Spring 2006

Developing a Methodology for Creating Flexible Instructional Information Technology Laboratories

Daniel J. Ziesmer
Regis University

Developing a Methodology for Creating Flexible Instructional Information Technology Laboratories

by

Daniel J. Ziesmer

A Project Report submitted in partial fulfillment of
the requirements for the degree of
Master of Science in Computer Information Technology

School for Professional Studies
Regis University
Denver, Colorado
April 21, 2006

Revision/Change History

Document	Date
Outline	July 2002
Abstract	August 2002
Chapter 1	August 2002
Chapter 2	August 2002
Chapter 3	January 2006
Chapter 4	January 2006
Chapter 5	February 2006
Appendices	August - October 2002
Chapter 1 – Revision	February 2006
Chapter 2 – Revision	February 2006
Chapter 3 – Revision	March 2006
Chapter 4 – Revision	March 2006
Chapter 5 – Revision	April 2006
Appendices – Revisions	April 2006
Chapter 1-5 – Final Revisions	April 2006

Acknowledgements

I would like to thank Mr. Donald Archer and Mr. Samuel Conn for the time they spent with me in working out the details of writing this project; I would like to express my appreciation for their patience following the long delay between the completion of my courses and the conclusion of this project. I would also like to thank my wife, Amy, who graciously tolerated the lost weekends, late nights, and general disruption to our lives during these years; without her constant assistance and understanding I would not be here today. Lastly, I would like to thank the institution and the employees of San Juan College who provided me with the opportunity to apply the principles of this project to the actual organizational operations; through various experiments and incarnations of lab design, I was afforded a unique chance to convert my ideas into reality.

Abstract

Developing a Methodology for Creating Flexible Instructional
Information Technology Laboratories

by

Daniel J. Ziesmer

April 21, 2006

Many schools – particularly the more dynamic segments of high schools and community colleges – have begun to undertake instruction in the areas of PC repair, networking (vendor-neutral and specific alike), operating systems, wireless technologies, and so forth. For some schools, however, this leap forward has come only with a later realization that there are tremendous startup costs and ongoing expenses associated with such endeavors, especially considering that many of these instructional elements have historically called for independent instructional facilities. From this perspective, institutions may find they have to cut their programmatic vision short in the face of harsher budgetary realities of supporting so many laboratories, or abandon their efforts altogether.

In this paper, it is suggested that this scenario does not have to become a reality. Instead, it is proposed that affordable, functional, and practical multipurpose Information Technology (IT) classrooms can be developed when a combination of good initial design and planning, affordable technologies, and mature business models are practiced. With the application of certain methodologies, a system can be created for any institution wishing to develop facilities and the means to support and mature them over time.

Often faced with budgetary constraints, space limitations, or uncertain financial support mechanisms, it is becoming important that higher education institutions engaging in the instruction of advanced computing and networking develop a process and methodology for establishing and maintaining computing laboratories that can service a variety of diverse and complex instructional needs.

.

Table of Contents

List of Figures

Chapter One: Introduction

Problem Statement

Once designated to the realm of vendor or commercial-based training facilities, information technology (IT) training and certification has rapidly grown in popularity in recent years at all levels of academic instruction as well. While this development has been tremendously positive for supporting the growing demand of a knowledgeable, well-trained information technology workforce, the ability to adequately address the infrastructure needs of the training programs themselves have gone largely unaddressed. The traditional laboratory design models historically applied to programming and application training courses are inappropriate in an IT environment, and new approaches are necessary. Furthermore, as budgets at all academic levels are continually eroded or stretched to meet existing obligations, the significant funding normally necessary to establish appropriate IT facilities will become more and more difficult to obtain.

This project proposes that – despite the unique instructional challenges of teaching many of the information technology courses and certifications today – affordable, functional, and practical multipurpose Information Technology classrooms can be developed when a combination of good initial curriculum planning, laboratory design, affordable technologies, and mature business models are practiced. With the application of certain methodologies, a system can be created for any institution wishing to develop facilities and the means to support and mature them over time.

Reasoning for Project

Many schools – particularly institutions such as high schools and community colleges – have chosen to undertake instruction in the popularly growing professional certification areas of PC repair, networking (vendor-neutral and specific alike), operating systems, databases, wireless technologies, and so forth. For some schools, this choice is a tremendous leap forward on many levels: although it provides broad new possibilities for their students, it often comes with a later realization that there are tremendous startup costs and ongoing expenses associated with such endeavors, especially considering that many of these instructional elements often call for independent or isolated instructional facilities, or constant replacement and implementation of the most current hardware and software technologies available. From this perspective, institutions may find they have to cut their programmatic vision short in the face of harsher budgetary realities of supporting so many laboratories, or abandon their efforts altogether.

Consequently, when faced with budgetary constraints, space limitations, or uncertain financial support mechanisms, it is becoming important that educational institutions that engage in the instruction of the information technology discipline develop a process and methodology for establishing and maintaining computing laboratories that can service a variety of diverse and complex instructional needs (Madison, 2002). Although research institutions and universities may have more opportunities to obtain the funding necessary to support specialized or single-purpose training facilities, smaller colleges, community colleges, and secondary schools are seldom as lucky (Patterson, 2005).

As an instructor of information technology and IT-certification programs at a moderately-sized community college, the author understands the importance of proper planning and budgeting when investing in new laboratory facilities. This process can be severely hampered, however, by a lack of knowledge or planning at the curriculum level. Without an understanding of what must be taught and the technologies involved/necessary, it is difficult to make a fair or accurate assessment of what funding needs to be invested.

In this project the author utilized his professional experiences and academic training to create a methodology for developing an IT laboratory design – including example layouts, technology recommendations, and future expansion planning – based upon popular professional IT certification curricula. It is hoped that the results of this project can be used to allow other institutions to make informed decisions when developing their own multipurpose laboratory system.

Anticipated Problems

There are three commonly associated problems facing IT instruction today:

1. Necessitated Rapid Curriculum Development

   Changes in any curriculum are inevitable, but its occurrence is much more frequent, and handling it is extremely difficult for many IT instructors today. Instructors and professors are expected to not only be able to stay abreast of new technologies and understand their practical implementations, but to also become subject matter experts in relatively short periods of time. Hence, this project must necessarily take into consideration the users of such facilities and the importance of the facilities to successfully incorporate as much flexibility into the design as possible (including technologies or subject matter that, as of yet, may not even exist).

2. Rapid Changes in Technology

   The situation is further compounded by the fact that a facility cannot remain static, but must be amenable to changes in the technology upon which it is based without affecting the multipurpose design and uses proposed. It will be vital to select current technologies that are upgradeable, replaceable, or can be implemented redundantly, yet done with minimal disruption to the facility's regular operations and multipurpose design.

3. Resource Allocation for Instruction in New Technologies

   In an ideal situation, a limitless budget and a plethora of expensive, dynamic technology systems could be used to establish an outstanding, fully automated, multipurpose facility. Most higher education environments, however, do not have ideal fiscal situations. Financial constraints, particularly in an expensive area like IT, weigh heavily on schools – and especially on those in which lower enrollments or fixed institutional funding mechanisms are strong or impenetrable barriers to larger budget proposals. In making recommendations on facility design and equipment, budgetary constraints should be given priority and facilities and equipment should be selected in such a way to minimize funding requirements wherever possible. It is especially important to strike an appropriate balance between functionality and costs.

In addition to problems faced within the field of study, this project also faces numerous practical hurdles. Research in the field of lab design is relatively narrow in scope, and most literature today focuses on the incorporation of technology into traditional classroom environments. The study of teaching information technology as a separate, unique field of study is quite new, and has only become a topic of discussion within the past five years or so. A review of literature did not produce more than a handful of discussions on the topic, resulting in the approaches taken in this project being somewhat *experimental*. Finally, there is a strong disdain for professional certification as a basis of academic instruction by certain types of institutions, and by many individuals in general. Some may dismiss this approach off-hand because of the selection of this curriculum model, but the process is intended to be sound for any well-defined curriculum model one chooses to use.

Anticipated Opportunities

In contrast to the barriers that lie in the way, this project can provide several unique opportunities for its intended recipients:

1. Documented Processes

   The development of a specific methodology in this project, even with its reduced scope, can reduce guesswork when implementing similar IT lab systems, whether they are educational, professional, or otherwise. Instructors can use this document as a set of guidelines for specifically defining needs for physical space, equipment, or funding.

2. Diversified Usability

   A methodology relying on competencies neutrally established by diverse groups (i.e. professional certification) should allow those establishing curricula to implement the results of this project without institutional or instructional bias in the results. Moreover, using fixed standards bodies (relevant to both vendor-neutral and vendor-specific certifications) should more easily allow for incorporating the new technologies, certification areas, or instructional methodologies from those bodies over time. In other words, this project can serve as an excellent foundation for developing curriculum and multipurpose labs that will be equally functional both now and in the future.

3. Curriculum Expansion

Academic instructors may make anecdotal analyses that lead them to believe that the facilities or equipment they have are inadequate to support new courses or topics. This document may serve to eliminate guesswork and identify or analyze the constant barrage of new technologies that are critical and/or preferred in a variety of IT-related certification areas.

4. Identification of Appropriate Technology

Equipment vendors, and even internal IT departments at schools and colleges, may often insist that their chosen solution is the only appropriate *professional* option when it comes to computers, networking equipment, network design, etc. in the classroom. With some careful consideration and a bit of research, however, most educators at the high school and college levels will find that there are many inexpensive or alternative technologies that are perfectly suitable for and comparable to the professional environments that students are likely to encounter in the workplace. This project will hopefully give some footing to educators who are looking to learn about how to identify types of technologies and “minimum required hardware/software” standards, instead of being told what specific technologies will or will not work.

Elements to be Discussed

In this project, several different perspectives will be addressed as part of the process of moving from concept to implementation:

1. General. The project itself can serve as a practical guide that IT instructional planners can use to coordinate their own developmental process for a multipurpose IT facility and the equipment contained therein.
2. Course Competencies. Before proper facilities planning can take place, an initial overview of the instructional objectives of various classes must be defined. A matrix is included to provide an example methodology of identifying the core competencies of popular academic and certification-based courses that can be used by an institution for mapping the placement of their own various courses into common rooms.
3. Operations Maturity. A breakdown of the various levels of IT training integration will be made according to available facilities, equipment, budget, and time constraints. It is important to provide some examples of the levels of maturity that a program can experience, and recognize that programs can be started without expensive or complex infrastructure investments.
4. Practical Implementation. Several appendices are included identifying examples of physical facilities layout, logical design, and popular, common, or affordable technologies for establishing facilities and implementing training equipment at the various maturity levels.

Scope of Project

The project will focus on creating a system for providing recommendations related to coursework in the field of information technology – more specifically, it will be limited to programs that provide instruction similar or equivalent to areas of coverage necessary to prepare for and successfully complete popular professional IT certification examinations.

Because the scope of research and general theory-based curriculum can be so broadly defined, and given that the scope of this type of instruction generally varies widely from institution to institution (and from professor to professor dependent upon individual research interests), analyzing lab design from this perspective becomes almost entirely meaningless. The individual needs or desires of each professor or institution would invariable demand unique lab designs for each, and thus a comprehensive planning approach would be very difficult.

