.

General Software Requirements regarding batch scheduling

There are a number of general requirements which cannot be specifically classed under any of the other categories however they are important to the overall software solution. Requirements such as the ability to view both windows and non-windows applications running via the batch scheduling tool on one instance. This is a feature which most although not all tools are now offering however, it is something that businesses such as company X want along side those which fall under non specific categories within the vendor checklists found during the literature review.

Job Execution and Recovery

The category of job execution and recovery has been identified due to the need to show how each tool handles jobs and more importantly the failure of jobs and the restart process. This is mainly as the ability to restart jobs and view the archive of jobs which have or have not completed is important to a business which need systems to be active 24/7. The job executions are also important to understand how jobs will be queued and prioritized when running on servers. This is an important requirement to all batch scheduling solutions and is a running trend though most software providers as it is important for failed jobs or schedules to be handled in an automated process.

Jobs and Schedules

Jobs and schedules is another important area regarding batch scheduling. All batch scheduling solutions have jobs which are managed via schedules so naturally it is important to asses how each tool handles jobs, what sort of jobs and how they can be scheduled. Again this is a running trend within all vendors of scheduling software as the jobs and schedulers are ultimately what are being produced and controlled. The need to understand how jobs and schedules can be managed is important to company X.

Scheduling Restraints &Dependencies

Restraints and dependencies all differ between software solutions and the differing dependencies between jobs can be important to a business. The ability to not only schedule jobs based on time frames but also on the completion or failure of other jobs is something which is important to help automate and process schedules. Even when issues and problems arrive it aids the efficiency of a schedule by reducing white space time gaps. Company X specifically require for exotic (non time based) scheduling thus it is important to asses how tools allow for dependencies to be scheduled.

Schedule and Job History

The ability to monitor and view archive jobs from the system allows for a business to analyse, improve and trace faults through an automated process. Therefore the ability to view schedule and job histories is beneficial to a company. This is another area which is important to most vendor solutions as it plays an important role in reviewing and maintaining schedules by allowing the users to review a schedule and edit it based on its run performance. This is also a key fault finding tool for scheduling allowing users to trace faults and alter schedules to suit, something which a company such as company X will need.

Scheduling Capabilities

The scheduling capabilities cover various aspects of how schedules can be run i.e. the ability to schedule jobs and plans based on specific business calendars or situations. This is important to company X who require for specific calendar events based on the business year and region which it is running. The reason being company X is dependant on the financial calendars for each region which their applications are running. Thus the ability to create multiple schedules for the same application running across various regions by simply altering the calendar which it follows is something company X would benefit from.

User Interface

Along side an HCI evaluation which will be performed separately there have been a few key requirements of the software which specifically company X require. These have been identified as improvements and useful tools which the users within

company X prefer to have. These have been included to ensure each tool has the ability to meet company X's needs.

Variable and Parameter Passing

The ability to pass parameters and variables through commands and applications running on the batch scheduling software is important to all software solutions. This allows for flexible scheduling and parameter passing which is something that batches require in order for jobs to process data and files in an effective manner. This has been identified as a key category through the literature review of the vendor checklists.

Within each of the 10 categories there have been a number of sub requirements which fall under 1of the 10 categories identified above. These sub requirements are individual detailed requirements or features which the tool should try to accommodate one way or another. These sub requirements were identified through the research of the manuals of the 4 tools (BMC Manual, 2005, Unicenter, 2006, Whil Hentzen, 2004 ORSYP, 2008). Not all requirements will or can be met by each tool. This checklist will be the foundation of the requirements matching exercise. The detailed checklist developed uses this categorisation within the methods section to develop a complete requirements matching evaluation through a process involving the technical experts within company X.

2.3 Software HCI Evaluation Techniques

The Research will not only look into the requirements of batch scheduling software but also try and look to the HCI factors of software to evaluate what best suits the users of the tool within company X. This is to evaluate the user interface of the system in question and allow for comparisons. To do this various HCI techniques have to be viewed allowing the best approach to be taken for the project.

There are various methods and reasons for a Human Computer Interaction (HCI) Evaluation to be undertaken upon a software product. Günther Gediga, Kai-Christoph Hamborg & Ivo Düntsch (2001) suggest that the goals or aims of the evaluation can be characterised in one or more of the following 3 questions:

1. "Which one is better?" This usually looks at a number of similar tools or software products and aims to compare them for the best overall in terms of HCI for the job.

2. "How good is it?" This aims to evaluate a single software product against various usability techniques and standards to certify or simply review the product to asses its use and validate how well it performs its specific purpose.

3. "Why is it bad?" This evaluation looks at why a software product is not seen as good or useable and aims to suggest improvements to the flaws to help aid a re-engineering process, or simply learn from the mistakes in the software for future projects.

The research also shows that goal 1 and 2 are seen as summative evaluation, while goal 3 is a form of formative evaluation approach. Summative is used in almost complete or fully produced software products and does not offer any outcomes for improvements or changes to the design. Where as formative aims to give feedback to a developing system to try and improve it before it reaches the publication stage of its lifecycle as indicated by the work of Scriven, Hix and Hartson (Scriven, M. 1967, Hix, D. & Hartson, H. R. 1993). This would therefore suggest that this research will require a summative evaluation method as the tools being evaluated in question are all fully developed and in use therefore comes under that of number 1 or 2 above.

Sabine McConnell (2006) along with Günther Gediga, Kai-Christoph Hamborg & Ivo Düntsch (2001) believe that the techniques can be broken down further into two categories either descriptive evaluation techniques or predictive evaluation techniques. The descriptive evaluations fall under the summative style of evaluation in that the users are asked to try and identify and comment on the actual problems or highlights within the software. Predictive evaluation techniques are more linked with the formative style of evaluation which are usually performed in the initial stages of development or design of the software product and look to identify recommendations and problems areas of the design for future versions or developments.

The predictive evaluation style differs from the descriptive style as predictive looks to analyse and evaluate products or systems earlier in the development stage to learn of and identify any problems before prototyping or screen development. However predictive evaluation types do not require any form of developed system to be in place in order to evaluate it. It involves a more theory based approach through a system to identify what

should be done in practice when developing the system as outlined in the research by Günther Gediga, Kai-Christoph Hamborg & Ivo Düntsch (2001).

As the author has identified this predictive style of evaluation will not help to identify "which one is best" and has been reviewed at a high level and disregarded as a possible tool for evaluating the batch scheduling tools in question. However further details and information on two predictive methods can be found under the research of Wayne D. Gray & Marilyn C. Salzman (1998), Andreas Holzinger (2005) and Gerald Kotonya & Ian Sommerville (1998) with regards to usability walkthroughs. Another method of evaluation is that of Expert inspection or heuristics evaluation which can be found in the works of Robin Jeffries & Heather Desurvire (1992).

The author therefore believes these methods are not suitable as the outcome will have no effect on helping to identify which tool is better at performing the required tasks. The aim of the research is not to identify improvements which can be made to the scheduling software so it will not use any of the predictive evaluation styles.

2.3.1 Descriptive Evaluations

Within the descriptive evaluations the techniques used are mainly the following: Behaviour based evaluation techniques, Opinion based evaluation methods and usability testing as explained by Wayne D. Gray & Marilyn C. Salzman (Wayne D. Gray & Marilyn C. Salzman, 1998). Behaviour based techniques include observations, think aloud and video methods. Where as opinion based evaluation methods include questionnaires, interviews, and QUIS methods.

The behaviour based methods are noted at a high level. The author has identified that the nature of the evaluations would require expert knowledge, which the author does not have, to fully asses the tools in a valid manner. The types of evaluation are also time consuming in terms of developing and performing due to them requiring a laboratory style set up so have been disregarder as possible techniques for this research. Given that the nature of company X being a well established organisation with an experienced user group it would be more beneficial to the research to take their opinions and views into account. This coupled with the fact that due to geographically dispersion of the user group setting up experiments would be logistically impossible, has led to the behaviour based techniques being disregarded. Some examples of behaviour based evaluation techniques can be found below:

Behaviour based evaluation techniques

Wayne D. Gray & Marilyn C. Salzman (Wayne D. Gray & Marilyn C. Salzman, 1998) outline the behaviour based evaluation techniques as being observation of the user's interaction with a system and require a form of the software or system to be in place before the evaluation can take place. This can be in the form of a prototype or fully developed working system. Therefore these forms of evaluation can only take place in the latter stages of a systems lifecycle.

For this reason only specific techniques can fall under this category, 3 of which are: Observations, Think aloud and Video Methods.

Observations

One method of behaviour based evaluations, as shown in H. Rex Hartson's work (Hartson, H. R. 1996), is user observations which are often one of the first forms of behaviour based evaluation undertaken. This involves observing how a user interacts with a system and allows for a various selection of data to be taken from the process. Conclusions can be drawn by observing how easily and quickly the user can understand how to use and interact with the software to perform the chosen task. Günther Gediga, Kai-Christoph Hamborg & Ivo Düntsch's work (Günther Gediga, Kai-Christoph Hamborg & Ivo Düntsch, 2001) also explain observations in greater detail.

Think aloud

The think aloud method of evaluation is another form of behaviour based evaluation. The think aloud method is almost an extended version of observation. C, Lewis's research (Lewis, C. 1982) shows that the think aloud method for evaluating software often follows a similar method as the observations in terms of setting up a scenario for a user using a system. The difference being the user has to talk through their though processes while interacting. This method however can have various disadvantages as indicated by R. D., Smith, M. C., Podd, J. & Varela-Alvarez's work (R. D., Smith, M. C., Podd, J. & Varela-Alvarez, H. 1995)

Video methods

Another way of incorporating the observations or think aloud is to record via videoing the users in a lab type environment. The user is recorder interacting with the system replacing the physical presence of the observer being next to the user as they use the system. This then allows for the evaluator to analyse the interaction and then interview the user or question them at a later point on the various areas which they have noted(Greenbaum, J. & Kyng, M. 1991).

However the author believes an opinion based evaluation may be the better approach to the research in terms of effectiveness allowing for a better collection of data based on the approach and methods which they provide. Given that the author has access to expert users via the case study, whose opinions are valued for this research; it would make sense to use their knowledge and understanding when assessing the tools. Company X has a long established batch scheduling user group who collectively have experience and opinions which are valuable to this research. By using their knowledge and expertises within this domain the author feels opinion based evaluations will allow for this quality feedback to be obtained and analysed giving a far greater overall conclusions and results to the research. Therefore the opinion based evaluation methods would be more beneficial to the project.

Opinion based evaluations

Meister's work (David Meister**,** 2003) suggests that the opinion based style of evaluations offer a more in depth and comprehensive feedback between user and evaluator. This method of evaluation allows for the users and the evaluators to offer their views, opinions and questions in relation to the software or system in question. This 2 way communication allows for the user and evaluator to gain an in depth

understanding of both the developer or evaluator and the user's needs and frustrations. The use of oral or written evaluation is often found to be the most widely adopted form of opinion based evaluations. Three well known and used methods are: Interviews, Questionnaires and QUIS's which are a more in depth form of questionnaires as highlighted by Greenbaum, J. & Kyng, M (1991).

Interviews

One method of opinion based evaluation is to use interviews. Interviewing users based on their experience with a system to analyse its effectiveness and productivity is a highly useful method of evaluation. This allows for a detailed coverage of the products highlights and problems with the users being able to give qualitative feedback in a subjective evaluation process. This means that the information and data received is far more valuable and in depth than a questionnaire or behaviour based evaluations. David R. Millen (2000) looks at using interviews as one part of a rapid HCI technique where by user requirements, feedback and analysis of a system can be taken at a face to face level to help the direction and findings of a HCI evaluation. This gives both evaluator and user a chance to have their topics covered in a face to face manner.

This method of interviewing can often be costly in terms of time and logistics with the users who must gain an understanding of any software you wish to interview them on. It is also costly in terms of guaranteeing time with both the interviewer and user to perform the actual interview. However where available it is one of the best forms of evaluation that can be undertaken within HCI (Greenbaum, J. & Kyng, M, 1991). Therefore for this research given that some users are in America it would be logistically difficult to perform such an evaluation for this research. For this reason the project will not use this method of evaluation.

Questionnaires

Questionnaires are sometimes the only alternative to interviews to gain both a quantity of data and user views this replaces the quality in depth data which the interview can give an evaluation a thought shared with Jurek Kirakowski, (2000). This replacement by no means can be seen as being as comprehensive as interviews however; it does offer the option for those situations where interviews are not possible often due to geographical dispersion or budget allowances. Afke Donker & Panos Markopoulos, (2002) researched 4 HCI techniques including questionnaires. They found that the difference between interview and questionnaire did not differ significantly although differences and issues were noted between the techniques such as the depth of feedback to a question.

Often geographical dispersal leads to questionnaires being adopted over interviews. The ability to email, post or set up an online questionnaire and have a good response from those which are not reachable via interview face to face due to lack of budget or time restraints allows for data to still be collected. Jurek Kirakowski, (2000) speaks more about questionnaires and the advantages to them. This method of evaluation is therefore seen as a possible and appropriate data collection method for this research as it allows for data to be collected across all user groups. Therefore this method has not been disregarded as a possibility.

QUIS: Questionnaire for User Interface Satisfaction

The QUIS approach developed by Chin, J. P., Diehl, V. A., & Norman, K. L. (1988) is used to gauge user satisfaction on the interaction with a system and is another form of opinion based evaluation. This method has been further developed by these workers at the Human / Computer Interaction Lab (HCIL) within the Maryland University. This form of QUIS tool is now available from the University Of Maryland Office Of Technology Commercialization which entitles the user to extract the appropriate questions from a list developed by the team to help build and construct a suitable QUIS evaluation for their requirements. Their current version of QUIS, version 7.0, contains a demographic questionnaire, a measure of overall system satisfaction along six scales, and hierarchically organized measures of nine specific interface factors (these include screen factors, installations, feedback, help menu etc.) (QUIS, 2008). This version is available for licensing and uses both online and paper systems.

The purpose of a QUIS is generally:

1. To help identify areas for redesign or improve actual design of a system
2. give managers a method of highlighting possible improvements in the tool
3. provide researchers with a validated instrument for conducting comparative evaluations
4. Or be used as a test method in usability labs

The author believes that, from the above, option 3 would be the purpose of a QUIS for use within this research area, allowing for comparative evaluations between the 4 batch scheduling tools. This also relates back the initial overall goals of an HCI evaluation outlined by Günther Gediga, Kai-Christoph Hamborg & Ivo Düntsch (2001) where by the purpose of this research would follow the goal “Which one is better?”. This would allow for the tools to be evaluated equally and assessed in comparison with an outcome determining the best tool. It would also allow for the expert users within company X to take part in the evaluation process thus add validity and experience to the results.