Conversely, adhering to professional or commercial certification examination content provides a consistent set of guidelines that transcend instructor, institutional, state, and even national definition. Certification examinations and the competencies they define are created with the input of hundreds of professionals within each exam's respective field of study, along with the input of a wide array of businesses, employers, and academia from throughout the world. Although still distrusted by some areas of academia, certifications are the most universally accepted criteria in the IT industry that reflect an unbiased, common frame of reference of what competencies can be expected from individuals certified in a particular area of study. Moreover, regardless of certain perceptions on the value of certification, they are nonetheless excellent tools for establishing a set of refined course competencies, they are commonplace requirements for most IT-related job openings today, they are well supported with tools and materials by virtually every academic publisher and course designer today, and they are firmly established in industry as having measurable academic, professional, and financial benefits for individuals.

Finally, the project is in no way intended to provide recommendations on specific facilities, equipment, or processes related to the process of teaching, but rather will focus only on IT resources and equipment that either allow for students to have practical hands-on experiences or which specifically support a suggested or required curriculum methodology as specified by IT certification vendors (i.e. Cisco, CompTIA, Microsoft, Oracle, etc.). See Appendix B for a list of the courses and/or certifications that have been selected for inclusion.

Definition of Terms

ACM: The acronym for the Association for Computing Machinery. According to their website, the organization can be described as "an international scientific and educational organization dedicated to advancing the arts, sciences, and applications of information technology. With a world-wide membership, ACM is a leading resource for computing professionals and students working in the various fields of Information Technology, and for interpreting the impact of information technology on society" (ACM, 2006).

Certification: A measurable record of a person's skills or product knowledge. Most certifications are sponsored by either corporations, which are usually relevant to a corporation's particular products or technological area of specialization, or by nonprofit organizations, whose intent is to provide an industry standard measurement for a specific type of technology or body of knowledge.

CIW: The acronym for the Certified Internet Webmaster program, a vendor-neutral certification program sponsor by Prosoft Corporation specifically intended to authenticate a person's knowledge of Web-related technologies, best practices, and business operations. It is one of the only web-related certification in the industry as of this writing.

CompTIA: The acronym for the Computer Technology and Industry Association. According to their website, the organization's goal is to "provide a unified voice, global advocacy and leadership, and to advance industry growth through standards, professional competence, education and business solutions. In order to most efficiently serve the industry and its members, CompTIA has developed specialized initiatives and programs dedicated to major areas within the IT industry. They include convergence technology, e-commerce, IT training, software services, certification, public policy and workforce development" (CompTIA, 2006).

CWNA: The acronym for Certified Wireless Network Associate, it is the first of three certifications in the only vendor-neutral program focused on wireless networking, applications, and security.

MCSE: The acronym for Microsoft Certified Systems Engineer, it is presently the highest level of certification by Microsoft Corporation on the knowledge of its client and server operating systems.

Virtualization: The process of creating an environment in which an entire operating system (virtual machine) runs as an application (guest) on another operating system (host) installation. Virtualization software is designed to present a simulated hardware environment to each virtual machine, which is in fact contained within the software application running on the host system. The simulated environment must be sufficiently powerful enough to allow for a guest operating system to function as if it were installed on a dedicated hardware platform (including memory, processing power, and driver support).

Summary

So often in the various fields of information technology, an instructor's emphasis is placed on students taking the knowledge obtained from a training regiment and applying it to outside businesses, organizations, research, and other external professional situations. What tends to be forgotten, however, is that this information is equally relevant and useful for instructors and programs as well, particularly when it comes to developing the technology infrastructure that allows students to obtain that knowledge in the first place.

Most educators focus on providing the highest level of training and education possible with the resources available; however, the fiscal demands of IT often leave many dismayed in the wake of declining state support and/or limited private funds. With this project, the author hopes to provide some alternative mechanisms that allows for educators to continue their mission, remain within budget, and not be forced to sacrifice quality in the process.

Chapter Two: Review of Literature

Introduction

It is ironic that the term most often associated today with anything related to computers is "information technology". Information technology actually represents the most recent broad discipline in the field of computing, but its high visibility in business environments and its rapid evolutionary tendencies bring it to the forefront of everyday vocabulary. In the early 1990's computers became the de facto tool of a modern business, and information technology found itself embedded in the functional, day-to-day operations of nearly every facet of an organization's operations. With this change came the inevitable problems of installing, maintaining, and upgrading this infrastructure that – today – serves as the backbone of companies. IT departments soon developed within organizations, and took upon the responsibilities of ensuring that the infrastructure was correctly designed and installed, operated reliably, and met the needs of the people in the organization.

It was not until the end of the 1990's, however, that academic entities began to respond en masse to these changes within the field of computing. Traditional computer science and management information systems (MIS) programs were not producing graduates who had both the knowledge and practical skills to function in this environment. The first to respond to these needs were, as usual, private training firms and independent training programs within organizations, but technical institutes and community colleges were soon developing certificates and associate degrees to provide an academic perspective on this subject. Information technology certification also began to surface during this same time: employers found themselves less able to rely upon the intrinsic value of a bachelor's degree to demonstrate the requisite skills necessary for information technology work, so certification provided by both vendors and vendor-neutral organizations soon made their way into the hearts of information technology and human resource directors alike. Today, institutions at all academic levels have introduced specific information technology degree programs, although – as will be seen later – not without some level of disagreement.

Information technology as an academic discipline represents the latest field of study in the area of computing. Its popularity has grown rapidly in recent years, in many instances because it undergoes an incessant change that corresponds to new hardware developments, increasingly sophisticated software design and functionality, and the rapid change of business and the organizations that implement the technology. Although commonly confused with 'information systems', information technology emphasizes the study of the technology itself, more so that the information it conveys. Information technology began in response to the practical, daily requirements of maintaining business continuity, and the discipline continues to strive to meet those needs today in its graduates: information technology employees must be able to select, configure, upgrade, maintain, replace, and support the entire infrastructure from end to end, including the end users.

Because information technology has only risen to a level of broad academic acceptance with the past five years or so, the idea of evaluating and codifying instructional lab design for information technology – much less any other computing discipline – has not been discussed in much of today's professional and popular literature for a variety of reasons:

- In recent years the thrust of discussions on classroom ‘information technology’ have been focused on implementing technologies for the benefit of pedagogical enhancement (i.e. the use of technology for teaching), and less on the actual instruction of information technology itself.
- The few articles authored about computer lab design are often the result of academic or professional persons whose articles are intended to describe their own microcosm of instructional needs, which are invariably unique to their own personal or research interests. There has never been a perceived need to analyze or extend the results to generalized lab design.
- As implied thus far, information technology training and education is a relatively new field in many academic circles, such that it is often not distinguished from the traditional Computer Science or Information Systems curricula that usually require far less sophisticated instructional lab equipment and designs, and which have had de-facto standards for a relatively long time.
- Most information technology training and education is strongly divided down two divergent academic approaches: workforce-directed education and training (long-supported and provided by industry and technical/community colleges), and the academic analysis of information technologies from more academic and theoretical perspectives (found more frequently in baccalaureate and graduate degree-granting institutions). As will be discussed later, these two approaches have resulted in a split in what is consider the “proper” educational pathway, and thus bisect what the best training facility layout should be.

Acknowledging the shortcomings of any previous body of work, this project is intended to lay a new, relatively uncharted path for considering laboratory design as a standardizable operation, as well as likely redefine some commonly held beliefs and dismiss some of the more common myths related to different types of information technology education.

Curriculum: Starting at the Beginning

The field of information technology – from an instructional perspective – is currently split into two factions: ‘practical instruction’, or instruction of existing, popular, or industry-demanded technologies, and ‘theoretical instruction’, involving coursework in the research, design, and development of new information technologies. (While these are oversimplifications, they tend to reflect a general inherent reality in academia). These factions also tend to be split down institutional boundaries as well: high schools and community colleges tend to focus on pre-existing technologies, experiential learning, and practical educational approaches, with a base of theoretical knowledge about the technologies’ functionality, while four-year and some graduate degree-oriented institutions prefer theoretical or broad-spectrum technology curriculum that is established more in traditional research and design than in implementation, forming a generalist approach to understanding such technologies.

So – why the concern with curriculum, and what does this have to do with lab design? Curriculum naturally defines what the courses in a degree program will consist of. In this field – where the subject matter is constantly being revised and renewed – educators have been given the rare opportunity to consider and select what curricular approach they intend to undertake to create the most successful program they can for both their students and their institution. To this end, each type of curricular approach has to be given some consideration.

A bachelor's degree in computing (for whatever specialization) represents the de-facto standard for theory-focused educational pursuits, and has arrived at that point by following a classically design educational model that has been in place for hundreds of years within the various scientific disciplines. The theoretical model argues that students should be given a broad, solid foundation of the technologies upon which modern hardware and software have been developed, so that students can utilize this knowledge later in life when the software and hardware change (even when the platform technology does not) (Carey, 2004). However, the traditional bachelor's degree programs in computing have been under attack from a variety of sources – and none more devastating than their own students. In the most recent Taulbee Survey (Zweben, 2005), undergraduate enrollments in computer science and computer engineering saw enrollments decline for the fifth straight year.

Two-year/associate's degrees, in contrast, have become increasingly attractive options for persons entering all computing fields, and have embraced a more functional approach to technology education. Although they do and can not inherently provide the same depth of knowledge found in bachelor's degrees that have an additional two years of study allotted to them, they are nonetheless attractive alternatives. The functional model argues that, while some theory is important, for the majority of information technology workers today the 'how it works' knowledge is far less important than understanding how to get it to work seamlessly end-to-end (Chadbrow, 2004). They also argue that platform technology itself has continued to become quickly outmoded in short order (i.e. the decline of serial and parallel ports, versus USB, FireWire, and Bluetooth), so technology implementers have little reason to delve into the complexities of design.

The most recent information in the 2005-06 Occupational Outlook Handbook from the Federal Bureau of Labor Statistics (OOH, 2006) would tend to support this perspective: while more than two-thirds of computer programmers hold a bachelor's degree or higher, computer support specialist and systems administrators are alternatively gaining employment using a dizzying array of knowledge, education, and certifications to meet qualification requirements. This suggests that – within the field of information technology-related employment specifically – employers are primarily interested in practical, concrete computing knowledge.

This is something that two-year degree programs are delivering: computer programmers and information technology specialists alike can become certified in programming languages and new technologies quickly and inexpensively through community colleges. In addition, former graduates who are interested in changing careers or establishing a new/additional area of expertise are also returning to a 2-year community college or technical school for additional training. As an example, in a 1998 InfoWorld Compensation Survey, 76 percent of the respondents indicated they held a bachelor degree (which is reinforced in the 2005-06 OOH statistics), but interestingly only 26 percent actually held degrees in a computing-related field.

Note that these suggestions are not meant to devalue the importance of having a baccalaureate degree in any way; in fact, U.S. Census Bureau figures show that the wage earning gap between associate's and bachelor's degrees has slowly but steadily grown over the past ten years, and at present doesn't show indications of changing. For today's pre-employees, a bachelor's degree will become more and more essential to ensuring quality job opportunities and lifelong intellectual and career growth. Instead, in the final analysis, what it means is that educators are going to have to make some hard decisions about how to address the field of information technology. Traditional educational models do not appear to be working as well in a discipline like information technology that undergoes more constant, evolutionary-like change than perhaps any other in the history of education. Industry and the lure of the workforce, at least for the near future, has shifted demand towards just-in-time training models that provide education relevant to current needs, as opposed to longer-term depth-of-knowledge models. Regardless of the approach – practical or theoretical – the "big" trend says that educational institutions and the curriculum they provide in the field of computing are going to have to be more flexible and adaptive to changing industry needs and rapidly evolving technologies. Which one an educator selects is going to depend upon the size of the institution, the types and needs of the student population, and the types of careers for which educators wish to prepare their students.

Curriculum and Certification: The Dark Side or the Dark Horse?

A popular area of controversy foreshadowed during the development of this project was the decision to select professional and industry-based information technology certification preparation programs as the curricular cornerstone of facility design. It was clear early in the project that it would be critical to establish courses and a curriculum design that could be assured of maintaining a consistent body of knowledge, so the educational regiments commonly found and used today for professional industry certification exam preparation were an ideal candidate. Over the years, many programs such as Cisco, Microsoft, Oracle, Planet3, Prosoft, and Sun Microsystems (just to name a few) have developed academically rigorous, pedagogically sound curricula and training systems that have proved their value and importance in educating forthcoming technology employees. Other training curricula, while not specifically designed by the certifying authorities, have also been authorized by these agencies and have equally withstood the tests of time and consumer satisfaction.

There is a good deal of controversy about using certification as a basis for academic instruction as a result of protectionism and personal bias, fears of the unknown, and the long-held theory that industry-driven education does not represent the same level of rigorousness, quality, or depth of knowledge as academically-designed curriculum. Many schools are realizing that preparation for the IT workplace requires a three-pronged approach: education, certification, and experience (Nelson & Rice, 2001), but not everyone is so certain.

One of the finest – and ironic – examples of this controversy can be found within the efforts of the ACM Computing Curricula project, which has attempted to unite educators, define the various computing disciplines, and establish idealistic curriculum/subject matter content models for each. In the most recent draft standard, the general committee had this to say about certification:

> "Market forces impact academic programs in various ways, some of which are beyond the scope of this report. For example, in recent years various forms of certification have become popular. The term 'certification' applies to a wide range of offerings which vary in important ways. Some certifications are vendor-specific (e.g., those from Microsoft, Cisco, etc.). Other certifications are available through professional organizations (e.g., IEEE-CS, BCS) and other organizations (e.g., ICCP). In some of its forms, certification competes with academic programs. It is clear that certification is a major trend. As with

> anything else, some certifications are more respected, others are more controversial, etc. When degree-granting institutions partner with vendor-specific certification programs, academic integrity becomes an issue. The reader should be aware that such partnerships invite controversy about academic integrity and ethics. It is beyond the scope of this report to address issues related to the broad range of certifications" (ACM, 2005).

At first glance a reader sees that the authors take a cautiously optimistic approach: not dismissing the value of certification-based education, but acknowledging that the controversy may exist. However, the writing team for the Information Technology volume – a group which openly acknowledges its intent to represent only four-year undergraduate programs in information technology, and a field which represents the greatest component of computing-related industry certifications – took an ironically less-subtle approach to the topic in their volume of the project:

> "The committee acknowledges the value of vendor and industry certifications, and encourages students to pursue them as they see necessary. However, we do not believe that academic credit should be offered for completion of such certifications, nor for training exclusively designed to prepare for these certifications unless it also covers all relevant learning outcomes defined in this document.
> Reasons for this position should be articulated. Many institutions offer certification training but do not have regional accreditation. Most certifications are practice-oriented and do not focus on the underlying theories and concepts. Additionally, many certifications are specific to a given vendor and are very narrowly focused. They therefore usually do not meet the learning outcomes defined in this document" (ACM, 2005).

The irony comes from that fact that the ACM Computing Curricula ultimately became the basis for this project's curricular foundation, into which certification preparation courses were inserted to ensure subject matter coverage.

The reality is that, like *all* educational programs, quality can not be defined by the title of the course, nor the name of the persons/organizations that prepared the curriculum. Rather, it is the manner in which the course is constructed, the experiences afforded to students in a safe academic environment, and the way in which the materials are prepared and taught to students. Educators at all levels are aware of the simple fact that even when two identical classes with identical syllabi are presented by two different instructors, the method of delivery, quality of instruction, and even at times the content can and does vary widely between the two. Instead, protectionist stances against industry certification can damage the ability to independently assess them as a viable option. At their core, certifications presently represent a combination of current, stable sets of theoretical and technical information, combined with the most advanced and current technologies on the market. In addition, certifications can (and in this project will be shown to) correspond extremely well to most of the experiential *and* theoretical subject areas that educators value, including those defined in the ACM Computing Curricula. They are universally consistent across political, social, geographic, and academic boundaries (which unfortunately seldom can be said for traditional academic coursework), and the certification preparation/training programs themselves generally define – by either the certifying bodies themselves or their authorized trainers/curriculum vendors – a common set of equipment and supplies that are necessary to provide both the theoretical and experiential components necessary to successfully learn the subject matter and prepare for the corresponding exams (Mason, 2003).

There are other allegations in the Computing Curricula which have been charged against certification as well: institutions who offer these types of courses may not have recognized regional accreditation, and that some forms of training may be solely geared to ‘passing the test’. Institutional accreditation is an issue independent of course materials, however, and issues test-driven education are potential realities in all educational levels and fields of study, suggesting the these insinuations are nothing more than diversionary tactics. Moreover, vendor-specific certification preparation regiments have been demonstrated time and again to hold intrinsic academic and experiential value when the certifying body either maintains significant market share and is representative of the leading edge technologies (i.e. Microsoft, Oracle, Cisco, Sun, Red Hat, etc.), or maintains a vendor-neutral stance and focuses on the technologies alone (i.e. CompTIA, Planet3, ICCP, SANS, etc.).

Two final ironies arise from this situation. The complaints most often made by the corollary side of academia – employers, businesses, and other industry representatives – is that certification does not necessarily go far enough to ensure that certificate holders have not just theoretical and 'textbook' knowledge, but real-world implementation skills as well. The other irony comes from the writing committee itself which suggests that:

> "At the highest institutional level, the administration must support faculty professional and departmental development activities. Such activities may include consulting work, professional society and community service, summer fellowships, obtaining certifications and professional licensure…"

The statement brings into question the validity and pervasiveness of the first argument: certification is an acceptable means of professional development for faculty, but it is not an appropriate "academic" pursuit for the students they are teaching?

In the interest of fairness, and if one looks at certifications historically, naysayers are not entirely incorrect, either: the basis of their arguments stems from the feeling that the skills and knowledge required for passing the certifications does not match the level of academic rigor found in "traditional" academic coursework. These critics are, by and far, absolutely correct when one considers the early days of certification design and academic materials. When vendor-specific and vendor neutral certification first came to the forefront in the mid 1990's, many offerings were put in place to serve as cursory demonstration tools that a person had some knowledge of a tool or technology, and were more often than not driven by the prospects of profitability and increased market share, rather than quality or the predictive validity of a person's skills. The educational tools made available by textbook vendors and training centers reflected a similar attitude, such that the materials to prepare seldom included little knowledge beyond the minimum necessary to 'pass the test'. This process made a dramatic about-face in the late 1990s following the collapse of the 'dot.com' economy and the influx of unemployed technologists onto the marketplace. The market saw a surplus of both educated *and* certified individuals, and left many employers and human resource managers scratching their heads from being unable to properly identify (at least on paper) the truly skilled and certified from merely the well-studied and certified. Poorly designed certification that was supposed to provide distinctive credentialing, ultimately resulted in creating the same situation with college degrees before the days of certification.

Since the early part of this century, however, certification testing and objectives have undergone dramatic transformations that have enhanced their credibility and strengthened their academic rigor to rival (and some may suggest even surpass) many 'traditional' academic courses. For the certification exams themselves, skillsets originally evaluated using a small bank of multiple-choice questions repetitiously recycled for five or six-year spans are now continuously upgraded, revised, and expanded to match an ever-chaining body of knowledge, as well as being steadily supplemented or replaced with hands-on components that require testers to validate their knowledge in a practical hands-on environment. The design of exams, once done by a handful of corporate-employed subject matter experts, are now regularly conducted with or even directed by pedagogical experts, college professors, professional industry trainers, and employers with vested interested in the use of the tool. Skills once defined by what a certifying organization's employees thought people should know are now compiled by experts from throughout industry and academia. Today's certification is seldom passable by the well-read alone, and testers must be intimately knowledgeable and skilled with both the theoretical and practical elements of the technologies relevant to the exam. The foundation of knowledge that testers are expected to have academically prepared for continues to broaden as the requirements to successfully function as employees in the field become greater and greater.

Which Came First, the Technology or the Problem?

With a large number of technological solutions often at one's disposal in today's world, the question most often posed by a designer is: how can someone possibly select the right technology for a specific problem? While this is certainly the most common format of the question, the problem lies in the fact that the question begins with selecting a technology, not identifying the facets of the problem.

Facility design and maintenance is a multi-faceted problem that often brings together many elements common in academia: pre-defined physical spaces, funding, constraints on faculty time, addressing the divergent needs of different types of students (i.e. traditional students vs. adult learners, newly-educated vs. continuing education), existing infrastructure limitations, etc. To simply select a technological solution that is a 'best fit' to a particular situation invites disaster, since any solution almost always requires an investment of time and labor that could be easily wasted when what appears good on paper does not ultimately work well in the classroom. Instead, program coordinators need to start making intelligent business decisions about what problems exist, and what should be done to resolve them.

On the other hand, to suggest a *business* approach to an *academic* problem is to invite additional controversy. Academics generally abhor the idea that they may have to run their operation like a business, and faculty may intentionally attempt to distance their operations from the business objectives of their institution. In the most recent 2005 InfoWorld Compensation Survey, one contributor's statement says it all:

> "Management won't reward and recognize IT's value until the working ranks start recognizing its value. The IT staff needs to be challenged by better understanding the business they're supporting and asking how they can positively impact the business solutions through the appropriate use of technology."

While this quote relates to industry, replacing the word 'business' with 'student' and 'staff' with 'faculty' in the preceding statement makes it equally applicable.

To properly develop any system, including an educational IT facility, management must begin by properly identifying what problems exist within the organization; unlike industry, where the problems are generally provided by management, and IT is expected to find the solution, in academia faculty must play both roles. It becomes faculty's responsibility to identify the unique needs and issues of their students (not just students in general), and find the solutions to meet those needs. Of course, this means that the solution for one computing program is unlikely to be the same as another.