The user is presented with a series of questions based on various aspects of the software such as the screen layout, the difficulty of use, the learning process and general use of the system. These are answered on a rating scale to provide numerical feedback, this feedback can then be analysed and compared across a selection of tools and users to help identify which tool best addresses the users needs. This method would be particularly effective when comparing different tools already developed. James R. Lewis (1993) shows how the method can be effectively used to gauge a client response to a software product. This style of approach best suits software which is in the later or completed stages of development as it requires a system to be in place in order for users to rate how each category of the tool performs. The full Maryland measures nine specific interface factors these being:

- Screen factors,
- terminology and system feedback,
- learning factors,
- system capabilities,

- technical manuals,
- on-line tutorials,
- multimedia,
- teleconferencing,
- and software installation

Not all of these areas are relevant or important to every software evaluation. Thus each section has to be reviewed and tailored to the needs of each specific evaluation. The author believes that tailoring a QUIS to incorporate the key areas which are relevant to the batch scheduling tools such as screen factors, system feedback and learning factors could help identify the differences between each tool in terms of the users satisfaction and ability to use the software easily.

It is therefore shown that the QUIS style of approach would best suit this research as it would allow for expert users to give their expert opinion on the software which they are working with on a daily basis. It will allow for the users to give a satisfaction rating on various areas which can be assessed to allow for conclusions to be drawn which will help to identify the most effective and best suited tool for company X.

However along side the QUIS itself a suitable framework should also be referred to in order to develop a valid QUIS with appropriate questions. An evaluation model/ framework can be identified which will help identify the areas which questions should cover and add validity to the QUIS itself. This is to ensure the questions are valid and comprehensive to an HCI evaluation. Evaluation models and frameworks are outlined below.

2.3.3 Evaluation Models/ Framework

The evaluation models or frameworks as they may be known differ from the techniques as they generally set out the standards or checklist of key HCI issues which software must cover to be effective. A number of these models have been developed by experts in the field over the years such as Jacob Nielson (Jacob Nielson, 2009) which have set out a number of qualities which should be covered and addressed in order to ensure software is usable. These can be used in conjunction with a technique to ensure the appropriate areas of software are being assessed.

ISO Metrics

ISO metrics are a framework of evaluation standards which can be applied in conjunction with an appropriate technique to conduct an HCI evaluation. These ISO metrics have been developed in conjunction and licensed by the International Organization for Standardization (ISO). Based upon the research and studies that they and professionals have conducted they have developed various standards covering almost all aspects of products and regulations and this includes that of software. The ISO was set up in 1947 and is located in Geneva Switzerland; it was set up to support and facilitate international trade.

“Standards ensure desirable characteristics of products and services such as quality, environmental friendliness, safety, reliability, efficiency and interchange ability - and at an economical cost.” (ISO, 2009).

These methods therefore are recognised standards that can be used to evaluate software. The ISO published the ISO IEC 90003:2004 in 2004 to replace the ISO 9000-3 of 1997 software standard. These ISO's explained how to apply the standards to computer software as researched by Praxiom Research Group (2004). However the ISO 9126 is an international standard for the evaluation of software quality. This ISO is separated into the following 4 categories:

- Quality model
- External metrics
- Internal metrics
- Quality in use metrics
  (ISO,2009)

However, the above ISO metrics require purchase of the lengthy legal rights to the document and translating it into a user interpretation of the findings against which the evaluator must construct a suitable evaluation checklist for which to evaluate the software against. Thus it is a lengthy and complicated method of evaluation to an inexperienced evaluator wishing to evaluate software.

R. Oppermanna and H.Reitererb's (1997) approach to using ISO metrics however follows the ISO9241 Evaluator. The key sections of this method can be seen in figure 1 taken from Software Evaluation using the 9241 Evaluator by R. Oppermanna and H. Reitererb, (1997). This is an adaption of the ISO 9241 multi part standard which covers humans using computers. The research took this ISO standard and applied it to the software itself in the method of a software evaluation and found it to be an effective solution. The key sections of the evaluation can be seen below in the diagram. Each of these sections has an extended set of areas to cover found in Oppermanna and Reitererb's research (R. Oppermanna and H. Reitererb, 1997).

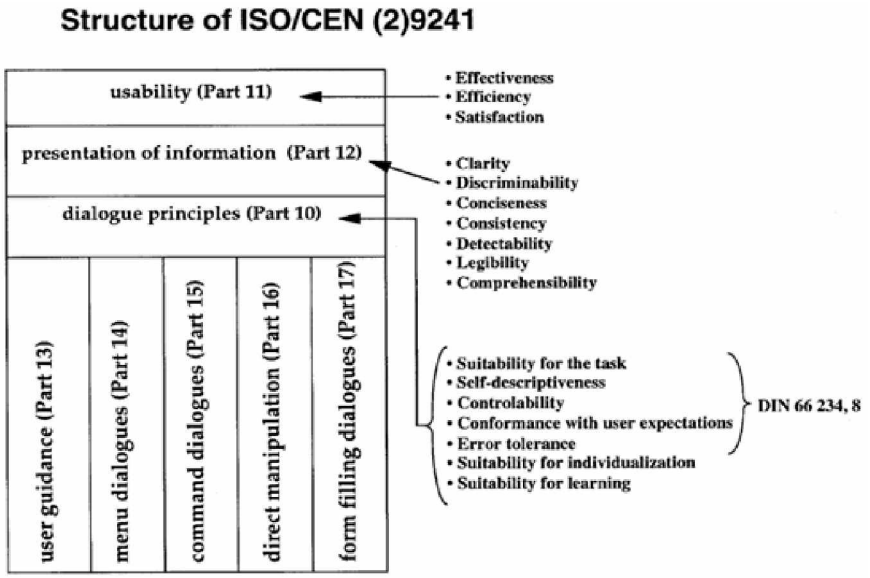


Figure1. Structure of the multi-party standard ISO 9241 (found in the work of R. Oppermanna and H. Reitererb, 1997)

This method following the ISO9241 ties in with the QUIS as the sections which it covers such as satisfaction, layout, suitability for learning and feedback for example are all areas which are also incorporated through the QUIS. Here it can be seen that there is a crossover between both the technique and the framework which is a good marriage. The author therefore feels that the QUIS follows most of the areas of the ISO 9241 and the two should be classed or followed closely when developing a potential QUIS for this research.

The author of the research believes that the ISO 9241 Evaluator by R. Oppermanna and H. Reitererb, (1997) would be a suitable and effective framework from which to base the questions of the QUIS by covering various aspects of the model to gain feedback from the expert users. This is as the framework covers key areas such as the usability, user guidance and visual display of information and dialogue on screen including menus and layouts.

2.4 HCI Methods Conclusion

Various other techniques have been used and developed to try and evaluate software in recent years. As the research in the literature review shows this is simply a selection of methods which can be adopted when evaluating software systems. The selection of techniques and models above are overviews of a vast amount of HCI developments which are in existence throughout the world of computing. It is clear however that a structured technique and model should be referred to for the development and evaluation of a software product. This will help to ensure the end product is usable and also help to choose the correct software for the job.

Based on the research and the literature review it is clear the approach that is required is a descriptive evaluation of the software seeing the products are all off the shelf developed tools and are not in development or require any feedback for improvement. The experts using the software within company X will also require being involved in order to find the best solution to the company specific problems. The experts within company X are using the software and tools on a day to day basis so their knowledge and understanding of the software will be important to the evaluation. This would also allow for the results of the evaluation to be used by company x regarding the best off the shelf product rather than look to point out the flaws in a developing tool for the developers.

The QUIS system along with a framework following the ISO 9241 framework is a suggestion which the author feels could be an effective and suitable marriage in order to develop a good QUIS questionnaire which follows a legitimate framework. This therefore will allow for the distribution and feedback of the questionnaire across company X for relevant and useful feedback which can be evaluated and from which conclusions can be draw. Its fuller development will be covered within the methods section of the report.

This QUIS HCI method allows for the expert opinions compliment the findings of the requirements matching exercise via these expert's views of the software. These should be related in that there is a clear difference between tools which allows for the research to offer its conclusions to company X in a valid manner. The use of company X staff members for their expert opinions helps validate the results and findings.

3 Methods

The methods section will outline the proposed techniques which the author has identified to carry out during the project in order to provide valid outcomes. These methods are those which have or will be performed during the course of the project.

As the project is a case study based approach of company X the author will undertake a case study of company X in relation to the software controlling the batch scheduling within the company. This will include various methods of data gathering and communication with the company. These contacts have already been developed via the 3 month placement undertaken at company X by the author and key contacts including managers within the batch scheduling team have been identified. It is through communication with these managers that the author has identified the methods and tools which are required for best approaching the research. These key contacts are important to the case study as they will provide the access and information which will allow for the results to be valid and a reflection of the company situation. These contacts are important as are any key contacts to a case study in terms of access and communications between researcher and institute or organisation. This has been identified as the best method of communicating the author's tools and methods to company X via email and interviews if required.

The contacts and experience gained by the author has already identified the following within company X:

3.1 Tools

The case study has identified that there is a number of batch scheduling tools used within company X. These tools have been adopted at various times during the development of the IT infrastructure within the company and as such there has not been a "company wide" solution identified. Therefore there have been 4 tools which are used throughout then company globally. The research aims to evaluate and provide an analysis/ evaluation of the tools to identify which best suits the company. This software evaluation, which has 2 parts both the HCI and requirements matching, will allow for an in depth overview and analysis of the results to provide conclusions to the evaluation which will hopefully benefit company X and any decisions which they make regarding batch scheduling within the coming years through the management of company X. It will also provide general results which are related to batch scheduling tools in general. The main contacts within company X are within the batch scheduling team within the European Headquarters which were made during the 3 month placement. These contacts will be heavily involved in the evaluation of the products and the distribution of any tools or questions to other areas of the company to gain feedback for the project.

The tools which have been identified are the following 4:

- Control-M,
- AutoSys,
- Cron
- and Dollar Universe

The reason the 4 tools above have been chosen is due to the current situation within company X. The background to company X's batch scheduling situation is important to the research and its purpose. Currently company X is migrating from Dollar Universe to Control-M within the European region applications. The main European headquarters adopted this migration to Control-M with it being supported by the American branch of technical support mainly dealing with the infrastructure and hardware. However the American arm of the company is more in favour of the AutoSys version of batch scheduling when using a scheduling tool. Therefore the overview of the 4 main tools within company X, including cron which is used to some extent, will allow for company X to view the best option as a company wide tool.

These four tools have been chosen for various reasons. Firstly, they are all used or have been used in recent times within company X thus the staff/users will understand and know the tools which are to be evaluated. They have also all got similar features with the exception of Cron as it is a free non-exotic and less flexible solution for scheduling jobs within UNIX, similar to that of the windows scheduler for windows platform. This leaves a fair evaluation between different types of product which company X has or use to try and find the realistic and best option in terms of requirements matching.

The requirements review will be done separately from an HCI evaluation which will be performed against all 4 tools to evaluate the best possible tool via the requirements and the tools usability. This will then inform company X with any future batch scheduling migration plans as it will provide a company standpoint and argument for or against the current options as well as the requirements matching.

Section 3.2 and 3.3 will outline the process and methods for developing and performing the evaluations on the tools in question. The requirements matching will outline how the requirements have been developed and the technique used to match ass tools with those developed. The HCI evaluation methods section 3.3 will show the process taken to develop and evaluate the tools using the QUIS evaluation technique identified under the literature review.

3.2 Requirements matching

As previously outlined within the literature review research there are 10 categories for which batch scheduling requirements can be classified under. These are shown again below:

- Alerting
- Event Scheduling and Control
- General Software Requirements regarding batch scheduling
- Job Execution and Recovery
- Jobs and Schedules
- Schedule Restraints and Dependencies
- Schedule and Job History
- Schedule Capabilities

- User Interface
- Variable and Parameter passing

These 10 categories each have a set of related relevant yet individual stand, alone requirements which fall under the general category. These sub requirements are detailed features which tools should try to accommodate in order to be an effective batch scheduling solution to company X. For example under the alerting category one specific requirement is that:

- Alert Triggers of Jobs and Job Schedules.

This relates to the tools ability to handle alerts and use alerts as a trigger of further jobs or schedules, for this reason it falls under the category of alerting. This clearly falls under the alerting category as it relates and effects the way in which the software handles alerts. This has been identified as a general requirement which runs through all vendor products and is also important to company X. Another example of a requirement which has been developed is:

- The need for exotic scheduling.

This falls under the Scheduling Restraints & Dependencies category as it relates to the way in which dependencies are created between jobs. The exotic dependencies include event based scheduling other than just time based scheduling and allow for schedules to run more efficiently by reducing the white space and idle time of the system.

These sub requirements have all been developed through analysing the 10 categories and the individual requirements each tool has relating to batch scheduling. These requirements are taken from each vendor manual for the 4 products in question. The requirements have then been placed under one of the 10 categories dependant on its nature. This list has then been validated via consultation with company X's contacts. The vendor checklists were also used to help guide the development requirements of each of the authors 10 categories via referring and relating to each category as seen fit. Company X also has various requirements which have been identified during the 3 month placement undertaken during the summer of 2008. These requirements are those which the company would prefer the tool(s) incorporate. These requirements are mainly incorporated in only advanced and sometimes basic scheduling types and specifically deal with the cross application communications which would allow for tight overall control and monitoring of the jobs used in the tool of choice. This is to ensure that the specific needs of the company and the general requirements are matched as close as they can be to ensure the software or tool is the best option in terms of functionality. This takes into account the size of the Multi-National financial organisation that is company X, who are relying heavily on the IT infrastructure in Europe, mainly the UK, but also America. The requirements are mainly specific to the alerting and the ability for the tool to alert external and 3rd party applications separate from the tools own reporting and alerting mechanism. This along side the ability for

the system to provide a single screen view of all applications running in multi regions in a graphical user display when possible helps to provide specific requirements which a tool must try and match.