There have been a variety of published articles over the years that have demonstrated specific solutions to the problems of implementing and maintaining laboratory environments for information technology training (LeBlanc & Stiller, 2004; Cherry, Phillabaum & Valero, 2000). The problem is that these types of articles don't address the uniqueness issue inherent in each institution's operations. Some examples include:

- The use of inexpensive PCs as routers using freeware software (Yoo & Hovis, 2004). While certainly a cost-affective solution for generalist router concepts, it ignores situations where instruction requires more specific education, such as the Cisco IOS or proprietary routing and switching protocols like ISL, TACACS+, or EIGRP.
- Remote access to laboratory networks (Yoo & Hovis, 2004), or network simulators in lieu of actual ones. What about experiential learning, however? Both of these solutions are exceptionally convenient for students who are unable to regularly complete their work on-campus, but neither simulators nor remote access are substitutions for physically interacting with the hardware and environment that students must have mastered to function in industry.
- Using hard drive protection software to prevent changes to drive images (Weeden, Scarborough & Bills, 2003). An excellent alternative to frequently re-imaging hard drives, but what about situations where students cannot complete their work before the next class begins? Convenience on the part of the facility is not always convenient for the student.
- Student-managed backup scripts for saving work (Weeden, Scarborough & Bills, 2003). This solution involved students running scripts which uniquely archived their work on a Linux operating system to a centralized server; students could retrieve their work whenever necessary and continue where they left off. An excellent solution for Linux machines, but there is not a similar capability with

Microsoft server operating systems, other than backing up the entire contents of the installation partition (which may not be a viable solution).

- Multi-booted operating systems (Belles & Miller, 2001). While this method supports multiple students on a single machine, one must consider the catastrophic consequences were one student to accidentally reformat the partition containing another student's work.
- Removable hard drives (Belles & Miller, 2001). While eliminating the problem of multi-booting, you introduce a new variable that a removable hard drive can be more easily physically damaged (introduction to unforeseen magnetic fields, accidental droppings), resulting in similar consequences.

As demonstrated above, technology for technology's sake may serve a particular operation's needs, but it can potentially lead to unexpected or unanticipated surprises. It is vital that decisions be made independent of personal preferences, comfort zones with technology, or foci on cost-effectiveness and simplicity alone.

Utilizing the Operations Maturity Model to Evaluate Progress

The development of an IT training facility is about more than ensuring student learning – it is both an organizational process and an ongoing responsibility that is indistinguishable from those processes and responsibilities performed by any company CIO or network administrator. A program coordinator and facility designer oversees a network that must remain usable and functional for a diverse set of users and employees (i.e. students), albeit perhaps with less practical control. In a learning environment the reigns normally given to a traditional network administrator must be necessarily loosened in the name of student exploration and learning. As such, however, this means that the process for developing an IT operations facility can also be measured in much the same way that one measures *normal* organizational facilities.

The Operations Maturity Model (OMM) is a derivative of the Capability Maturity Model (CMM) first introduced by the Carnegie Mellon University Software Engineering Institute in 1993. Defined succinctly, the CMM is a framework for defining and achieving the most efficient means of implementing a software development process for an organization. The OMM utilizes a similar framework; however, its focus is on defining the most efficient means of obtaining operational maturity and optimizing the value of the IT infrastructure within an organization, including technology deployment, implementation, and maintenance (ExtraQuest, 2002).

Although the OMM is always implemented slightly differently by each organization, it is generally defined by five levels of progressively improved 'maturity' within an organization (iStructure, 2003; ExtraQuest, 2002):

Level 1: Informal. This level represents the minimalist organizational structure. Architectures are undefined, operations are handled manually, processes are non-standardized, and documentation is virtually non-existent. Reliability is usually low or very inconsistent. At this level the reliance on human interaction is high, making human errors correspondingly more likely.

Level 2: Controlled. This level represents that at which the majority of IT operations exist. Procedures and processes defining workflow are in place, and documentation is usually in place, but there are still problems with meeting the documented expectations. Standardization may be attempted, but is unlikely to be fully realized. Services levels become more consistent, but strategic planning and consistent automation are still out of reach.

Level 3: Automated. This level represents where more companies are successfully transitioning to, primarily as a result of better, more affordable technologies that can be easily implemented. Standardization or other hardware cost-reduction efforts have been put into place, the majority of the tasks have been documented and automated, and system failures become less frequent with better recovery strategies. There are still weaknesses in planning and implementing the proper support structures for new technology investments, automation routines are still arduous tasks to initially develop, and in-house skill levels are difficult to maintain in the face of ever-changing technologies; hiring and human resources costs (including training) are escalating, sometimes offsetting the savings gain by improvements in other areas of the operation.

Level 4: Instrumented. This level represents the highest practical level that IT operations are likely to ever attain. 99.999% uptime (less than 6 minutes of downtime per year for 24/7 availability) has almost been attained, and failures that may occur are anticipated and well-prepared for. Management is in complete control of the operation environment, and can readily adapt to change. All processes have been automated, are predictable, and are completely supported within the operation; staff are able to focus almost entirely on continuous quality improvement instead of daily routine. Ongoing costs are readily managed, and budgeting can be accurately predicted and assessed for the future. To reach this level, costs should be expected to escalate significantly from Level 3, making this level very unlikely to achieve due to ever-changing organizational needs and the technologies necessary to meet them.

Level 5: Continuously Optimized. Automation and uptime are essentially perfect. IT systems are regularly adjusted, fine tuned, and enhanced *without* human intervention – the system is entirely adaptive. Technology and the operations drive all changes within the system, instead of management reacting to unexpected needs. Processes are fined tuned, and require minimal support staff to maintain. To achieve this level, it is suggested that the only means to do this are if:

- You have strong champions of technology within all facets of the organization.
- The business operates in a relatively stagnant industry.
- All software and hardware systems utilize artificial intelligence or other sophisticated heuristic algorithms.
- An organization is willing to invest significant financial resources.

In the final analysis of the framework, it is not possible, reasonable, or in many cases even practical to attempt to reach Level 4, and certainly Level 5 of the model. The overall model is however, a method for evaluating what an organization does operationally, if it does it well, and if there are areas within a company's IT operations that can improve service levels (within reasonable and attainable parameters). Moreover, it is reasonable to use this model to evaluate specific subsets of an IT operation and determine if those areas are more likely to achieve a higher level (data redundancy, archiving, and backup is one such possibility).

Education has always been intensely focused on evaluating the learner and his/her skills and knowledge. However, as education inevitably becomes an increasingly more competitive and service-oriented industry, so too will educators need to evaluate their own processes, systems, and mechanisms (i.e. operations) for providing that education and training to students. Using this model early in the process can give designers and/or information technology educators a tool for better assessing their learning environment and make changes that mature their programs both academically and technologically.

Summary of the Topic

Although there have been many efforts both historically and presently to develop solid curriculum models for information technology, and there have been many attempts to demonstrate one person's or school's efforts to develop an IT laboratory facility, there is virtually no prior work on attempting to establish a *methodology* for developing facilities based upon the developer's selected curriculum model.

Attempts to incorporate industry certification-based curriculum are also exceptionally uncommon, almost certainly primarily the result of most published research coming from four-year and graduate level institutions, rather that high schools, technical, and community colleges where such curriculum is far more pervasive.

Finally, the use of the Operations Maturity Model, or any other similar business maturity models for that matter, is seldom discussed or applied in academic environments. Pure research into the uses of the model are very limited, and practical applications of the model appear to be generally limited to IT consulting firms or information technology infrastructure systems designers and experts.

Contribution to the Field

This project will hopefully, in some small way, contribute to a greater development of methodologies for laboratory design, as opposed to the publications that have focused only on individual approaches. Methodologies in academia are generally prefaced around the idea of improving the pedagogy and andragogy of students; little attention is given to the specific operational aspects of running a classroom. Additionally, the use of business models in classroom operations are sometimes seen as an affront to the intellectualist endeavors and ideals generally present in higher education. This project should begin to dispel the myth that business models don't work in academic environments, by showing how well the OMM model fits into laboratory operations. There is also the belief that there is no single method of creating facilities. Although it may be true that there is no single implementation scheme, it is hoped that this project will show that a common method of approaching a design can be developed.

Chapter 3: Methodology

Life-Cycle Model Used

This project was initiated using a standard systems development life cycle (SDLC) Waterfall Model for the developmental process. As appropriate, this project methodology utilized the first four phases, building upon one another as follows:

a. Concept: The project is established to develop a functional model for developing academic multi-purpose information technology laboratories. Because a project of this nature has not been formally tackled in the past, it was necessary to spend additional time in the design and codification phases to generate a workable result.

b. Requirements: The project scope was necessarily refined and parameters established for the types of activities, level of training, pre-existing environmental conditions, etc. that would be used in creating an objective model. The model would not be entirely functional for all lab design environments that might be encountered, but could nonetheless serve as a guideline for future designers who will certainly encounter new pedagogical, andragogical, and technological changes.

c. Design: Based upon the requirements defined, an analysis of software, hardware, and instructional requirements was completed to find common course requirements and logistical crossover points that would allow labs to service multiple information technology training demands.

d. Codification: A correspondence to the Operations Maturity Model was established that identified progressively more sophisticated technologies and strategies that could be implemented dependent upon any financial, human resource, or facilities limitations that might exist in attempting to establish a new lab environment. Testing, installation, maintenance and retirement phases of the process were not addressed as part of this document.

Requirements

There were numerous issues to consider when developing or designing an instructional facility, but perhaps one of those most challenging was how the space was actually going to be utilized by various instructors, dependent upon what courses are taught, personal teaching preferences and styles, equipment installed, and unforeseen changes for the future. In many instances of technical education, the types of technologies are known and relatively stable, but in the field of information technology equipment types, sizes, and even the way fundamental core technologies operate can change with the passage of just a few short years. To this end then, a set of criteria for what might be expected to occur in a classroom had to first be defined, even if the final result would be somewhat artificially contrived.

The first question asked: what is going to be taught/done in these rooms? Although there are numerous instructional approaches, even a cursory review of common college and university curriculum will identify a simple core set of technologies students must understand within the field of information technology. This does not, however, provide the guidance needed for some attempting to develop their *own* curriculum and associated laboratory infrastructure.

The initial development of a standardized lab design must first accept some form of standardized curriculum that will provide a structural identity for the types of courses that will be held in the facility. To this end, the most logical selection was the ACM Computing Curricula 2005, a result of a joint task force between the Computer Society of the Institute for Electrical and Electronic Engineers (IEEE-CS) and the Association for Computing Machinery (ACM). This work embodies the most recent and comprehensive set of curricular expectations for all areas of computing education, and was one of the only comprehensive documents to specifically address Information Technology at the time this project concluded (October 2005 draft release). Although there are numerous opinions and controversies that may arise regarding the specific subject matter and depth of study for each area, the Computing Curricula document represents the laborious work of hundreds of subject matter experts, instructors, professors, administrators, and guidance by the two largest representational bodies of computing.

Appendix A contains a brief introduction to the Computing Curricula and an outline of the computing topics and relative expected performance capabilities of an individual graduating from a college with a four-year degree in information technology. (Please note that the intent of this document is not to present the entire efforts of the ACM task force. For a complete overview of the Information Technology volume of Computing Curricula project, visit the ACM Special Interest Group for Information Technology Education [SIGITE] at http://www.sigite.org/.) It is important to point out that, for the purposes of this project, not all components of the Computing Curricula were selected for fulfillment:

- Some of the components are exceptionally academic, implying that they would not require rigorous laboratory training, or any hands-on experience whatsoever
- Whereas the Computing Curricula is first and foremost designed for four-year

degrees, its coverage is geared towards a much more conceptual approach to the technologies. Some of the topics simply did not correspond well to the typical objectives of high schools or two-year colleges and the populations they serve.