The full list of the 91 sub requirements which have been identified collectively across the 10 categories can be found in Appendix A. Each category having its own table of requirements, a full list of with description of each tool has also been included. These have been identified via the literature review and a review of company X as part of the case study. All requirements have been identified via assessing the features of each of the 4 tools alongside the recommended requirements which the vendor checklists incorporate. This has led to the requirements being categorised and included within 1 of the 10 categories. All sub requirements directly relate to the category which it falls under. For example:

- Alerting includes the requirement to alert to a blackberry device via email

This has been derived from the Control-M tool offering this feature and company X using it throughout the new schedules. Thus it is a requirement which is advanced in terms of batch scheduling and required by company X as a useful feature. Therefore it has been included within the alerting category as it alerts users via the blackberry device.

As explained in the research into the requirements of batch scheduling the list of requirements (Appendix A) have been identified against which each tool will be compared to, and developed on the basis of general and specific batch scheduling needs for company X and also generic batch scheduling. These requirements have then be matched with the tools individually to give a summary of the features and requirements which they perform and under what category they best fall out of the 10 batch scheduling categories. The tools have each been studied via the manuals and vendor websites to check each requirement off with each tool. This will allow for each tool to be given a rating or score based on the amount of requirements which it can meet and perform. The higher the number of requirements matched the higher the level of batch scheduling tool from the 3, basic, native and advanced and also the most relevant and suited solution to company X.

Of the 10 categories it has been identified with close consultation with company X that 4 of them are more important to them specifically over the other 4. However the other 4 categories are still important to the general make up of a batch scheduling tool. Thus they are still important in identifying a good batch scheduling solution for the research. The 4 categories are listed below:

1. **Alerting**

   As alerting is ultimately the way in which the tools communicates errors or problems to the users and company it is seen as the most important category from these 4. This is as the company rely on the software informing and

handling any problems effectively to allow for users to intervene and correct any issue which will minimize any downtime.

2. Schedule Restraints and Dependencies

Scheduling restraints and dependencies are seen to company X as important in terms of progressing onto a more efficient tool for batch scheduling. Some of the tools previously or still in use within company X only handle jobs in a time scheduling basis. For example job A takes 10 minutes to run so gets a times slot of 15minutes to allow for an over run. Then job B will start 15 minutes after job A. However this style of scheduling does not efficiently or effectively run as it leaves time gaps and possible errors when jobs do not finish in time. The new style of scheduling allows for job A and B to be linked to ensure job B starts immediately after job A finishes via a dependency between the 2 jobs. This style is what company X are looking to use more across the board and so is important to analyse how the tools each handle jobs and dependencies.

3. Variable and Parameter passing

Variable and parameter passing is another important category as company X has to ensure that the tool can handle cross platform and application communication and processing. A key factor in this is the ability of jobs to carry and pass variables and parameters which are application and job specific. Variables such as commands for scripts or file paths are important to ensure the job performs the correct job on the correct file during the running of a schedule. Therefore this is something which the tool must incorporate.

4. Schedule and Job History

The final priority category which company X identified after analysis of the 10 categories was schedule and job history. This ability to archive and view jobs which have already run or failed is important to help trace faults in schedules and also minimise the down time of a system by identifying the errors in a real time run through. Therefore a tool which can handle the archiving of jobs will be preferred to that which cannot.

However, the remaining 6 categories are still important to company X and batch scheduling overall so they will and must be included in the review of requirements to ensure a comprehensive coverage is incorporated in the requirements review.

The method of requirements matching will allow for the following hypothesis to be tested:

- An appropriate tool will match all or most of the requirements of company X based upon a review of each tool against a developed set of requirements

identified via the literature into basic batch scheduling features a tool should offer.

The results of this requirements matching will be evaluated via a simple scoring system as all requirements on the list are as important and relevant as each other. This is a simple and effective method of calculating the best suited tool in terms of matching requirements. This will form the first part of the two stage evaluation process of batch scheduling tools. It shall also provide an outcome of the first hypothesis. However, the results when presented will be finer grained and place emphasis in the prioritisation company X identified.

The next part of the evaluation will take part in the form of an HCI evaluation. This will be the second part of the 2 part evaluation of the software.

3.3 HCI Evaluation

The HCI evaluation has been identified as an important method of analysis to help understand the user's likes and dislikes when it comes to the tools they are using. As with this case study the users involved are all expert users of the system and the tools in question are all fully developed commercial tools the method of HCI evaluation chosen is the QUIS. This QUIS method as it aims to identify from a group of expert users the tool which they best prefer. This method differs from other HCI techniques as it does not aim to improve or suggest improvements for the tool in questions, it simply aims to identify "which is best" for the task in hand. The evaluation will be developed and reviewed by experts of the systems, one of which is the author, and follows that of the Maryland university example. It will be developed and undertaken to ensure that the tools are reviewed by the users to identify the best tool for use within company X in terms of usability and performance. The main style of approach to be taken by this project will follow the goal of "Which one is better", outlined in the literature review, as it aims to review the selection of tools and identify the most suited/ best in terms of HCI and usability for company X. Therefore this method requires an opinion based approach as identified within the literature review. This will take place as a case study using the employees of company X who are working with the software in question on a day to day basis thus are qualified to analyse and evaluate. This will ensure that quality feedback and responses are obtained which relate to the case study company.

The HCI evaluation to be developed and carried out follows the QUIS method as the literature research has shown that this approach would best suit the evaluation of various tools by expert users, given that the tool is already developed and in use. This HCI evaluation is not aiming to suggest possible improvements to the software which is in use, it is therefore aiming to identify the most effective solution to company X via assessing the tool by their users. This QUIS (Questionnaire for User Interaction Satisfaction) is based upon the version create by IBM and Maryland University the aim is to evaluate the user interface and experience they have when interacting with the tools. The QUIS uses a rating system for each question this rating being between 1 and 10. The user answers a series of questions relevant to the tools and their use within the company. These responses then allow for a general rating for each category of the QUIS which in turn allows for the tool to be assessed for user satisfaction and

compare what tool is the preferred option for the job. It may also help company X identify possible problems with the implementation of the tools in use.

The author believes that this approach is the best to formulate a possible solution to the following hypothesis:

- The best suited tool of choice will have a user interface which the staff believes to be the best quality in terms of meeting their wishes as a tool for batch scheduling.

The QUIS questionnaire was produced using both the IBM example developed by James R. Lewis. (1993) and the QUIS of Maryland University (QUIS, 2008). The questions have been identified by the author via review of the two existing examples, tailored to meet the needs of the tools within company X and then vetted and verified as suitable and relevant by a manager within company X's batch scheduling team. Thus the QUIS used here has 6 main sections outlined below:

- Introduction
- User Reactions
- Screen Layout
- Terminology & Information
- Learning
- And System Capabilities

Each of the categories above have been identified as key areas of importance based up those which appear in the QUIS examples analysed previous. Both the author and another expert of the tools in question identified that these areas were of most relevance to the evaluation of the tools. They were identified via an overview of the example QUIS's. Each section has been tailored via phrasing and questioning the user in a relevant and useful manner.

The sections which have been deemed irrelevant to the evaluation of the software and current within company X, due to their nature, have been outlined below:

- On-line tutorials - Not seen as relevant due to company X having specific naming standards and development processes which must be followed. This therefore makes most online help irrelevant and not worth assessing.
- Multimedia - The software in question does not have any valuable or testable multimedia that would benefit from an evaluation
- Teleconferencing - This again is not used within the software in question thus is not relevant
- And finally software installation- This is not relevant as the infrastructure and system are already in place within company X which is controlled in a strict manner. Thus there is no purpose to evaluate its performance.

The purpose and details of the 6 sections of the developed QUIS are outlined below:

The introduction section has been included to obtain an overview of the experience and nature of the user in terms of previous tools used, number of years used user's sex and age. These will be used to analyse the majority users of each tool across company X and are included in most questionnaires to obtain the general details of the users whom are answering the questionnaire. It is also used to verify the experience of the users to ensure that they are classed as expert users of the system thus validate the feedback of the users. It will therefore ensure quality of the feedback to be assured by knowing the users experience of the tools and the subject of batch scheduling.

The second section named user reactions is aimed at gaining the general user response to their opinion on the tool in general. This section includes general questions about the tool which the questionnaire is referring too. Questions such as "What is your general opinion of the system overall?" with a response as a rating of 1-10 between awful and excellent. These questions have all been selected to gain a general overview of the user and how they interact with the tool in terms of their satisfaction. More specific aspects of the software are covered within the other categories but this section covers the user's opinion of working with the tool, its ease of use and power of its capabilities. This is to obtain a general feel for the tool and is questions which are included in the QUIS of Maryland University.

The screen layout section has taken a majority of the questions from the QUIS of Maryland University with tailoring to ensure they are relevant to company X and the tools in question. These questions are based around the information and layouts of the tools menus and the movement between the different areas of the tool. Questions such as "How do you rate the amount of information that is/can be displayed on screen?" are included which are answerable via scale of 1-10 Inadequate to Adequate. The questions are tailored to ensure they are relevant to the tools as the QUIS of Maryland University questions were covering aspects which were out of the scope of the software in question. Questions regarding the Image of characters on screen, Character shapes (fonts) used and Highlighting on the screen are all those which are excluded as they would not give useable or measurable data which would improve the results or research. Therefore only the questions which the author and company X managers believe would be most suited to analyse after collection.

The terminology and information section of the authors QUIS was developed to collect data regarding the information and terminology used throughout the system which in turn shows how much training and teaching is required to understand and work with the system. The amount of batch scheduling specific terms used and the information displayed helps determine how difficult a tool is to use thus have been included to obtain how difficult the tool is to work with on a day to day basis ensuring it is logical and relative to the tasks the user is performing. This was included as batch scheduling tools often rely on knowledge from the user and their ability to understand what each option or feature will do when interacting and using the software, this therefore evaluates the ability for users to understand and make sense of the tool based on the instructions and terminology used.

Learning has been included in the QUIS as it allows for each tool to be analysed and asses the learning curve for the average user. This allows for company X and also this

research to evaluate the users learning experience of the software which shows how quickly users can work and effectively set up schedules using the software. This allows for decisions to be taken on the cost of implementation and the speed of the bedding in period where by users settle to the new tool. The costs of training and timeline for the software to become effective can be evaluated using this section of the QUIS thus company X believed these questions, tailored from the IBM example, were suitable to the research.

System Capabilities is the final section of the author designed QUIS and is aimed at specifically targeting the power and ability of the tools. It allows the tool to be questioned for reliability, response time and error correcting. Questions like this allow for the user to show how much they trust and rely on the system which plays an important role within company X as they believe users must believe in a tool and have confidence in it otherwise they will not adopt it as a their preferred choice. This therefore is an important area of the QUIS and has based its questions on that from the QUIS of Maryland University.

The ISO 9241 outlined again below is closely linked with these chose areas and has been used to help ensure the sections which have been developed for the QUIS are all valid. Each section of the QUIS can be linked directly to the diagram below. One example being part 12- "presentation of information" shown within the diagram below linking to section 3 of the QUIS which evaluates the "terminology and information displayed on screen". This ISO diagram below was used when developing the questions along side the QUIS examples to ensure the appropriate sections of the software were being categorised and evaluated correctly. Testing the relative areas appropriate questioning.

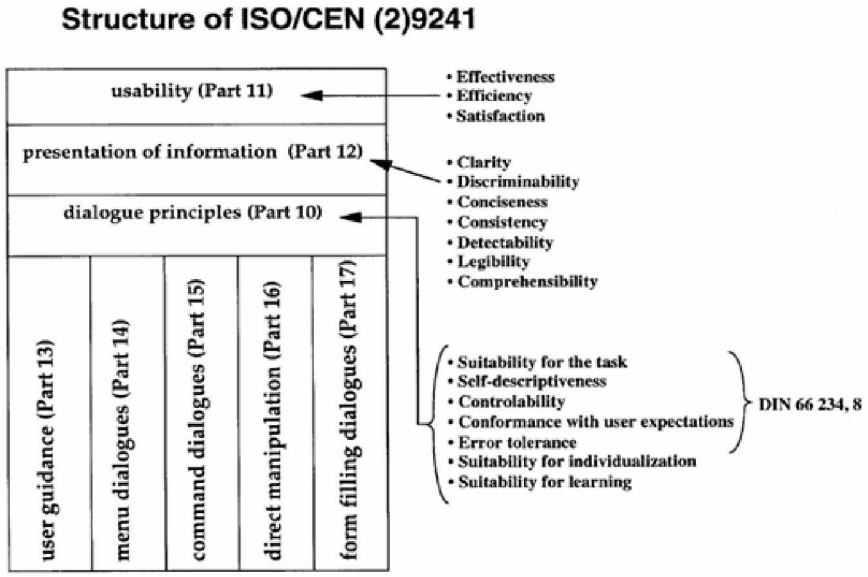


Figure1. Structure of the multi-party standard ISO 9241 (found in the work of R. Oppermanna and H. Reitererb, 1997)

This developed QUIS for the research was distributed using www.surverymonkey.com which is an online survey tool which allows for the creation, distribution and collection of results to large user groups. The QUIS will be developed and created for each of the 3 tools being assessed. This will then be sent to

all 3 user groups via 3 separate web links to the 3 QUIS's. The main contact within company X will be responsible for sending these links to the appropriate user groups to obtain responses. These results can then be analysed and downloaded to allow for the author to present and conclude the findings. The results section of the report outlines the responses and user group sizes for each tool in question.

The full QUIS which was distributed can be viewed within the appendix A detailing all questions the users were faced with.

4 Results

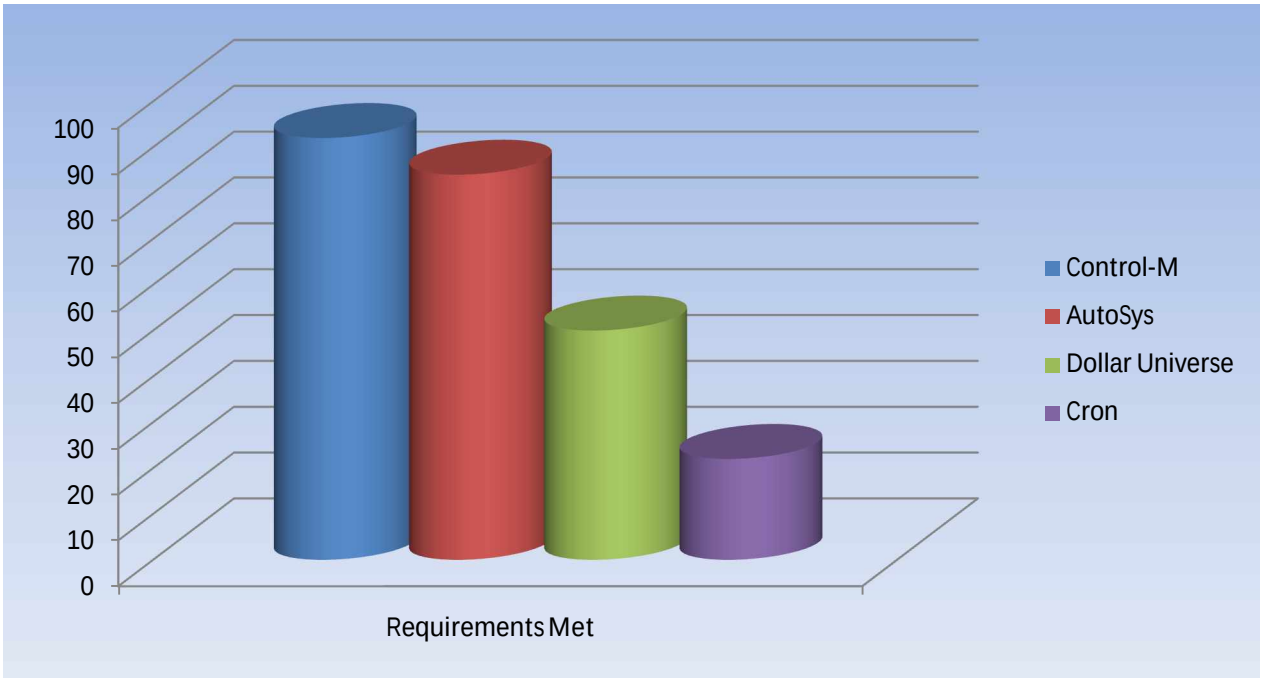
Section 4, the results section, will highlight the findings of both the requirements matching and the HCI QUIS evaluation which has been conducted following the above methods. This section will highlight the key differences between basic and advanced batch scheduling tools and also suggest that the difference between Control-M and AutoSys is not great enough to force an immediate change.

Full results of evaluations are available to view within the Appendix A of the report.

4.1 Requirements matching

Appendix A shows the requirement matches for each tool. These have been matched through close analysis of each tools features via the manuals and websites of each tool to identify if each tool can individually meet each requirement; a slow and time consuming task which has lead to an in depth coverage of each tool. These results have been checked and evaluated using a scoring system which allows for percentages to be calculated for each tool. These percentages are the number of requirements which the tools meet per section or otherwise stated.

4.1.1 Overall Requirements



As can be seen from the above chart out of the 91 requirements there is a clear difference between Control-M and AutoSys compared to Dollar Universe and Cron. The graph above shows the percentage of requirements each tool meets overall from the 91 in total. The figures below show the exact number of requirements each tool meets of the 91:

Control-M 84/91 or 92%
AutoSys 77/91 or 84%

Dollar Universe ($U) 46/91 or 50%
and Cron 20/91 or 22%

These overall requirements suggest the overall category which the tool will fall. As it has already been highlighted there are 3 categories of batch scheduling, these being Native, Basic and Advanced. The above graph would be in keeping with this trend showing that Control-M and AutoSys fall under this advanced category which, through analysis of company X's requirements, would appear to be the style of batch scheduling tool required. However Dollar Universe does attempt to keep pace with these advanced tools, it falling under the basic category, with a score of 50% of all requirements being met. This would suggest that company X could implement such a tool however advanced features may not be easily implemented thus perhaps not the best option for the needs of the company. Cron is highlighted as the native tool with a score of 22% suggesting that its use should be restricted to those applications which do not require any true batch scheduling features, only the very basic of applications or jobs should be implemented with this tool where an advanced tool is not required.

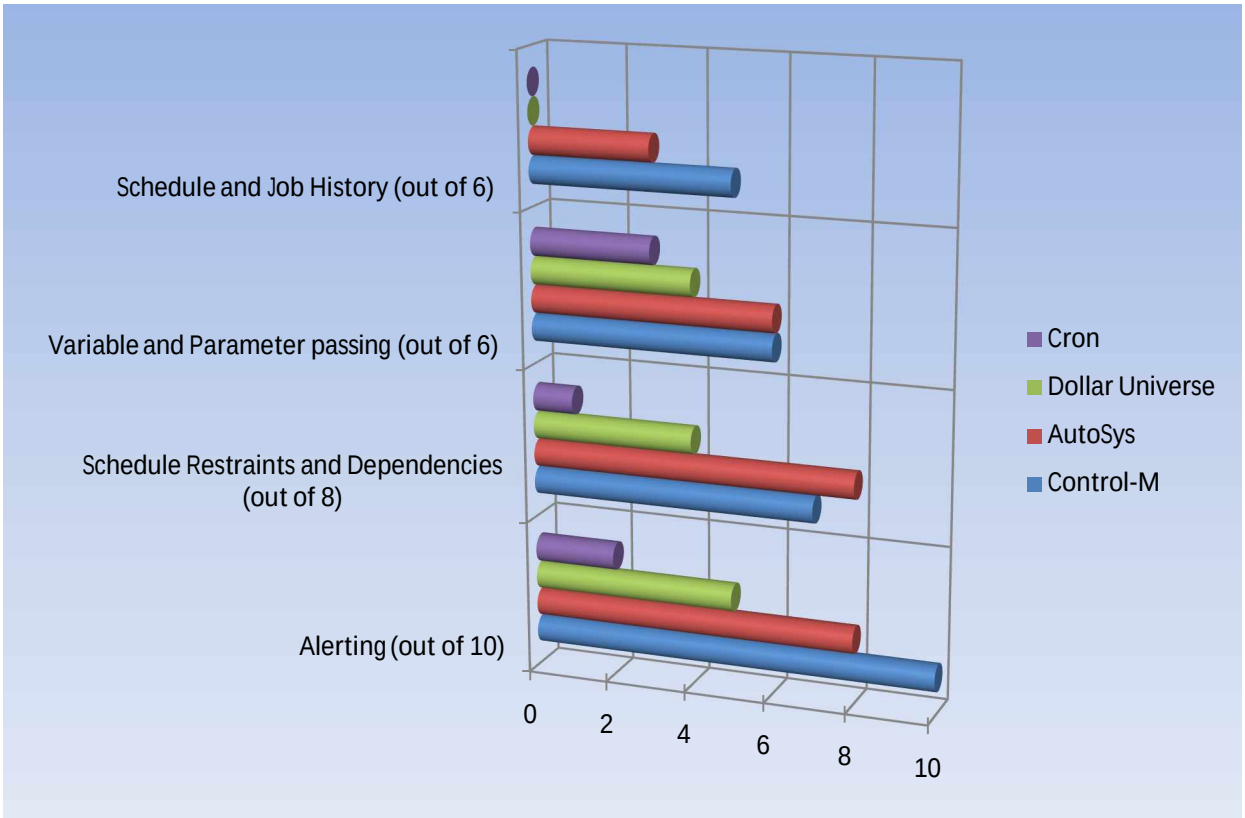
However this overall review of requirements does show that there is potentially little difference to separate Control-M and AutoSys. With both tools falling under the advanced category and only separated by 7 requirements in favour of Control-M there is no real evidence to choose one tool over another. The results only slightly fall more towards Control-M however this may not be enough of a difference to disregard or migrate away from AutoSys completely.

These figures can be broken down further to highlight the areas individually under each of the 10 categories identified for the requirements during the literature review. These 10 categories being:

- Alerting
- Event Scheduling and Control
- General Software Requirements regarding batch scheduling
- Job Execution and Recovery
- Jobs and Schedules
- Schedule Restraints and Dependencies
- Schedule and Job History
- Schedule Capabilities
- User Interface
- Variable and Parameter passing

4.1.2 Priority Requirements

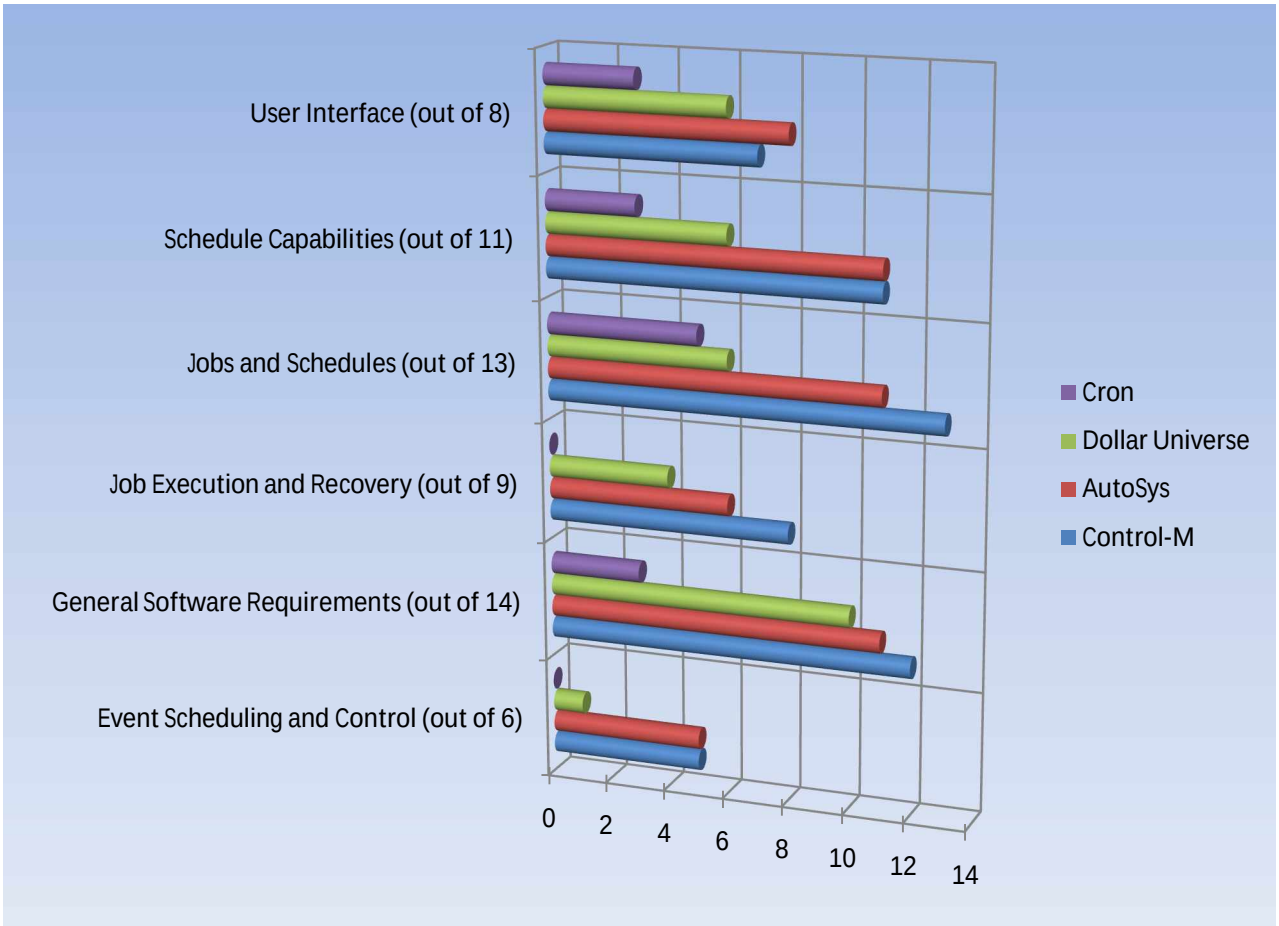
Of these 10 categories 4 were identified as priority or superior areas which the tools must aim to excel at in order to match the needs of company X. The 4 categories which have been identified by company X have first been evaluated and the results shown below:



A full overview of the results and each specific requirement that each tool meets can be found in Appendix A where the full review is covered. But from the summary results which can be seen above it is clear to see that Control-M currently covers the most requirements which are important to company X. From the results above Control-M covers 28/30 of the requirements for these 4 categories which is closely followed by AutoSys which has 25/30 requirements matched. Dollar Universe and Cron do not come close to matching the needs of company X at this point and perhaps this is more to do with the fact that Cron and Dollar universe are both Native and Basic batch scheduling tools and are not included in the Advanced category which it now looks like company X will require for their batch scheduling solution. AutoSys is not far off Control-M in terms of the number of requirements which are met across these 4 key categories and therefore again there is no real evidence to disregard AutoSys in favour of Control-M with such little difference between the two.

4.1.3 Remaining Requirements

The remaining 6 categories, which are important to any batch scheduling tool, must be taken into account when conducting this review have been analysed below. These areas are still important to batch scheduling tools and must be incorporated throughout any tool in order for it to be effective; however they were not noted highly in terms of priority from company X and have been analysed separate from that of the 4 priority categories.



As you can see from the summary of the remaining categories there are 2 clear leaders in terms of the requirements section of this software evaluation. Both Control-M and AutoSys are clearly the more comprehensive batch scheduling solutions. They both cover over 80% of the requirements by some distance which shows they are clearly advanced scheduling solutions. The vast gap between Control-M and Dollar Universe and the other tools is more due to the fact both these tools are advanced batch scheduling solutions thus cover the basic and more advanced features of batch scheduling. Dollar universe seems to be a more basic solution which company X may not benefit from making their primary scheduling tool. Cron however is clearly a native tool as it does not offer a comprehensive or effective method of batch scheduling on a large scale. There is however yet again little or indeed no difference between Control-M and AutoSys across these remaining 6 categories. This would indicate that both are effective solutions to batch scheduling and can be implemented to some effect in both cases. Further evaluation therefore is required to differentiate between these two tools if possible.

The full results of the requirements matching are shown in appendix A along side a complete run down and explanation of each individual requirement. This requirements list can be used for any batch scheduling tool to evaluate its general performance in terms of features which it can perform. However this is only one part of an evaluation. The implementation of these features can only be fully assessed along side an expert opinion of the software and how it performs in a real working environment and in production.

The following objective has been reached upon completion of this section of the research. This objective was initially highlighted at the beginning of the report and has been an essential part of the project itself:

- Refine more detailed batch scheduling requirements for businesses such as company X this will be done via a case study of company X and their needs with regards to batch scheduling. This may include interviews with those within the company. This will help to focus the evaluation of the software by identifying the current tools in use and the situation regarding the batch scheduling within the company. The research can then be tailored to try and best cover the needs of the company and offer valid and useful conclusions. This will produce a developed list of requirements which a good batch scheduling tool should try and match. This will be done via collating the results of the literature review and initial case study of company X. This will allow for in depth requirements to be set out for comparison with each tool. This checklist will then be used as a valid evaluation technique via assessing each tool and the requirements which it can or cannot meet. This will help to prove the effect the tool would or does have within company X based on the features and power it has available.