- Some of the components are inherently designed only to be addressed as part of a third or fourth year of study, and as such fell outside the scope of this project.
- Some of the topics are included solely to reinforce degree programs that are using these guidelines and that may be seeking independent accreditation from various entities. One of the pre-texts of the Computing Curricula is "to provide a blueprint to create accreditable programs" (ACM, 2005) for specialized organizations like the Accreditation Board for Engineering and Technology (ABET).

For the purposes of lab design, topics were selected that represented the definitive theoretical knowledge, practical skills, and experiences graduates are likely to encounter upon graduating from a two-year information technology degree and entering the workforce.

Design

The design issue in this type of experimental approach can quickly become awkward and require numerous revisions to discover a workable solution. The author opted to begin simply with establishing a list of supplies and equipment that best correspond to the types of instruction outlined. Given the broad descriptive language of the Computing Curricula's suggested courses, the wide array of equipment available, and the diversity of approaches that an instructor may take to provide training for a particular subject area, however, this proved to be exceedingly difficult to do. To rectify this impasse, industry certification preparation guidelines were selected to fulfill the subject matter requirements.

To do this successfully required a three-tiered approach: the author first needed to identify what certification preparation courses and other generic coursework could be reasonably accomplished in a two-year course of study, then identify and verify what subject areas of the Computing Curricula would be covered by selected certifications, and finally identify what types of hardware and other facilities would be required to facilitate the essential instructional elements of each course.

Whereas IT certification is primarily a tool for identifying and codifying specific subsets of knowledge to prospective employers and clients, identifying the popular and valuable certifications required the author to only look to current job ads, posted career opportunities, and certification magazines (Dice.com, 2006; CertCities.com, 2006; Monster.com, 2006; MCPMag.com, 2006). From this popular array of selections, specific certifications (Appendix B) were chosen for their ability to reasonably provide training and instruction at the college freshman and sophomore levels, and their clear correspondence between each exam's set of knowledge domains covered. There were many popular certifications that were considered, but many were duplicitous to one another, were poorly represented in job openings, or simply fell outside the reasonable level of knowledge expected/possible by students in a two-year program of study.

Next, two matrices were established. The first matrix (Appendix D) served to readily identify and cross-reference the knowledge areas of each certification preparation course (Appendix C) with each of the knowledge areas of the 2005 Computing Curricula.

For many of the certifications, their expectations and specific subject matter covered was carefully defined and explicitly explained by the certifying body, allowing for simple point-to-point relations between the certification and the Computing Curricula. For some of the elements, however, it was necessary to default to the author's personal experience to assess the correlation between these two seemingly disparate approaches to information technology education. Given the author's personal experiences in teaching certification preparation courses, becoming certified, or developing curricula in most of theses certification areas, he was able to assess what (or if) certain areas of the certification process accurately matched with the Curricula. Interestingly – given the careful selection of appropriate certifications – every curricular area could be fulfilled (although a proper match in the Curricula's suggested number of hours could not be readily established, and is outside the scope of this project).

The second matrix (Appendix E) was created to analyze what types of equipment would be necessary to provide instruction in a classroom commensurate with properly preparing a student for the skills and knowledge required to successfully pass an industry-driven certification exam *and* fulfill the expected knowledge areas of the Computing Curricula. This process was much more concretely defined, as an innumerable collection of self-study and instructor-led books, textbooks, lab manuals, instructional aides, etc. exist to guide this analysis. A brief list of some of the sources used follows the matrix.

The second matrix presented several problems, the first of which presented itself as identifying and filtering out the key components that would serve as fulcra for the design and structure of a laboratory environment. The list was pared down to three primary considerations facing the flexibility of design:

- **Location/Locale.** Although PCs are generally expected, ancillary equipment like routers, switches, rack-mounted components, etc. are equally likely in an IT instructional environment. The first concern became the portability of equipment, as well as how the equipment itself was to be used. Two components were selected which served to most likely be associated with this problem: Rack-mounted equipment and PC hardware diagnostics and troubleshooting. It was presumed that rack equipment would be very difficult to transport amongst multiple labs, and that its typical correspondence with server room design training would necessitate a special place in any facility; likewise, labs involving the creation or dissection of a personal computer could have cascading and potentially disastrous consequences in situations where these types of hands-on labs where conducted on the same machine(s) necessary to provide, say, programming classes, that immediately proceeded it.
- **System Manipulation.** The ability to easily and readily modify a system to meet a particular training need is critical in a flexible laboratory design. In a modern network, one of the most labor-intensive processes is the installation (or imaging, if such software is available) of a computer's operating system. Although in the final analysis there are several options considered with regards to handling this process in a lab environment, nonetheless the process itself remains a vital operational aspect of each and every laboratory room.
- **Network Protection.** The third and final consideration was guided by the simple principle that, in an information technology classroom environment, the potential

for IT-related problems to occur is extremely high. The nature of many classroom procedures lend themselves to a greater likelihood of installing services that might not otherwise be desirable on the campus network (such as alternate DHCP, DNS, and Active Directory domains) or conducting activities which intentionally introduce high levels of risk (such as security labs involving packet sniffers, viruses and worms, Trojan horses, and spyware). In these situations there is a strong desire to filter incoming traffic into these facilities, and firewall or outright prevent the transmission of information out from these facilities onto the primary campus network.

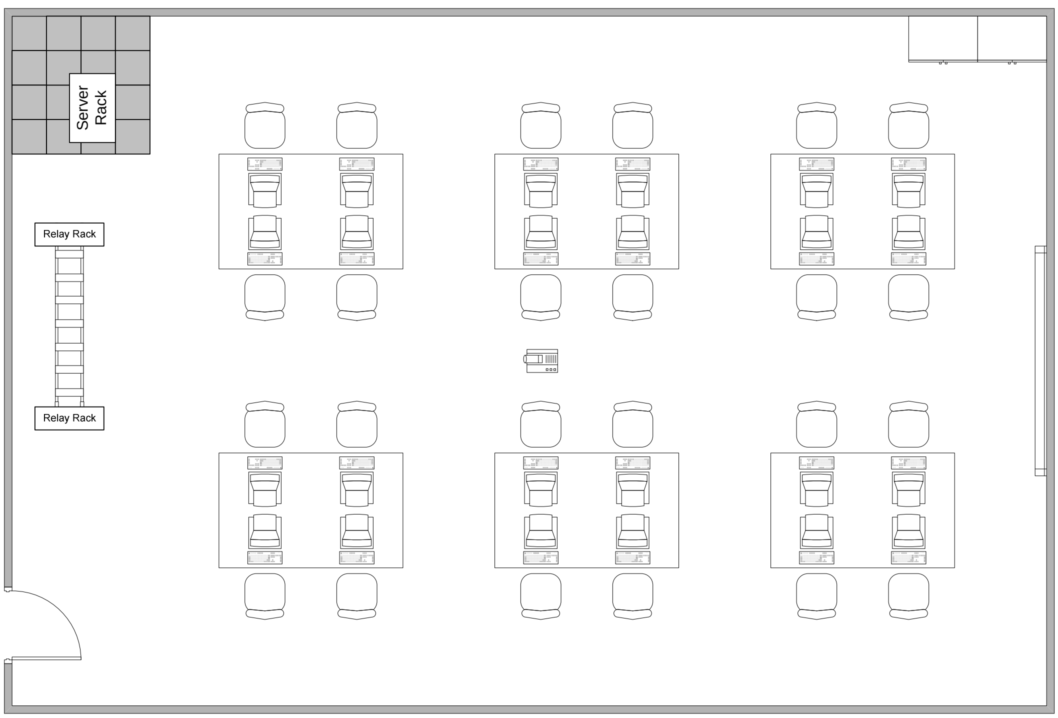
The result of this analysis generated a filtered matrix (Appendix F) which quickly identified key courses, and began shaping a contextual environment for the desired limited number of labs. Similar courses were also easier to identify. This approach demonstrates how a simply-defined matrix eliminated the guesswork from deciding how courses might best work together in a particular environment.

Codification and Results, Part 1: Laboratory Examples

The first result of this project was to identify an example set of laboratories that would work utilizing the matrices' organizational approach, and present a set of technologies and design recommendations that might serve a newly developed facility. The matrices created suggested a facility with a minimum of three laboratory units, each configured for the unique needs of the classes assigned to them.

Laboratory 1 – PCs, Servers, Security, and Wireless

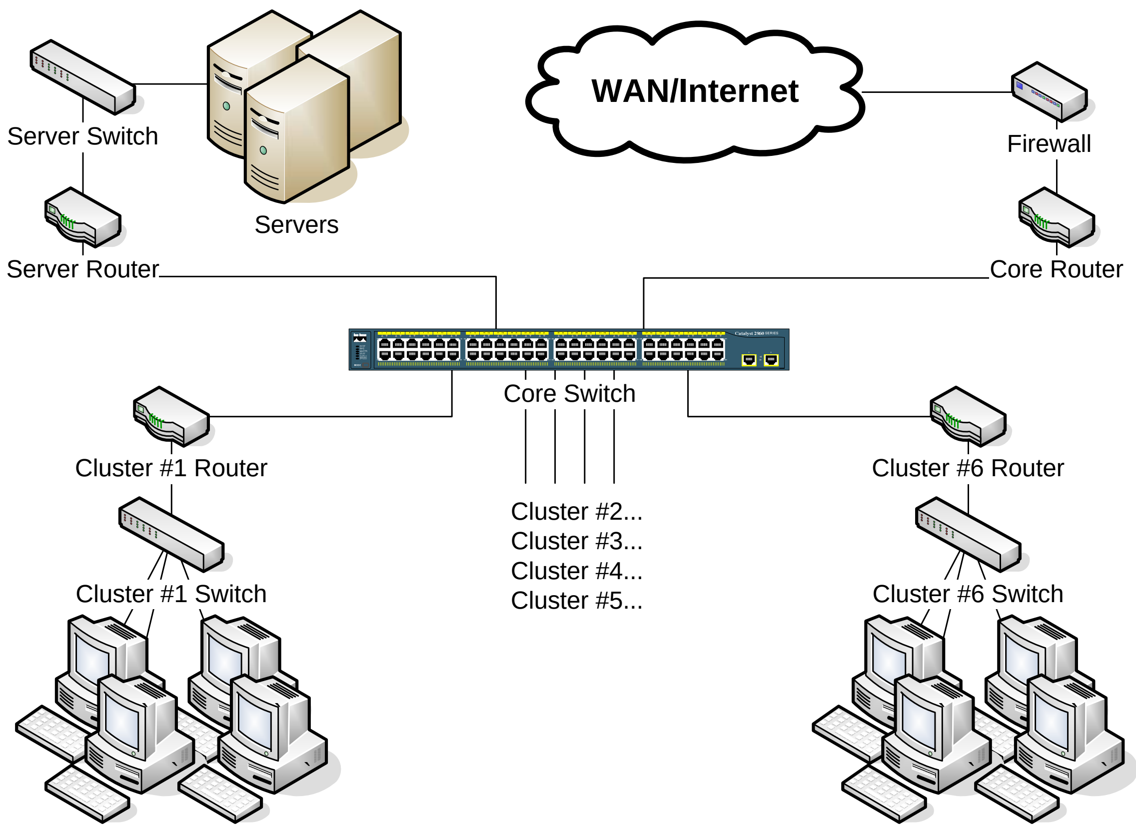
This laboratory's primary objective is to ensure that the activities held within it are retained within the room's environment, so as not to adversely impact courses which require stable environments, or the remaining campus network. The focus of this room would be to establish an environment in which anything in the room is *unsecured* and *disposable* – operating system installations, applications, and even the hardware itself are allowed an increased likelihood of failure. The possibility of corruption or damage by means of internal activities or external attacks would be high, making all forms of information in the room vulnerable to loss.