This requirements matching section which is outlined as an objective has been undertaken to help the overall research question and test the hypothesis which were set out by the author. The author believes this objective has been particularly effective at testing the following hypothesis:

- An appropriate tool will match all or most of the requirements of company X based upon a review of each tool against a developed set of requirements identified via the literature into basic batch scheduling features a tool should offer.

The hypothesis claims one tool will be a clear winner in terms of requirements for company X. This hypothesis has been tested via the requirements matching exercise which has identified that Control-M is the leader matching the greatest amount of requirements, from the developed list, closely followed by that of AutoSys. This has been tested and the results shown indicate that Control-M is the most effective tool as it meets the most requirements however; AutoSys is not that far behind that it can be disregarded as a suitable tool of choice. However, Dollar Universe and Cron are not appropriate tools for the needs of company X and the results clearly reflect this.

4.2 HCI QUIS

This QUIS questionnaire has been developed and distributed across company X for completion. This distribution took place via using www.surveymokey.com premium version, this allows the user to create online surveys or questionnaires and collect data across all the users who reply.

The full questionnaire based on the IBM and QUIS of Maryland University can be seen in full in appendix B. This lists the 3 QUIS's for each tool as distributed across company X. The results have been collected via survey monkey (www.surveymonkey.com) which is a website which allows for surveys to be created online and results collected which can then be analysed. This website tool allowed for the author to distribute the QUIS across company X on a global basis throughout the various departments which used batch scheduling tools on a regular basis without having to post, meet or communicate in a time consuming one to one manner. This was done via a contact within the batch scheduling team of company X distributing the web links to the appropriate user groups. These users were then able to complete the QUIS at their own free will during the month of March. This allowed for the subject users to respond at their leisure which potentially increased the response ratio. As of the end of March after the distribution of the QUIS for all 3 tools the responses stood at 17 for Control-M, 8 for AutoSys and 12 for Dollar Universe. These figures are relative to that of the user numbers of each tool. The user estimates based on conversations with company X management that Control-M has between 500-600 everyday users of the tool. Therefore around 3% of the population of users managed a response (based on the total average number of Control-M users being 550). Dollar Universe user numbers are estimated to be in the region of 120 everyday users thus a total of 10% of the user population responded. And finally AutoSys, which is the tool of choice within the American section of company X, has an estimated number of users around 300-400 managed a response of 2% (based on the total average number of users being 450). These figures therefore can fairly reflect a view of the tools across the population of company X given that Laura Faulkner (2003) findings of her research show that between 5-15 users is sufficient when assessing the usability of a system or tool and that increasing this figure beyond 15 doesn't specifically improve the findings of the evaluation.

The QUIS research section has disregarded CRON and does not evaluate it like that of the other 3 tools regarding the QUIS evaluation. The reason for this being CRON is a very simple and native scheduling tool. As the requirements analysis section of the evaluation has shown the tool does not have anywhere near the number of features which company X would and do require. Therefore the author feels performing a QUIS evaluation on the tool would simply be a waste of resources both to the author and company X especially as CRON offers no real interface to assess. Any outcomes of such a QUIS would not be relevant as the tool will not be viewed as the overall solution to the large scale batch scheduling, instead this native solution will simply become the last choice of tool to company X when no other scheduler can be used or is not needed.

As previously outlined within the literature review section there are 6 parts to this HCI QUIS shown below:

- Introduction
- User Reactions
- Screen Layout
- Terminology & Information
- Learning
- And System Capabilities

4.2.1 Introduction

The introduction section has been included to validate that the users are all experienced to a level which puts them under the expert category which is required for this for of evaluation. The results of this section can be seen below:

How many years experience have you had with batch scheduling tools? (average)	
Control-M	5
AutoSys	11
Dollar Universe	6
Overall Average	**7.3**

As you can see above each user group has responded to the question giving an average of 7.3 years of experience with batch scheduling tools across the sample of 36 users in total. This therefore gives a clear indication that the users are highly experienced enough using scheduling tools to give valid responses and feedback regarding using such tools.

The users were also asked how long have you worked with the tool in question? And the summary of responses can be seen below:

How long have you worked with the tool in question? (average)	
Control-M	Less than 1 year
AutoSys	5+ years
Dollar Universe	1-2 years

It can be seen that the summary indicates AutoSys and Dollar universe users are well established using the tool in question for a number of years. However Control-M does fall short of general users being experienced in using the software across the sample of 17 user responses. Control-M however is a new tool to the users within company X thus this result was expected. This is however still valid feedback considering the users previously answered that they have an average of 5 years experience using batch scheduling tools in general, this in conjunction with the fact that Control-M has the most responses from all 3 tools helps prove the feedback is valid and typical.

To further validate these responses the users were asked on average, how much time do you spend per week using this tool? The replies fell under 4 categories ranging from 1-10 hours through to 30+ hours and the average response is shown below:

Number of hours a week working with the tool in question: (average)	
Control-M	1-10
AutoSys	30+
Dollar Universe	1-10

The responses here again suggest that AutoSys users are by far the most experienced users which when considering the tool generated the least amount of feedback further helps to justify that the responses are strong typical results which are from highly experienced users using the tool on a regular basis with a number of years experience. Dollar Universe and Control-M users are on par using the system less that AutoSys however still having experience and use which validates the responses for the remaining of the QUIS.

4.2.2 Overall Satisfaction

The above chart overall satisfaction percentage for the users using each individual tool. This percentage shows the tools in comparison to each other and has been calculated via working out the percentage each tool meets with regards to the QUIS questions:

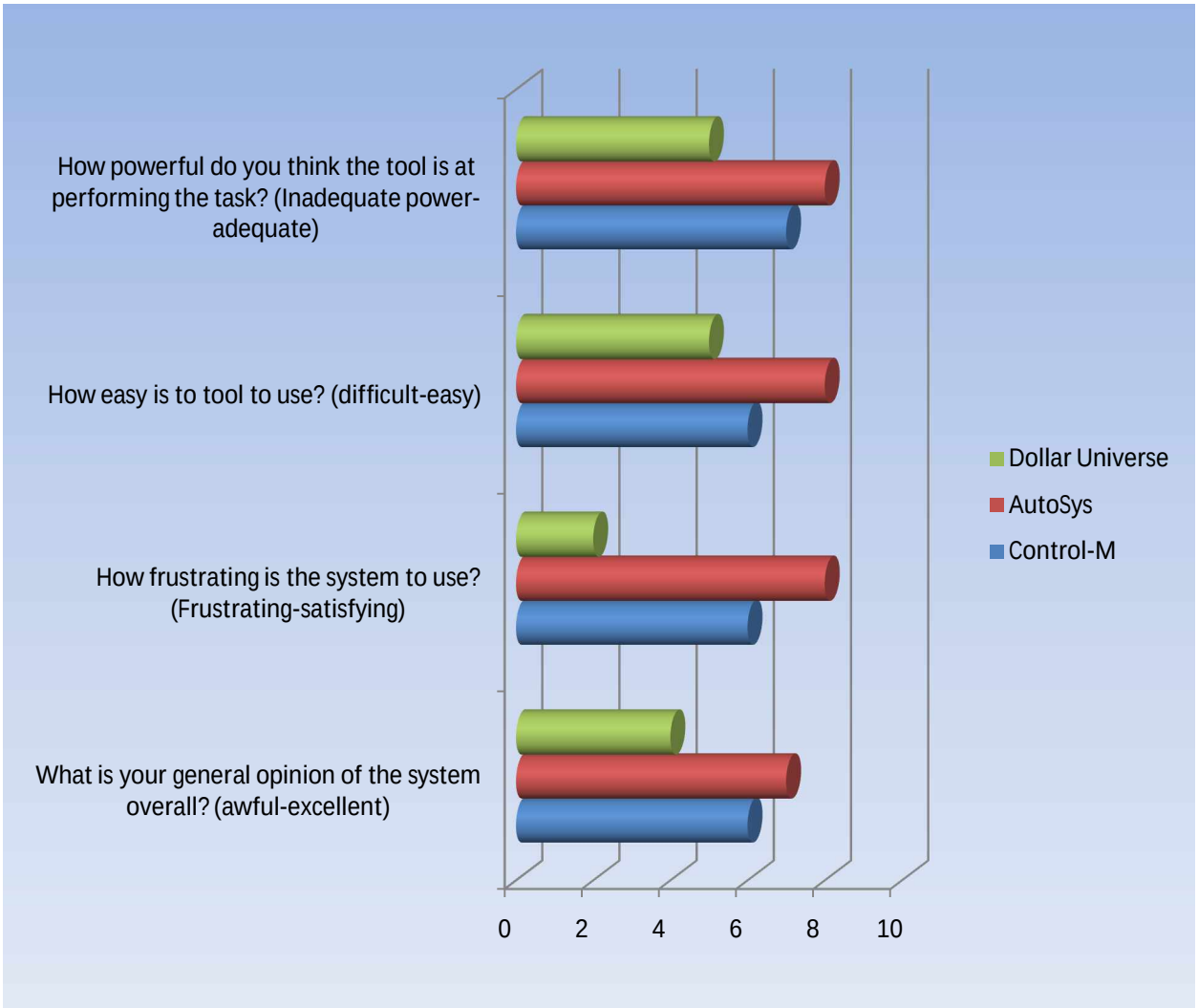
Tool	Satisfaction score out of 350	Percentage out of 350	Percentage compared other tools (out of 184%)
AutoSys	248	71%	39%
Control-M	225	64%	35%
Dollar Universe	172	49%	27%

As you can see from the above chart and table there is little difference between AutoSys, the leader on 71% average satisfaction, and Control-M with an overall satisfaction of 64%. However it is clear that Dollar Universe lacks something which these tools have it only scoring 49% satisfaction thus clearly not the tool of choice to the users in question. These figures indicate that from the expert user group thus sample of users from within company X they believe Dollar Universe to be the least effective tool to perform the batch scheduling tasks they desire. This would suggest that the tool of choice, as indicated previously in the requirements matching evaluation, is between Control-M and AutoSys.

The further breakdown of each section of the QUIS can be seen on the following pages with analysis.

4.2.3 User Reactions

The user reactions section below outlines the user's general views of the tools individually with regards its ability and their experience when interacting with the system.



The above results show the summary and average responses for the user reaction section of the QUIS. The questions are outlined along the vertical axis and the response from 1-10 across the horizontal axis.

As you can see the above graph is in keeping with the overall satisfaction of the users with each tool in that Dollar Universe is clearly falling behind the others. The users find that Dollar Universe is the most frustrating tool to use thus obviously has key problems when the users are interacting to perform the tasks they require. This frustration, something which Dollar Universe causes, may have effects on how the users reply or interact with the system, thus ultimately, prove to be the downfall of the system. As with many users and products if the product is frustrating to use then the users will be reluctant to use and persevere with it to perform their tasks. A system has to make the user's job easier and not complicate it further via frustrating them to the point where the job becomes more of a chore. Therefore this QUIS will identify

the areas which frustrate the users of Dollar Universe the most. Control-M and AutoSys are only closely separated when it comes to the satisfaction of the users. This close gap could be linked to the user experience of each tool, the fact that AutoSys user group has more time working with the tool collectively compared to that of Control-M. As you can see both groups however do believe that Control-M and AutoSys are both powerful tools which are both capable of performing to the needs of the company and users which is backed up with the previous results of the requirements matching exercise.

Further to this user feedback regarding to the user reactions section of the QUIS shown below highlight some of the issues between the tools:

“$U has a poorly implemented GUI which severely hampers its usability. The $u chosen paradigm is flawed in places, resulting in the need to create "hacks" to work around shortcomings in the product.”

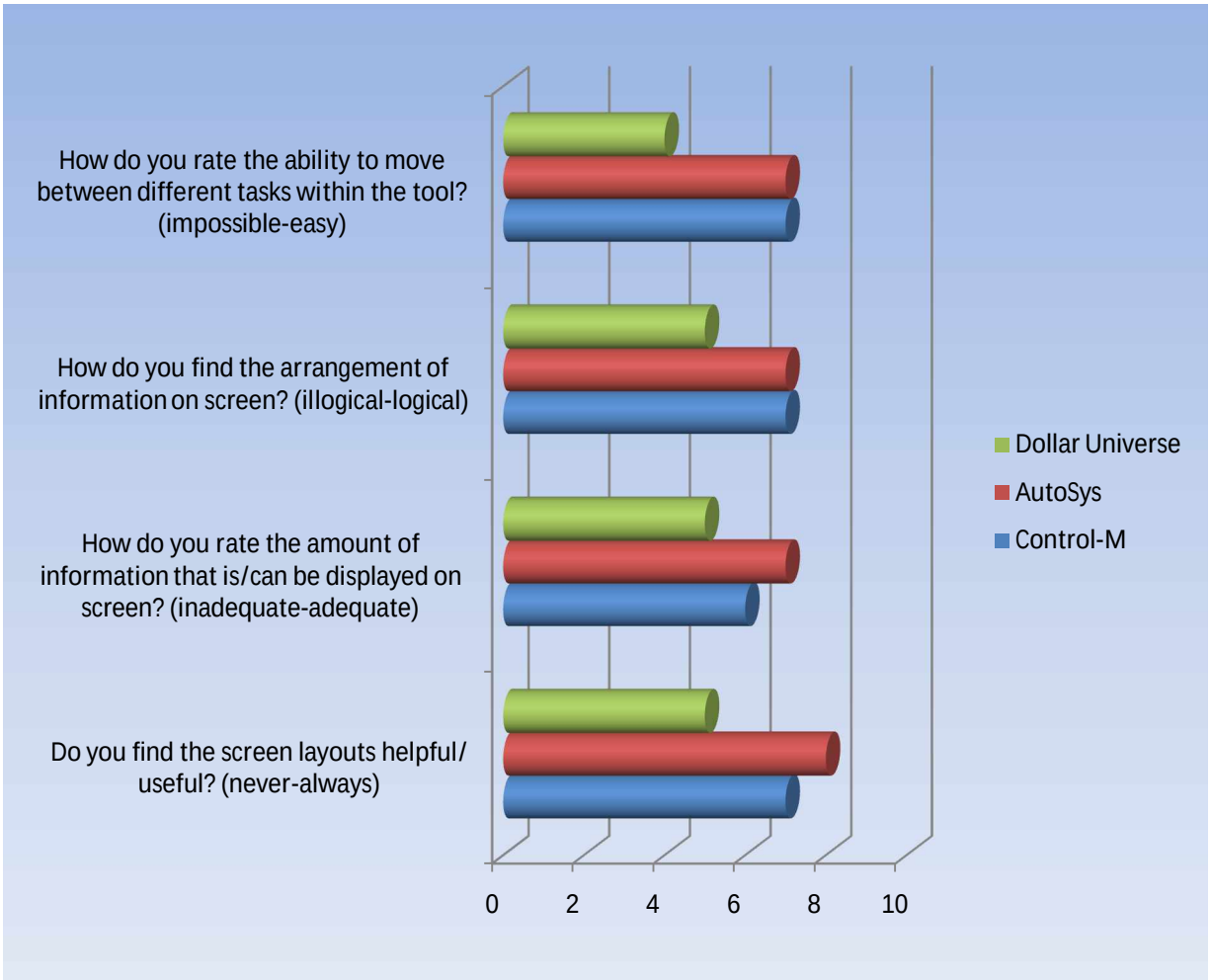
This above user states that often the user must work out “hacks” to make the tool do what the user requires due to the poor implementation of the GUI. This therefore adds to the frustration and generally poor experience most users seem to have with the system.

The same comment section from the control-M user group below clearly shows the benefits of a slick GUI interaction system.

“From experience, CtM enables a developer to create schedules in a very short time period, extremely important in COMPANY X’S fast-moving environment. Also, the diagrammatic layout detailing the dependencies in a 'waterfall' form provides the tool with a user-friendly interface and a quick tool of reference. CtM is fit-for-purpose.”

4.2.4 Screen Layout

The screen layout section of the QUIS analyses the user's expert opinion and views on the layout and detail of the tools interface they are interacting with on screen. Thus allowing for the tools impact and impressions to be rated in terms of how useable the tool is via its layout.



The above results show the summary and average responses for the screen layout section of the QUIS. The questions are outlined along the vertical axis and the response from 1-10 across the horizontal axis.

The above results of the screen layout section of the QUIS shows that again Control-M and AutoSys are closely linked in terms of information and layout on the screen. Dollar Universe again falls behind these two tools as so with its performance in the requirements matching. This clearly shows the preference for a visual and graphical user interface with which the users can interact. As stated previously within the report Dollar Universe offers a command line based interaction with the system and can often be harder for the users to perform tasks due to its very nature. On the other Control-M and AutoSys versions, which were evaluated, offered the users a very in depth Graphical User Interface (GUI) which are believed to be preferred by many users these days as it offers faster interaction methods which the user can interact with better due to the visual nature of them. This QUIS backs up the belief that users do prefer the GUI based system over that of the command line interaction style. This

therefore, may be another reason for the extra frustration of the Dollar Universe users. They do not have these GUI style interactions with the system to aid their developments and tasks within this. The new version of Dollar Universe would have to be analysed in comparison to that of the Control-M versions and AutoSys versions used within company X to identify improvements in its performance of the QUIS evaluation. However, as this version is not implemented currently there is no need or reason to evaluate it for company X as part of this case study.

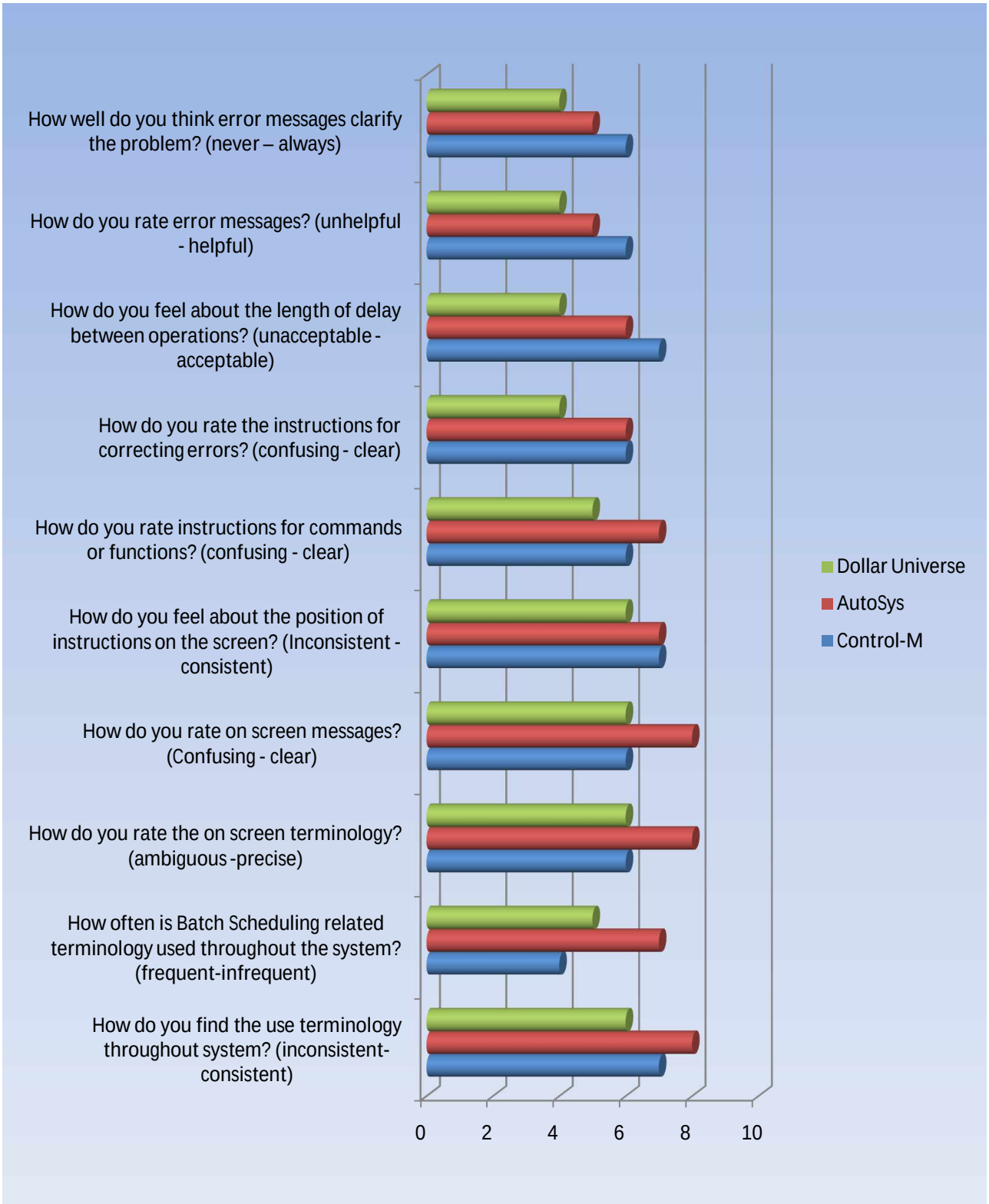
One comment which backs up the problems found using Dollar Universe with its designed screen layout is outlined below from one user:

“The interface has not been designed with any consideration of usability. The approach has clearly been to drop data on screen hap-hazardly.”

Clearly this user feels that the system feels like a database feed of the jobs and executions rather than a tool for controlling the flow of jobs, data and processes. This again relates to the lack of a GUI in place for the user to navigate and interact in a more meaningful manner.

4.2.5 Terminology & Information

This section is used to analyse the user's reactions to the systems language and feedback when instructing the user through their daily operations and tasks. The feedback helps to provide analysis with regards to the ability to learn while using the tool and also for users to be assisted with the terminology & information which the system offers them.



The above results show the summary and average responses for the terminology and information section of the QUIS. The questions are outlined along the vertical axis and the response from 1-10 across the horizontal axis.

The terminology and information section of the QUIS is interestingly the only section where Control-M almost outshines AutoSys across the section. However AutoSys does regain its lead as the questions continue but Control-M is even seen to be on par with AutoSys across some of the questions regarding terminology and information. The results also highlight again Dollar Universe falling behind the others. This section does therefore highlight the importance for the need for good terminology and on screen information in a system to aid the user in their tasks. This section shows both Control-M and AutoSys both clearly deal with the terminology and information better than that of Dollar Universe. This is possibly due to the fact that Control-M and AutoSys have GUI style interfaces opposed to the command line system of Dollar Universe. The nature of this GUI style offers the developers the chance to illustrate and communicate with the users in a more interactive manner which could help convey this information. The use of a proper GUI allows for the arrangement of information and terminology to be placed at appropriate sections on the screen during the relevant interactions to help the user navigate and understand their actions on screen better than that of a command line system. This is just the nature of command line and GUI interfaces and one of the reasons why many users prefer the GUI over that of the command line.

Although AutoSys user did not provide any real commentary to any of the free text which followed each section of the QUIS control-M and Dollar Universe did. With regards to the terminology and information a common problem the Dollar Universe users found is highlighted below:

“Some error messages are in French. Some error messages give a rough idea of what the problem is, but don't really offer any solutions”

The system seems to have issues with displaying the errors in French with no real help in solving the problem. Something which may be related to poor development processes from the vendor which has caused inconsistency throughout.

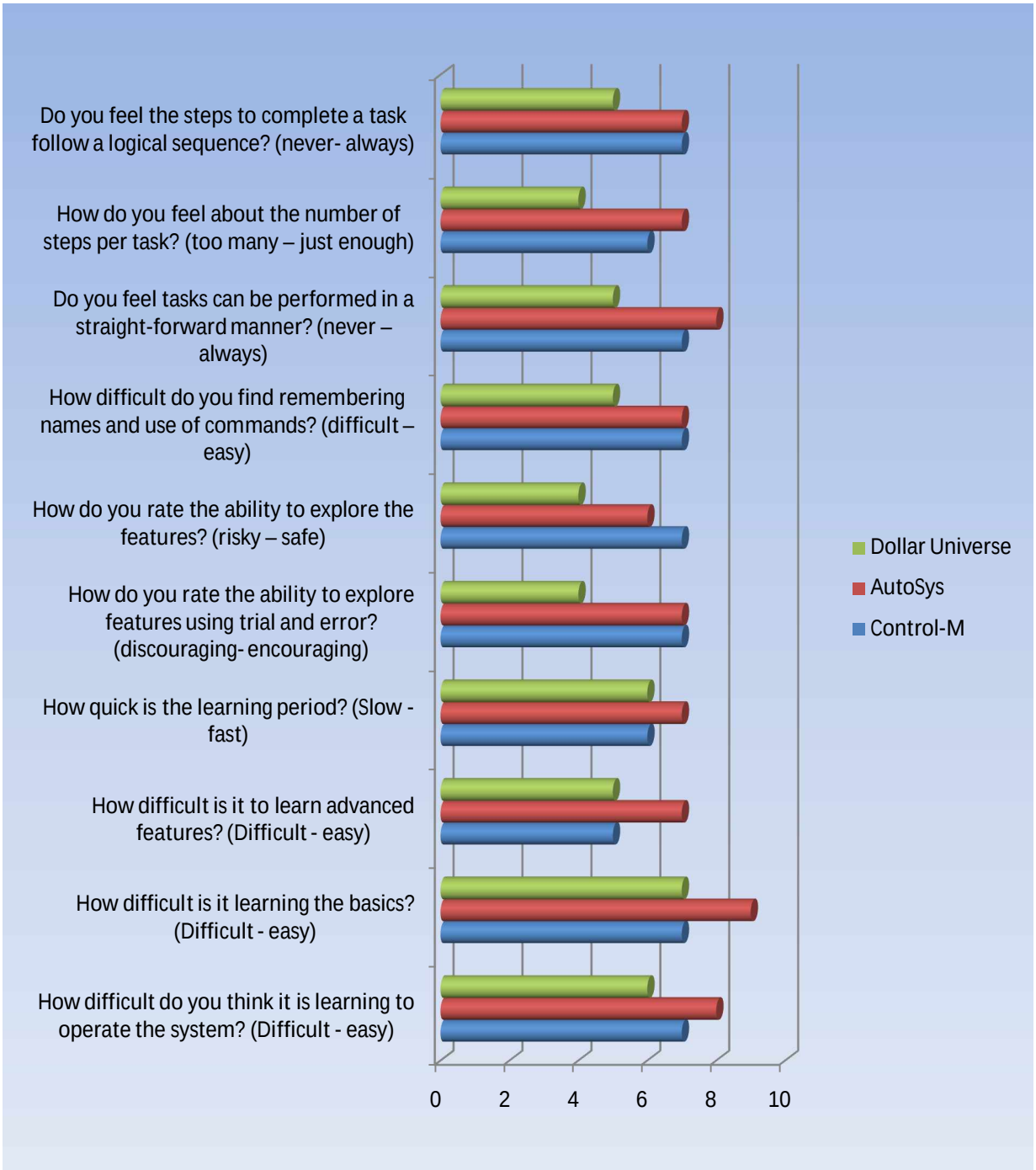
Control-M however does seem to have its problems also:

“Database errors are not passed through to the GUI, which means that frequently the error messages simply tell that something did not work, but there are no references to why. This makes extending the tool properly quite hard.”

This issue again could be related to poor development or quality from the vendor to ensure errors such as this both don’t make it through to the final system or if they do then they are handled correctly.

4.2.6 Learning

The learning when interacting and using any tool or software is vital to its success or failure. Therefore the QUIS section gathers data on how easy each tool is to learn by assessing a number of areas shown through the questions below with analysis.



The above results show the summary and average responses for the learning section of the QUIS. The questions are outlined along the vertical axis and the response from 1-10 across the horizontal axis.

Satisfaction when it comes to the user learning experience is important. If a user finds it hard to start using a system then the less likely they are to adopt and use it as the tool of choice. The harder a system is to use and learn then the more reluctant user groups will be to recommend and suggest the use of the tool throughout the company. Thus the easier a system is to pick up and use, the more chance of the system being used and widely adopted. The same trend is highlighted where by Control-M and AutoSys are ahead of Dollar Universe which falls behind vastly as in previous sections. The data however between AutoSys and Control-M may not be as accurate as it could be. Control-M is relatively new thus the users's, having only used the tool for less than a year, are fresh with the learning process and are in fact still learning the system. AutoSys users however are far more experienced in terms of using the tool and have perhaps forgotten how difficult the learning process is as they, on average, have been using the tool for 5 years or more. This difference in time between the learning experiences of each group of users may suggest the results are not as accurate as they could be. Ideally the users would both have had the same level of training and experience with the tools to identify the tool which is best to introduce and pick up for new users. This said the gap between Control-M and AutoSys is not that great that one could be seen as vastly easier to learn.