The layout presented above is an example of a clustered approach, in which each 'island' of four computers represents a unique, separable component of the network, in cases of security concerns. Each PC would be configured with multiple operating systems (Microsoft and Linux, minimum) using a hard drive imaging system, but would be otherwise devoid of applications and other programs commonly installed on campus computers.

Open spacing around the systems is supportive of computer maintenance and repair functions, in which easy access (up to two persons per system) is possible when positioned in front of and to the side of each system. A ‘mini’-server room is also created (top left) to include an elevated floor, rackmounted servers, cabling, and mockup plenum-based HVAC. The area serves the multiple purposes of a demonstration unit, security for classroom servers, and housing for a rackmounted instructor workstation accessed via wireless keyboard, mouse, and pointer.

The relay racks are designed to hold switches, routers, and firewall devices for easy access by students. This area provides the independent termination point which connects this room to the rest of the world via a dedicated connection that is terminated *outside* of the campus firewall, and which maintains its own independent firewall system.



The logical layout of the room is relatively straightforward. In this example, an entirely switched infrastructure is maintained using a simple hierarchical backbone. Each cluster is independently routed and switched, so students (and instructors) can carefully regulate the flow of data to and from each unit, and maintain up to seven independently functioning networks within the room. By providing this flexibility, it also allows for security experimentation in which different clusters can be designated as 'black hat' (attacker) or 'white hat' (defender) networks representative of a larger network environment.

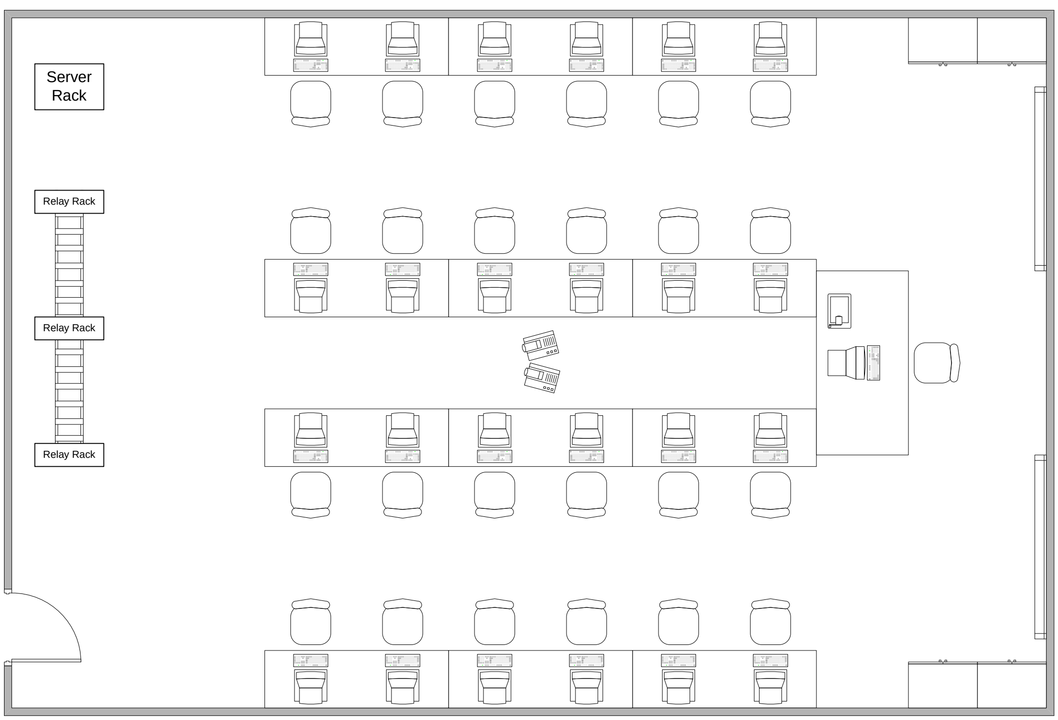
A remaining – albeit unusual – consideration for inclusion *very* early in the process (prior to construction, if possible), is the construction a Faraday cage around the room. The idealized Faraday cage is a solid conductive shell that encompasses all six sides of an environment and is connected to earth ground; its purpose is to absorb and dissipate electromagnetic signals, preventing them from both entering and leaving the enclosure. In practical implementations, the cage is frequently created using very thin copper sheeting that is applied like wallpaper to the room, or using an extremely tight metallic (usually copper or brass) mesh of wire that can be easily placed behind drywall, in poured concrete or wood subfloors, and above ceiling drywall or tiles. At first glance, the Faraday cage sounds like an inane attempt at absolute security, but its importance cannot be emphasized enough when it comes *not* to security, but to providing instruction on wireless technologies.

In order for students to understand and experience the variety of different electromagnetic interference patterns, affects upon transmission quality and stability, and the ability to explore all of the frequencies available to information technology within the RF spectrum, it is vital that you create as sterile an environment as possible from surrounding interference. As a matter of course, however, in most buildings one can now find wireless access points being widely deployed for organizational purposes, including serving as replacements to existing wired infrastructure. When training students on wireless, it is seldom as simple as temporarily disabling these access points to conduct class. Equally so, the relocation of these classes to a non-wireless location has also grown increasingly difficult to do as wireless signals now permeate all walks of life: schools, churches, small businesses, large enterprises, and even most residential neighborhoods. The practical solution is not to keep moving the training, but to artificially create an environment where students can understand what could happen to a wireless signal, instead of having to constantly deal with does happen to their signals in a non-protected environment.

Classes/Certifications Supported: CompTIA A+, Server+ and Security+, CIW Security, Planet3 CWNA.

<u>Laboratory 2 – Cisco, CIW, and Operating Systems</u>

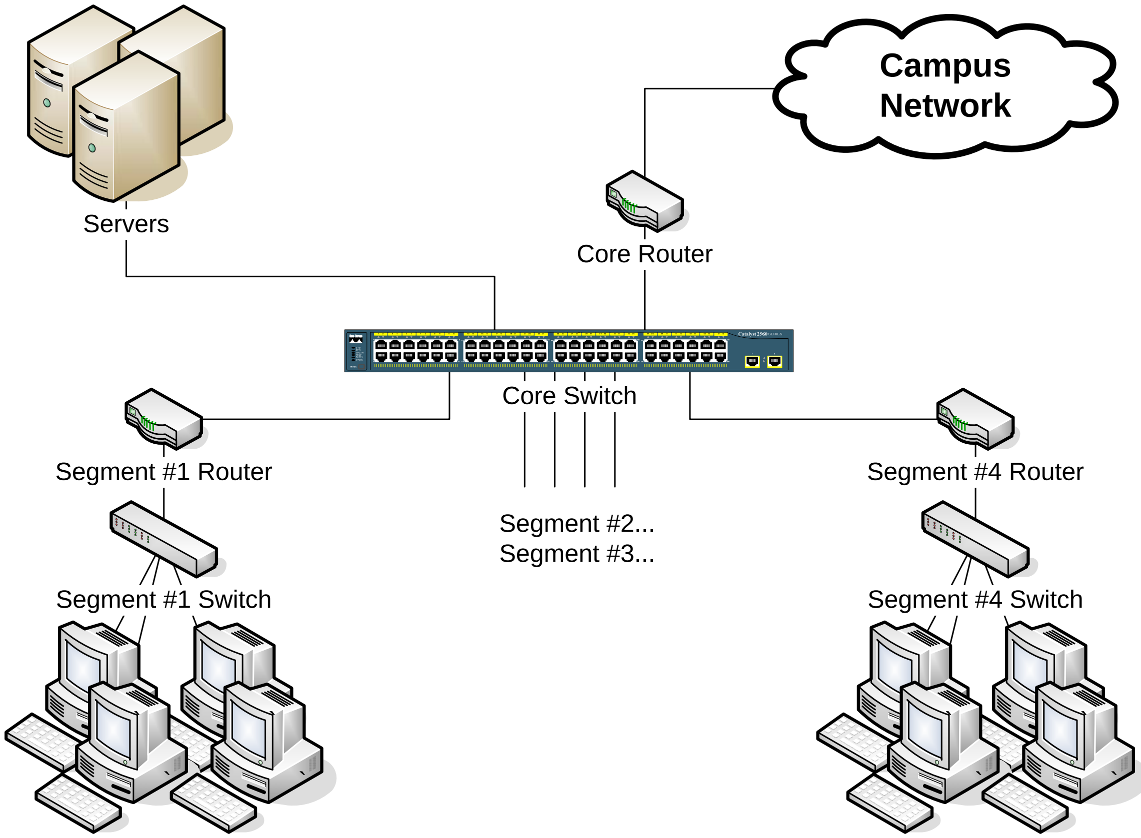
This laboratory's primary objective is to ensure the undisturbed preservation and maintenance of a student's operating system installation as vital to a student's success. The focus of this room would be to establish an environment in which its content are *protected* and *isolated* – operating system installations, applications, and the hardware must all be unmodified by non-IT students, and even institutional IT support. In an operating systems course, for example, the training and ongoing work of a student is centered on the actual operating system installation.



The layout presented above is an example of a segmented approach, which is slightly different from the clustered approach used in Lab 1. The room is divided into four segments of six computers, each segment capable of representing and functioning as a separate network. Like Lab 1, there will be times when each segment will need to operate (or least apparently function) independently for certain networking purposes, such as demonstrating WAN routing and certain communications protocols.

One of the distinct physical changes here is that students are better able to interact with one another – seats are more closely spaced so students can equally more closely interact with one another. In operating systems instruction, for example, it is exceptionally common for one system to function as the client, and the other the server; in a later assignment, the roles may be reversed. This facility is intended to better encourage and support the group interaction that is more likely to occur in some types of classes.

The server rack in this room is strictly for containing support equipment, thus playing a decentralized role in the physical layout. Conversely, there is a greater likelihood for more networking equipment – especially routers, switches, and multiple connections for each computer (Ethernet and serial). As a result, an additional relay rack and wiring ladder is placed in this room.

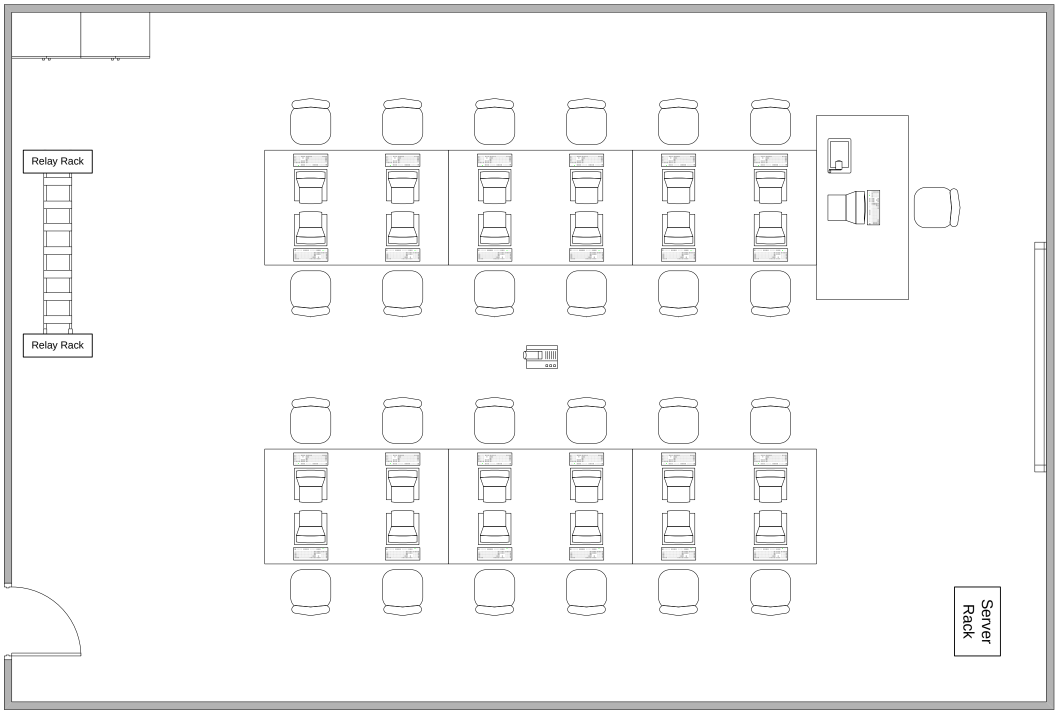


Another key difference in this lab lies in its logical structure: this laboratory remains connected to the campus network, even though traffic will be routed and filtered to ensure that certain types of traffic neither ingress nor egress from the room (i.e. DHCP). Computers will be configured with dual NICs and serial connections, and all wiring termination points for the PCs will be in a series of racks containing hardware necessary for advanced networking training (i.e. Cisco). Multiple connections allow PCs to function properly when configured to act as routers (as may be found in operating system-based Routing and Remote Access instruction or other WAN technologies, for example), while direct serial connections provide a simplified mechanism for students to configure routers and switches 'at the source', without having to physically move their PCs. This lab will be configured to support multiple operating system installations through whatever technological means is most viable (i.e. removable hard drives, external drive systems, virtualization, etc.). In addition to the student's specific installations, a generalized operating installation will also be maintained in this room for the purpose of providing a stable application and programming environment (i.e. Java, C++, Macromedia, and Web development courses).

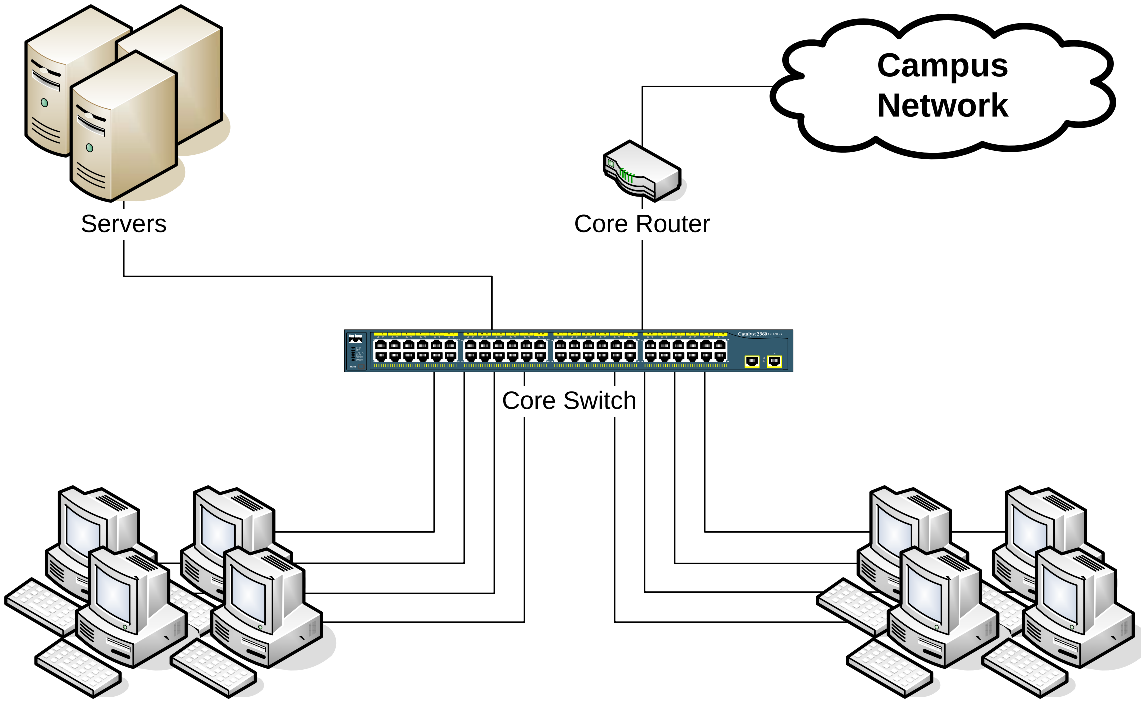
Classes/Certifications Supported: Cisco CCNA, ComputerPREP CIW, CompTIA Linux+, Microsoft MCSE.

Laboratory 3 – Networking, Programming, and Programs

This room would most closely serve as the 'jack-of-all-trades' for an IT program. This laboratory's primary objective is to ensure a solid environment (utilizing an institution's IT support staff) in which classes can be conducted without concerns such as network access, unusual hardware, or modified PC configurations. The focus of this room would be to establish an environment in which its content are *stable* and *supported* – introductory programming and networking classes, for example, need a stable environment in which students do not have to concern themselves with IT contextual issues that fall outside of the subject they are immediately studying.



The physical environment is reflective of an effort towards simplicity, cost effectiveness, and manageability. Two standard rows of twelve computers each are wired to a single classroom switch located on one of the relay racks in the back room. Although the primary focus of this room is to support non-network related courses, the relay racks are sufficiently affordable that they are included in the room to support any future needs or unforeseen applications. The PCs are configured in a traditional fashion with a single installed operating system, a single NIC, and regular connectivity to the campus network. Overall, the facility is as such that its simplicity of design virtually eliminates the likelihood of unusual problems or misconfigurations, and thus ensures stability of the environment for courses that require this attribute the most.



The logical layout of this lab is equally unimpressive. With a centralized switch and a single router to provide traffic filtering, when needed, this room is the low-cost leader in the entire facility. This is not to say, however, that the room cannot accommodate future needs. By terminating all PCs within the room itself at the relay rack, there always exists the ability to add additional routers and switches, and achieve comparable functionality to the other two labs presented.

A server rack is still maintained in this room to provide localized servers, if necessary. Possible server-based inclusions may support Web publishing, training for Microsoft Project Server, or enterprise databases (Oracle, SQL Server, DB2).

Classes/Certifications Supported: Introductory programming, CompTIA i-Net+ and Network+, some ComputerPREP CIW courses, word processing, spreadsheeting, databases (Access, SQL Server, Oracle), and other applications.

Codification and Results, Part 2: Operations Maturity Model

The second result of this project was to identify and associate the laboratory design with a maturity model for implementing these technologies at various types of institutions, and identifying approaches that institutions can use to advance through the levels of the model.

As previously discussed, institutions seeking to develop or enhance their information technology studies programs can find great advantages from evaluating their own processes against the Operations Maturity Model (OMM). Whereas this project focused on the organizational layout of laboratory facilities, some concrete laboratory solutions are suggested – based upon the matrixed objectives, requirements, and physical components defined and identified in this study – that can serve an institution using the OMM as a means for improving their own services.

Level 1: Informal

Perhaps surprisingly (or perhaps not), most educational institutions operate at this informal level (at a minimum) in almost every academic classroom setting, in which faculty often haphazardly and independently develop educational training systems that are focused on the individual instructor’s personal experiences, tastes, skill sets, and general comfort level. It is presumed that any organization with some level of IT funding and the ability to establish any formal academic setting will minimally operate at this level. No suggestions are made.

Level 2: Controlled

This level suggests a controlled but cost-effective environment, and is the primary basis for this project: to move a program from Level 1 to Level 2. Movement to this level still requires buy-in from all faculty involved, and may be difficult in some academic environments. Some of the technical suggestions for maintaining a Level 2 operation include:

Removable hard drive systems. Inexpensive used or low-capacity drives often make good choices and do not require dramatically-modified or expensive PC systems. Removable drives can be created using lockable trays or external SATA connections that allow each student to have their own installation that can be easily protected by simply removing the drive and storing it away in a secure cabinet. These solutions are cost effective, and have the distinct advantage of not requiring more frequent or regular backups of students' work, while providing security by protecting that work from other mischievous students or even the occasional virus/spyware attack (when the drive is not installed). The downside to these systems is that they are inherently more susceptible to physical damage. An accidental drop during transportation to and from a storage unit can render a drive useless, while physical impacts and movement when/if they are connected externally can cause minor damage and data corruption over time. More technologically advanced alternatives exist, but they usually require a more significant investment to ensure stability and availability for students.

Occasional backups of data and information. Implementing inexpensive

DVD-R/RW drives for occasional backups is an easy choice. Dual-layer burnable DVD technology is still a bit expensive in terms of the media, but drive units now rival their earlier incarnations in terms of cost, availability, and ease of use. At 8.4GB per dual-layer DVD, a single disc is generally more than sufficient to back up an entire hard disk containing an operating system, installed services, a complete installation of Microsoft Office, and a variety of maintenance programs and education tools. Of course, at 45-60 minutes per system to backup, the process will require some time be allocated by the student, teaching/lab assistant, or instructor to oversee the operation. Extra software must also be purchased, since most backup tools included with the various operating systems (such as with XP and Server 2003) do not support backing up data directly to CD/DVD media. For a class of 25 students, backing up their drives once a month per semester would require approximately 100 dual-layer disks, which can be purchased in bulk for as little as 50 cents a piece. On the other hand, the process can generate 150 hours of work or more *per class* each semester – a significant time investment for an instructor with multiple courses to oversee.

Faculty-maintained lab systems. It can be both expensive and difficult in high schools and colleges with shorter academic timeframes (i.e. two-year vs. four-year programs) to find and retain persons who are adept at understanding and maintaining and academic lab environment. Although permanent lab employees could resolve this concern, it is seldom an affordable alternative. To this end, then, it would be necessary that faculty configure, maintain, and oversee the core functions of the laboratory. Certain responsibilities, such as the backup process mentioned above, may be shifted of to work study-based lab assistants, but the quality of such employees cannot always be guaranteed, and the instructor must be prepared to carry the burden of the network alone, if necessary.