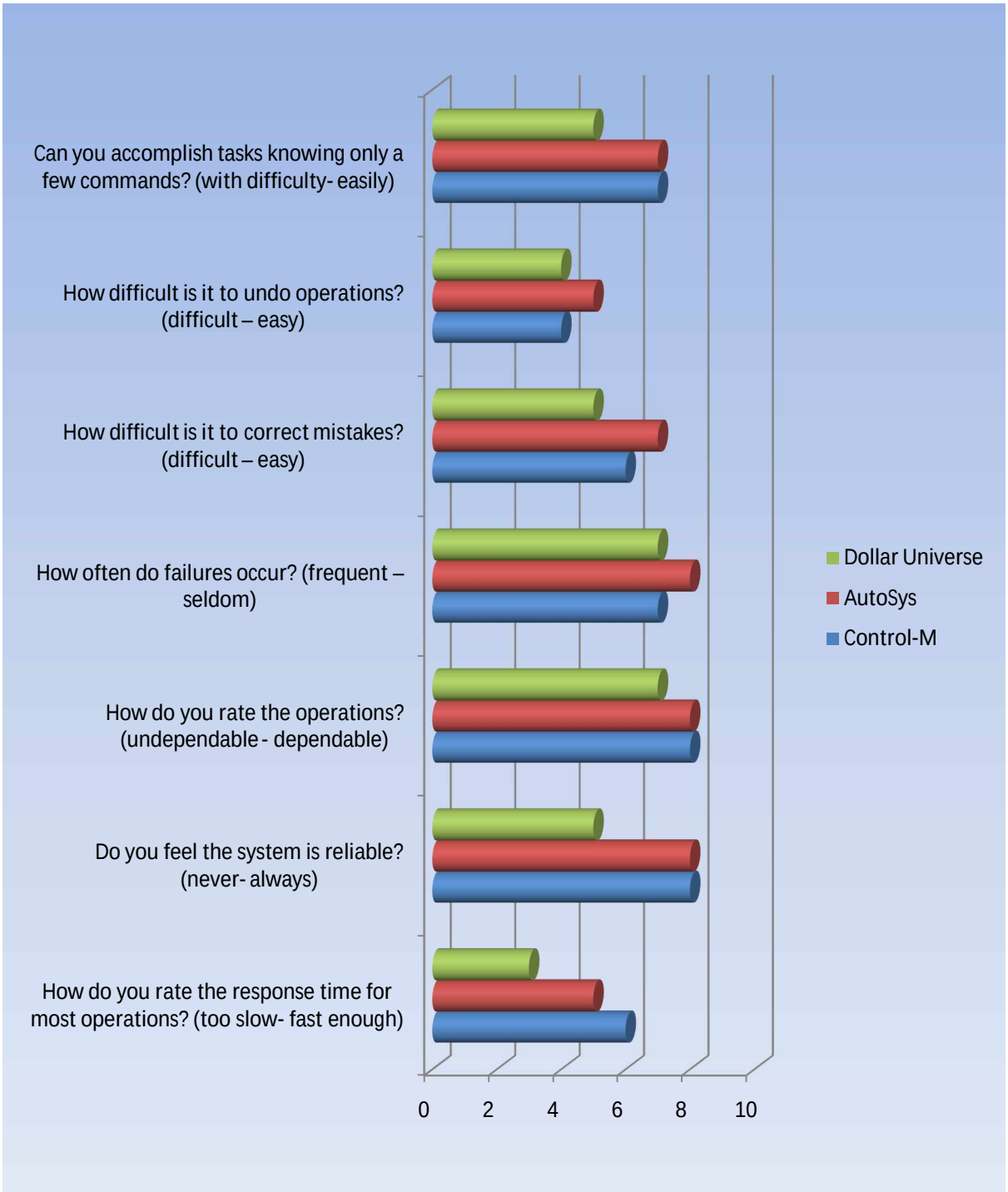
Summing up the learning process of Control-M one user commented:

"Tool is easy to pick up, but takes a while to master all the more advanced functions"

This helps emphasise the tools simplicity to learn but difficulty to master. This however is something that given time the users will be able to do. Given that the tool is relatively new this commentary is generally expected from the lack of experience with control-M.

4.2.7 System Capabilities

The final section and results to the QUIS summarises the systems performance in the eyes of the user. Mainly the power, reliability, and how easily operations are changed are covered here to obtain the users views of the systems ability.



The above results show the summary and average responses for the system capabilities section of the QUIS. The questions are outlined along the vertical axis and the response from 1-10 across the horizontal axis.

The system capabilities section is used to highlight the satisfaction of the tool in terms of its capabilities and performance. These results have Control-M and AutoSys on par with each other with Dollar Universe again falling behind. The performance is an important factor to company X which the users of Control-M and AutoSys clearly feel has the adequate performance benchmarks to meet the needs of the company. The tools are trusted to be reliable without failures thus the users believe and are willing to use the systems to perform the tasks in hand. This trust is key to the managers and users of company X fully adopting a system to handle its batch scheduling needs. Obviously the users feel Dollar Universe does not have this reliability or robustness which the users want when handling these important batch scheduling tasks. This therefore is something which the users feel adds to the satisfaction and overall user experience when interacting with such systems.

Some general responses to the tools in question when asked "Please enter any feedback or additional comments here:" are shown below which help to prove the findings of the QUIS which the author has reached.

One user replied that "$U is a stable tool, but the environment in which I use it is an extremely demanding production environment, and therefore it falls short in regard to configuring complex dependencies and the interface is often slow to respond". This clearly shows part of the frustration of the user finding the tool is not up to the task of which is required within the environment company X runs. The QUIS and requirements matching results show that this is a problem with Dollar Universe being classed within the "basic" category of batch scheduling solutions, the expert users comment simply backs up this view.

Control-M user's replies also add weight and value to the findings of the overall project. One user states "Quite nice experience using the tool so far powerful as well. A distinct advantage being, you get an overall status of the application that you are running looking at the Control M GUI." This suggest, as does the findings of the QUIS, the GUI style of user interface being preferred to that of the command line style which Dollar Universe follows.

One other user of Control-M states "I believe that exposure to the system will allow familiarity, and then a few of the ticks in the survey can move up one or two places" which helps to narrow the gap between the experience of AutoSys and Control-M user responses suggesting that Control-M user's may improve their scores given more time with the tool in question. This also helps validate that 17 responses from Control-M compared to the 7 of AutoSys is not as drastic a gap as first seen. The fact that the AutoSys users are more experienced compared to the relatively new users of Control-M shows then need for more Control-M responses to gain a fair sample of the user group.

It can be taken that this QUIS has been a valid method for testing the hypothesis of:

- The best suited tool of choice will have a user interface which the staff believes to be the best quality in terms of meeting their wishes as a tool for batch scheduling.

This reason being the QUIS has collected the views and opinions of the staff within company X. These views are expert opinions as the users are experienced in using both batch scheduling tools in general and also the tools specifically in question. This can be seen via the introduction section of the QUIS which, as spoke about previously, validates the responses of the users. Therefore the overall results of this HCI evaluation in the form of a QUIS can help to identify the tool of choice which may prove or disprove the hypothesis outlined above. The QUIS also helps to complete the following objective which was outlined within the aims and objectives section of research.

- Develop a suitable HCI evaluation method for the research via using the results of the literature review to then develop an evaluation which each tool can be evaluated against and conclusions drawn. This will be used as the HCI evaluation for the project and hopefully provide valid outcomes and conclusions based on the method developed with understanding and knowledge from the literature review being used as the basis for it.

This shows the valid process of research which has been taken place from the beginning of the project throughout. The author has identified an objective and developed it through the research time period to help validate and test the ideas and hypothesis. This development of an HCI evaluation in the form of a QUIS and the actual relative evaluation using expert users of these tools lead to the above objective being completed.

As you can see from the results above AutoSys is the leader in all areas of the user reaction evaluation. However when you take into consideration that the user group responding from the AutoSys group were far more experienced in using the tool compared to that of the Control-M users. With this taken into consideration it can be seen that the AutoSys users are simply more comfortable and experienced with the tool they use to perform the batch scheduling tasks. Therefore any slight difference in user response, such as in the above graphs, between Control-M and AutoSys could be put down to this gap in users experience with the tool in question. This can be seen with the response to the question “How long have you worked with the tool in question?” where by Control-M average response was less than 1 year and AutoSys response was 5+ years, this being due to the Control-M tool being relatively new throughout company X. The author therefore believes that if the Control-M users had the same experience with the tool as that of the AutoSys users there would be a reduction in this gap between the user satisfactions. One possible way of measuring if this is to perform the QUIS again after a given period of time, a suggestion of 2 years is reasonable, to allow for Control-M users to gain more experience and knowledge with the system. This would then perhaps show a reduction between Control-M and AutoSys in the QUIS results. However it must be stressed again that this gap is not one which is massive enough between the two tools to completely disregard one tool from the other.

This conclusion to the results of both the requirements matching and HCI QUIS evaluation is in keeping with the initial objective shown previously in the research and highlighted again below:

- Evaluate the tools used for batch scheduling in company X both via the requirements checklist and the HCI evaluation to give outcomes and results.

> This will be the 2 part evaluation of the HCI and requirements which will ultimately allow for the research to gather and develop outcomes and conclusions via these2 main data gathering techniques.

This objective has been reached via the completion of and analysis of the 2 sections of the research. The final conclusions, findings and thoughts of the author are summed up in the following conclusions section.

The completion of both the requirements matching exercise and the HCI QUIS evaluation concludes the practical research gathering and analysis which is shown above. The following conclusion section of the report outlines the overall findings and thoughts of the author based upon the process which has been followed through this case study process of evaluating 4 batch scheduling tools.

5 Overall Conclusions

Through the process of research, developing appropriate evaluation methods and performing a 2 stage evaluation of the tools within company X various conclusions and outcomes of the research can be noted and drawn. The requirements matching and HCI QUIS evaluation of the 4 batch scheduling tools used within company X has allowed for these conclusions to become clear.

5.1 Summary

Overall this research project has had a valid process and reasoning throughout its stages of its lifecycle. Ultimately the research has aimed to answer the question of:

- Which of the 4 batch scheduling tools identified would best meet the business needs of Company X?

To do this the author has performed a detailed case study using company X. During this a set of requirements for batch scheduling have been identified and used to analyse 4 of the tools in use within company X. These results have then been analysed above to identify the differences between the tools, their features and styles. Along side this requirements matching a detailed HCI evaluation has been developed and used to asses the tools which are used within the case study company. This HCI evaluation took the form of a QUIS which assessed the expert user's opinion of multiple key sections of the software. Again these results were analysed and assessed to determine the overall reactions and findings of the users between each tool. From this overall research, which has been validated via close communication and cooperation with managers and working groups within company X, conclusions and solutions have been drawn by the author with regards to the question and process which has been followed.

5.2 Requirements Matching Conclusions

The research is in depth and valid throughout the case study process of company X. The research shows that clearly there are a number of requirements which company X need. It also shows that, through clear communication and cooperation with company X, a set of requirements for batch scheduling has been created. The subsequent list of requirements which has been used to asses each of the 4 tools allows for a comparison between each. These results which are explained in the results section of the report highlighted the gap between Control-M and AutoSys when compared to Dollar Universe and Cron. This section of the research was also effective at highlighting Cron's lack of ability to handle large scale batch scheduling that a decision was taken not to perform the QUIS evaluation on the tool due to it being a waste of resources for such an ineffective tool. The results did show that Control-M is the most effective tool which supports the most requirements in the list and supported company X's European choice of tool. However, AutoSys was only slightly behind thus the QUIS would help to analyse the interface of the tools using the staff of company X as the experts.

5.3 HCI Conclusions

This second section in the form of an HCI QUIS then highlighted the gap again between the 2 tools, Control-M and AutoSys, and that of Dollar Universe. The difference or reason for this gap being that Control-M and AutoSys use Graphical User Interfaces which offer rich user interfaces with point and click operations being backed up with on screen visuals that assist the user in their daily operations, however, Dollar Universe uses a dated command line interface where by the user must interact via commands and code. This difference is reflected in the results as Dollar Universe lags being the other tools in almost all aspect of the QUIS highlighting the user's satisfaction levels are met far more successfully when they have the GUI style system for interaction. This is also reflected within the requirements matching exercise where by Dollar Universe again falls behind the other 2 tools. The reason being Dollar Universe falls under the basic category of scheduling solutions where as Control-M and AutoSys are classed as advanced solutions which best meet the needs of company X.

5.4 Final Overall Conclusions

Control-M therefore falls behind AutoSys in terms of user satisfaction. However, AutoSys users were more experienced in using the tool in question compared to the Control-M user group. This would suggest that the user group for Control-M may grow in confidence and satisfaction given the same experience with the tool as the AutoSys users. It has been highlighted on a number of occasions that the gap is only slight between the tools and it is not vast enough to disregard one from the other. The research conclude that Control-M may not satisfy the users like that of AutoSys however, the tool does satisfy the business needs of company X better than that of AutoSys, as shown in the requirements matching exercise. However, the margin between both these tools is so narrow that there is no real tool of choice which stands out from the other enough to choose 1 specific solution for company X. For this reason it would be specifically company X's own preference to choose between wither Control-M or AutoSys. This research simply highlights these two tools as both effective solutions to large scale batch scheduling and suitable for company X.

Overall the final and most important conclusion, which the author has drawn from the research, and the conclusion to the overall question "Which of the 4 batch scheduling tools identified would best meet the business needs of company X?" suggests that of these 4 tools, Control-M is the tool of choice for company X. Considering the situation regarding company X where by Dollar Universe is being phased out and replaced by Control-M within the European region, yet AutoSys is the tool of choice in America. It can be seen that company X have made the correct decision to migrate from Dollar Universe as it is a basic solution to the advanced batch scheduling problem the company faces. The choice of tool between America and Europe however is where the problem lies. The company should look to review the batch scheduling situation again in the coming years comparing the size of the user group of Control-M and AutoSys and their satisfaction levels via a QUIS, similar to the one performed during this research. This would allow for the Control-M user group to have grown and settled with the tool, allowing for a better comparison between tools and perhaps tipping the balance in Control-M's favour in both the requirements matching and QUIS evaluations. At present there is no need to migrate the AutoSys group to Control-M and indeed such a move

would be potentially dangerous considering Control-M is relatively new and potential long term problems could become an issue for company X to deal with. Running both systems in parallel allows for company X to have 2 effective solutions which could, at a later point should it be required, allow for a migration of AutoSys onto Control-M for a company wide solution. This would allow for one solution which allows for close control and monitoring of all batch scheduling jobs via one tool, infrastructure and system.

A summary of the conclusions reached are:

- Graphical User Interfaces often help the user's to interact and perform their tasks easier than that of Command Line systems like Dollar Universe.
- Cron is not an effective solution for large scale batch scheduling as it is of a native nature
- Dollar Universe is not as effective at large scale batch scheduling and can be classed within the basic category of batch scheduling tools
- Dollar Universe does not satisfy the user's when they are interacting with the system like that of Control-M or AustoSys.
- Control-M does beat all of the other evaluated tools in the requirements section therefore the more effective solution.
- AutoSys does satisfy more users however the users are more experienced and comfortable using the tool. Control-M on the other hand not only satisfies the users but also the needs of the business.
- Control-M is the sensible tool of choice for the European migration away from Dollar Universe and AutoSys, the American tool of choice, which could perhaps be phased out at some point in the future should company X demand a global solution to batch scheduling.

5.5Additional Conclusion

However further to the conclusions above the research also has another outcome which could benefit not only company X but any organisation or batch scheduling vendor wishing to grade their product on a non bias and fair checklist. Based on the above results, findings and conclusions the author proposes that the developed batch scheduling requirements checklist, which has been implemented during the research, allows for the 3 categories of scheduling tools to be identified. The author feels that based upon the result it can be taken that an advanced scheduling tool falls under that which matches 75% of requirements or more. Basic scheduling solutions fall between 35% and 74% where as native falls below the 34% mark. These proposed categories are based on the basis on ORSYP, vendor of Dollar Universe, research (ORSYP resource centre, 2009) which identify the 3 categories and suggest the ability and features of each. Matching the categories with a percentage of requirements matched shows those which are covering little or varied requirements should fall under this first and most simple category which is native. The tools which have more that native and less than advanced are known as basic and the author believes this falls between 35% and 74% using the requirements checklist developed during the research as it covers the native features and majority of features that most customers would need. However, those tools which hit 75% or more are those which cover the more advanced features and those which only large organisations and companies will require as a batch scheduling solution. This is

based upon the findings of the requirements matching which has taken place. As the outcome clearly shows that advanced tools are separated from basic and likewise basic are separated from native in terms of the gap between numbers of requirements met by each tool. This gap is seen to be enough between all 3 types of batch scheduling solutions that the author believes it can help categorize any batch scheduling tool. The gap between the tools is 35% or greater which the author feels is sufficient to place a tool into a specific category. This 35% gap between each category indicates there a clear difference between the types of solution which can be used and also allows for batch scheduling vendors to improve their tools to a standard which would class it as advanced thus appeal and benefit their clients better.

It can therefore be seen that the research has developed an in depth batch scheduling software evaluation which incorporates both an HCI evaluation and requirements matching evaluation which can be applied to any batch scheduling tool in order to obtain its classification and benchmark satisfaction levels. These can then be compared against the results and findings of the 4 tools within this particular research and could even be used to evaluate future iterations or releases of the tools assessed through this project.

6 References

ActiveBatch, (2006), *ActiveBatch scheduler checklist*, Available from: http://h21007.www2.hp.com/portal/download/dspp/files/unprotected/partners/5383/ProductCatalog/FriFeb04010350259PM36505052005%20Job%20Scheduling%20Criteria%20Checklist.pdf

Afke Donker, & Panos Markopoulos. (2002). *Comparison of think-aloud, questionnaires and interviews for testing usability with children.* Eindhoven University of Technology.

Andreas Holzinger (2005) *Usability engineering methods for software developers*

Arif Mohamed (2006), *How grid power pays off for HSBC.*

BBC News, (2008). *Timeline: Global credit crunch.* Retrieved November, 2008, from http://news.bbc.co.uk/1/hi/business/7521250.stm

BMC. (2002). Business integrated scheduling for banking and finance.*1*

BMC Manual. (2005). *Control-M/Enterprise manager user guide*, BMC.

BMC Publication. (2008). *BMC CONTROL-M – an IT workload automation platform*. BMC Website: BMC.

BMC Website, (2008). *BMC website.* Retrieved November, 2008, from www.bmc.com

CA. (2007). *CA autosys workload automation solution implementation* No. MP311940808, CA.

Carroll, J. M. (1997). Human-computer interaction: Psychology as a science of design. *International Journal of Human-Computer Studies, 46*(1), 501-501-522.

Central Intelligence Agency. (2008). *The world factbook*. https://www.cia.gov/library/publications/the-world-factbook/index.html: Central Intelligence Agency.

Chin, J. P., Diehl, V. A., & Norman, K. L. (1988). *Development of an instrument measuring user satisfaction of the human-computer interface*, in CHI '88 Conference Proceedings: Human Factors in Computing Systems, 213-213-218.

Clark, E. (2002*). The death of British manufacturing?*, 2008, from http://news.bbc.co.uk/1/hi/business/1871493.stm

Crosby, R. W., & Greiner, C. (2006), *Keeping pace with mainframe evolution, 2006, 1*(1)

D. Kallal, A. (2007). *Running batch jobs in ms-access: What ever happened to the punched card?* Retrieved October, 2008, from http://www.members.shaw.ca/AlbertKallal/BatchJobs/Index.html

David Meister. (2003), Conceptual foundations of human factors measurement. *Lawrence Erlbaum Associates,*

David R. Millen. (2000). Rapid ethnography: Time deepening strategies for HCI field research. *Proceedings of the 3rd Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques,* 280-280-286.

Ferrarini, E. (2003). *Introduction to enterprise job scheduling: A crash course in cutting IT ops costs.*, 2008, from http://nextslm.org/jobscheduling.shtml

Gartner. (2007). *Magic quadrant for job scheduling* No. RA2 05282007, Gartner.

Gartner Group. (2000). *Research notes on batch scheduling solutions,* Gartner.

Gerald Kotonya, & Ian Sommerville. (1998). *An introduction to requirements engineering* (1st ed.) Wiley.

Greenbaum, J., & Kyng, M. (1991). *Design at work: Cooperative design of computer systems* (1st ed.) Lawrence Erlbaum.

Günther Gediga, Kai-Christoph Hamborg & Ivo Düntsch, (2001), *Evaluation of Software Systems*, Encyclopaedia of Computer Science and Technology

Hartson, H. R. (1996). *Human computer interaction: Interdisciplinary roots and trends*, Department of Computer Science Virginia Tech.

Hix.D, & Hartson, H. R. (1993), Developing user interfaces: Ensuring usability through product and process. *New York; Wiley,*

ISO, (2009)., *International organization for standardization website,* Retrieved January, 2009, from http://www.iso.org/iso/home.htm

Jacob Nielson. (2009). *Jacob Nielson website.* Retrieved November, 2008, from http://www.useit.com/

James R. Lewis. (1993). *IBM computer usability satisfaction questionnaires: Psychometric evaluation and instructions for use*, No. Technical Report 54.786, Human Factors Group.

Jurek Kirakowski, (2000), Questionnaires in usability engineering, *Human Factors Research Group,*

Laura Faulkner, (2003*) Beyond the five-user assumption: Benefits of increased sample sizes in usability testing*, Behaviour Research Methods, Instruments, & Computers, *35 (3), 379-383*

Lewis, C. (1982), *Using the "thinking aloud" method in cognitive interface design*, New York: IBM *Research Reports RC 9265 (40713), IBM Thomas J*. Watson Research Center.

MVPSI Systems, 2007, *MVPSI Systems Enterprise Scheduling Checklist*, Available from: http://www.mvpsi.com/

ORSYP, (2009), *Dollar universe website*, Retrieved November, 2008, from www.ORSYP.com

ORSYP, (2008), *Job scheduling: The right tool for a tough economy*, ORSYP White paper.

Patrick Weaver, (2006), *A brief history of scheduling*, Mosaic Project Services PTY LTD.

Praxiom Research Group, (2004), *ISO/IEC 90003 introduction*, Praxiom Research Group.

QUIS, (2008), *QUIS website*, Retrieved January, 2009, from http://lap.umd.edu/quis/

R. D., Smith, M. C., Podd, J. & Varela-Alvarez, H, (1995), *Protocol analysis and Verbal Reports on Thinking*, available from: http://www.psy.fsu.edu/faculty/ericsson/ericsson.proto.thnk.html

Robin Jeffries, & Heather Desurvire, (1992), Usability testing vs. heuristics evaluation: Was there a contest? *ACM SIGCHI Bulletin, 24*, 39-39-41.

R. Oppermanna, & H. Reitererb, (1997), Software evaluation using the 9241 evaluator, *Behaviour & Information Technology, 16*(4/5), 232-232-235.

Sabine McConnell, (2006), Distributed predictive and descriptive data mining. Queen's University.

Scriven, M, (1967), The methodology of evaluation. In G. Tyler, & M. Scriven (Eds.), *Perspectives of curiculum evaluation* (1st ed., pp. 39-39-83) Rand McNally.

Tidal, (2008), *Guide: tidal enterprise scheduler*, Tidal.

Tim Grieser, (2007), *The Evolution of Job Scheduling: CA's Approach to workload Automation* ,IDC

UC4 Software, (2008), *Beyond job scheduling: The road to enterprise process automation*, UC4.

Unicenter, 2006, Unicenter AutoSys Job Management Version 4.0, Available from: http://venkatstore.tripod.com/ML-GELP/html.html#4.0

Wayne D. Gray, & Marilyn C. Salzman, (1998), *Damaged merchandise? A review of experiments that compare usability evaluation methods. Human-Computer Interaction, 13*(3), 203-203-261.

Whil Hentzen, (2004), *Cron explained*, Hentzenwerke Whitepaper Series.

7 Bibliography

Anonymous. (2004). FDIC issues guidance on software evaluation. *ABA Bank Compliance, Washington, 25*(12), 9.

Anonymous. (2005). Best practices: Software evaluation for mid-sized organisations. *Business Credit, New York, 107*(8), October 2008.

Anonymous. (2007). Want to waste time and money? Then get a computer. *Irish Times,* 12.

Anonymous (2004). FDIC issues guidance on software evaluation. *ABA Bank Compliance, 25*(12), 9.

B. Wong, & R. Jeffery. (1995). Quality metrics: ISO9126 and Stakeholder Perceptions. *The Second Australian Conference on Software Metrics,* 54-54-65.

B. Wong, & R. Jeffery. (2002). A framework for software quality evaluation. *A Framework for Software Quality Evaluation,* 103-103-118.

C. V. Ramamoorthy, & S. F. Ho. (1975). Testing large software with automated software evaluation systems. *The International Conference on Reliable Software,* 382-382-394.

Carney, D. J., & Wallnau, K. C. (1998). A basis for evaluation of commercial software. *Information and Software Technology, 851*(860)

Choi, H., & Nicholas, K. M. (2005). Software evaluation: EasyReg international.

Hutchinson, T. E., White, K. P. J., Martin, W. N., Reichert, K. C., & Frey, L. A. (1989). *Human computer interaction using eye-gaze input, "Systems, man and cybernetics,* IEEE.

ISO. (1997). ISO 13407. *International Organization for Standardization,*

ISO/IEC. (1991). ISO/IEC9126 information technology software product evaluation - quality characteristics and guidance for their use. *International Organization for Standardization,*

Jacob Nielsen, & Donald A. Norman. (2001). From use to design. *Wireless world: Social and interactional aspects of the mobile age* (1st ed., pp. 195-195-206) New York ACM.

Jacob Nielson. (2009). *Jacob Nielson website.* Retrieved November, 2008, from http://www.useit.com/

John Rieman, Marita Franzke, & David Redmiles. (1995). Usability evaluation with the cognitive walkthrough. *Conference on Human Factors in Computing Systems,* 387-388.

Johnston, J. (2005). VSE: A look at the past 40 years.

Joseph Kaye. (2006). Evolution of evaluation in HCI. *HCI Seminar Series,*

Karat, C., Campbell, R. L., & Fiegel, T. (1992). Comparison of empirical testing and walkthrough methods in user interface evaluation. *Proceedings ACM CHI '92 Conference,* 397-397–404.

Nielsen, J., & Mack, R. L. (1994). *Usability inspection methods* (1st ed.) Wiley.

Nielsen, J., & Molich, R. (1990). Heuristic evaluation of user interfaces. *Chi'90 Conference Proceedings, Empowering People,* 249-249-256.

O'Donnell, A. (November 2003). Making batch scheduling easier. *Insurance and Technology, 28*(11), 17.

Ohnemus, K. R., & Biers, D. W. (1993). Retrospective vs concurrent thinking out loud in usability testing. *Proceedings of the Human Factors and Ergonomics Society 37th Annual Meeting,* 1127–1131. .

Quadt, D., & Kuhn, H. (2007). Batch scheduling of jobs with identical process times on flexible flow lines. *International Journal of Production Economics, 105*(2), 385.

R.W. Root, & S.Draper. (1983). Questionnaires as a software evaluation tool. *Conference on Human Factors in Computing Systems, the SIGCHI Conference on Human Factors in Computing Systems,* 83-83-87.

Roberts, T. L., & Moran, T. P. (1983), The evaluation of text editors: Methodology and empirical results. *Communications of the ACM, 46,* 501-501-522.

Scott MacKenzie. (1992). Fitts' law as a research and design tool in human-computer interaction. *Human-Computer Interaction, 7*(1), 91-139.

Smith, S. (2000). Will you be able to face your bank manager? *Accountancy, 125*(1282), 52.

Stamelos, I., & Tsoukias, A. (2003). Software evaluation problem situations. *European Journal of Operational Research, 145*(2), 273.

Timewell, S. (2008). Retail banking: Harnessing technology to create tomorrow's retail bank. *The Banker,*

Tory, M, & Moller, T, (2005), Evaluating visualizations: Do expert reviews work? *Computer Graphics and Applications IEEE, 25*(5), 8-8-11.

Veikko Surakka, Marko Lllim, & Poika Isokoski. (2004), Gazing and frowning as a new human--computer interaction technique. *ACM Transactions on Applied Perception (TAP), 1,* 40-56.

Wong, B. (2004). The software evaluation framework "SEF" extended. *Faculty of Information Technology, 46*(15), 1037.

Wong, D. H., Rexha, N., & Phau, I. (2008). Re-examining traditional service quality in an e-banking era. The International Journal of Bank Marketing, *26*(7), 526.