Level 3: Automated

An automated operational environment is rare in most organizations of any kind, much less in an academic IT environment which necessarily dictates often dynamic and abrupt changes in the operational infrastructure in order to accommodate changes in technology. The ability to anticipate and mechanize system failures and established automated, repeatable and well-document processes are exceptional difficult. There are some technical solutions that can bring a facility closer to this object, however. They include:

Virtualization of all student computer interfaces. Virtualization has provided an amazing opportunity in academia to utilize file-based images of entire operating systems. Virtualization takes advantage of the default high-capacity hard disks, inexpensive memory, and powerful dual core processors available in today's simplest personal computers, and optimizes the flexibility with which students can work in a lab environment, both simulated and real. The easiest mechanism for distributing "virtual computers" involves the creation of a single installation of an operating system that can be placed in a shared folder and quickly distributed to any number of PCs on a network (barring any mitigating licensing issues). Virtual PC, Virtual Server, and VMWare are readily available low-cost virtualization programs that can present these installations to students as if they were a separate, individualized PC computer. Moreover, corrupted or damaged images can be readily replaced, backups of the student's work within the image are simply a function of backing up a handful of files (and which does NOT necessarily require specialized backup software). Beyond this, virtual software supports the customization of the 'virtual hardware' for a particular lab: additional hard drives, network cards, and even SCSI devices can be 'added' to a students virtual machine with the click of a mouse.

Setting the technical aspects aside, virtualization also makes sense from a student-centered learning perspective. Traditional hardware-based environments restrict student's work to the physical location of a machine. Virtualized systems, however, can be duplicated and even transported (or transmitted home given sufficient bandwidth) so that students can work on their system from any capable PC. Moreover, by setting up a set of two or more virtual images for a student, each individual can maintain their own virtual network on a single machine.

Requirements for virtualization, however, often necessitate a significant initial investment. Practically, the physical PC systems that will support virtualization require a minimum of 1GB of memory for the host operating system, and then at least 512MB per simultaneously running virtual machine. Processors should be as advanced as possible, and even multiprocessing should be strongly considered. Each virtual image can easily reach 3-4GB of hard drive space with a default installation, and approach as much as 8-10GB in size per image with additional software installed.

Remote desktop solutions. Remote desktop solutions (such as Microsoft's

Terminal Services or Citrix MetaFrame Server) have a distinct advantage over some automated solutions – they provided centralized access and control of multiple ‘presentations’ (or server software-based PC-like terminals), and can provide this service to anyone anywhere in the world. Backups can be done for a single machine, and the number of student connections is limited only by the hardware and bandwidth available. Such a solution also eliminates the need to maintain large numbers of high-end PCs to support many traditional laboratory requirements. There is one downside, however: the cost of these systems can easily exceed $100,000 or more when fully implemented in an operational academic setting, make the investment significant enough to offset any PC hardware savings. Combine this with the fact that most educational institutions are still expected to provide on-site PC systems anyway, and the justification to administrators and other fiscal monitors is likely to be very difficult unless the need or benefit of remote access is considerable.

Automated, regular backups of all systems. Assuming virtualization is in place, backups are dramatically simplified and can be done with greater frequency over a network when given sufficient localized bandwidth. Gigabit to the desktop would be strongly recommended, as would a localized backup server, on which the data would need to traverse no more than a single switch. Put into a practical context, if one assumes each student maintained two virtual images averaging 5GB each, and that there were five courses that utilized the virtual machines with 25 students in each one, total data to back up would be approximately 1.25TB. Not so long ago this volume of data would have appeared to be insurmountable in terms of both storage costs and time to backup, but inexpensive Serial ATA hard drives and RAID controllers mean that a 2TB storage array can be added to an existing server for less than $1500. If backups are rotated so as to occur every three days per system, this process is also easily achievable using an existing (but upgraded) CAT5e network infrastructure. Of course, this begs the question: do backups need to be done every three days? The answer, however, will always be quickly answered with a resounding 'yes' by any student who realizes that their hard work is constant threatened by the potential failure of the local hard drive.

A completely documented environment. A solidly documented infrastructure is critical to ensuring the long-term stability and supportability of an information technology environment. Documentation must be maintained for the purposes of divesting responsibilities for maintenance (when appropriate), driving sound decision-making for upgrades and equipment replacement, troubleshooting and repair, and coordination of instruction and laboratory preparation amongst the various instructors who would utilize said facilities. At the core of this process, there are three areas of specific documentation that must be maintained: laboratory design and equipment specifications, and curriculum-equipment requirements, and systems baselines and ongoing system performance records.

- Proper documentation which outlines how a laboratory both physically and logically routes data is important to being able to develop curriculum that will work within that environment. Stable records of facility drawings and the wiring plant should be combined with the more dynamic records of switch/router/firewall configurations, server hardware and operating systems, installed service packs and hotfixes, and any currently running programs on the network.

- Without knowing what hardware and what configurations will be necessary to carry out student labs and experimentation in a course, it is extremely difficult to properly "place" a course within a particular facility. The matrices developed and discussed earlier in this project are representative of one method in which this can be accomplished.
- One of the most fundamental instructions in an information technology-related course is the importance of knowing how a system is supposed to perform, as compared to its current level of functionality. To not implement the same processes so vehemently lectured to students is hypocritical at best, and potentially costly at worst. Even in a laboratory, it is important to constantly be aware early of whether existing equipment is adequate to carry out the tasks expected in an upcoming academic term. Operating systems and servers should have baselines created for them when first put into production, and then again following each subsequent academic term. Routers and switches should be evaluated as well, even though sluggish performance may be less noticeable in a lab environment where the net traffic flow is likely to be substantially less than a production environment. Good planning is critical in academia, where instructional-side budgets may have to be prepared years in advance, and purchases must be anticipated for technologies that don't even exist on the market yet. It is seldom that educators are afforded the same just-in-time planning that corporate management is more likely to provide (or even encourage).

Designated laboratory support staff. This is perhaps the most difficult objective to achieve under Level 3, not so much as a consequence of locating qualified or capable individuals – but in the simple fiscal realities of academia. High schools and colleges alike find budgets continuously squeezed when it comes to dollars for support staff and operational expenditures. Couple this with academic programs like information technology that have a high 'investment dollar-to-return' ratio, and the situation becomes more bleak. In the end, the use of trained students paid on federal or state-subsidized work study dollars is one alternative, but program directors must be prepared to invest time in a continuous recruitment, training, and retention process that will never wane. Better options including establishing a combination of the aforementioned student positions with one or more permanent support positions that allow a person to transition from student to new graduate (both high school and college), with the opportunity to be employed in-field at the school, albeit with modest pay and benefits, and acquire experiences that allow them to more successfully transition from the often-perceived 'inexperienced graduate' to 'viable employee'.

Levels 4 and 5: Instrumented and Continually Optimized

Both conceptually and realistically, it is almost impossible for any organization of any size or type to realize Levels 4 and 5 in either the academic presentation or daily operations of information technology. In fact, most consulting firms that specialize in the application of the Capability Maturity Model and its variants are seldom willing to make any contractual obligation to the possibility of an organization achieving either of these levels. Of course, while there are technically not any functional barriers to achieving these levels, there are several practical ones (ExtraQuest, 2002):

- Moving from Level 3 to Level 4 is extremely time and resource intensive (i.e. attempting to achieve nearly 100% automation and 99.999% uptime or higher)
- Ever-changing business or educational requirements in technology result in an operational target that is never static.
- Level 5 is defined as attaining near ‘operations perfection’.

Such an objective should always be sought – even if they affect only modest elements of operations – but there is usually the simple limitation of too rapid a change in the field, and not enough funding to support such an aggressive objective.

Summary

The design and results of this approach demonstrate that, first and foremost, the selection of a course of study and a defined curriculum to utilize are pivotal components to designing and implementing a successful laboratory design. Without these, the intent of the laboratory goes undefined, making selections for specific equipment and infrastructure nearly impossible.

Second, the selection of specific equipment can vary widely depending upon particular training needs and individual instructional styles. Although some general guidelines can be given regarding the current state of technology and the possibilities that may be used to create a functional and successful laboratory, indeterminate variables such as the manpower available, budgeting (both one-time and recurring), class sizes, student skill level, etc. will play an important role in making a specific selection.

Third, the application of a methodology for demonstrating the successive maturity of an operational design provides an excellent mechanism for gauging the quality of an information technology environment. Although the operations maturity model presents an idealistic set of operational achievements, it can nonetheless serve to identify weakness in an environment and encourage laboratory managers to focus on the importance of a continuous improvement process.

Chapter Four: Project History

Project Beginnings

The project was initiated as a result of the author's own personal involvement in the development of an Information Technology Center at San Juan College in Farmington, NM. In the Fall of 2000, only a couple of months after he began his new career as a full time instructor at the institution, he was approached by the Dean of his school and the President of the institution to explore the costs and requirements for developing a new IT Training Center. Needless to say, this was a daunting task for anyone, much less someone who was still new and trying to learn the basic functions of a new organization. Nonetheless, the task was clear and work was diligently carried out to present a list of needs, approximated costs, and even a general design recommendation for the facilities. One key advantage during this process was the fact that the space was already pre-defined: an 'unfinished basement' of sorts was allocated for the project, below a recently constructed classroom facility, with a relatively open floor plan from which the Center could be molded from the more than 22,000 square feet of space available.

The author initially sought the assistance and advice of coworkers, but whereas they, too, had no expertise in this level of development, they choose to exclude themselves from the entire project. Another instructor within the author's department served as an architect in a previous career, and assisted in developing a set of blueprints for how the facility would be physically laid out in terms of walls, electrical, doors, and rough locations of student desks and instructor workstations.

It was around this time that the author realized that much of what was being done regarding the design of the space and the facilities within it was essentially guesswork. During this same time, the author and another instructor were working on developing the academic curriculum and degree programs that would utilize these facilities, but whereas the curriculum and courses would be new to both persons, there was no real sense of what would be required of the facilities to successfully conduct labs and other hands-on activities with students. Educated guesses were made at what would be needed by using a variety of textbooks and lab manuals (although not necessarily the ones to be adopted for the courses). This review provided a sense of what kinds of activities would occur, but the harder reality was that much of this was being done blindly.

The core of the work for this project occurred well after the facility was built and educating students. Nonetheless, over the years the facility has continued to show deficits in their ability to provide the quality of classroom instruction and training sought by the instructors. Over time the faculty have had to face the consequences of systems and designs that were put into place without understanding the interconnections between curriculum expectations and the physical environment. Even up to the time of the completion of this project – some six years after the facility was developed – additional money is still being investing, and more time spent, in an effort to further manipulate and mold the facility. Many aspects of each room have been redesigned to better serve students and more accurately support the needs of the courses held in them.

It is with these experiences in mind that the project was undertaken: taking a good hard look at what technologies are involved in teaching a particular course, finding similar courses that can more easily be meshed together, and developing facilities that are specifically designed to meet the requirements of the courses. For some observers, the project did not have the same gritty technological underpinnings as many that were presented to Regis University, but many understood the critical role that planning and design play in Information Technology architecture: both technologically and physically.

Project Management

Unlike most projects that go through this process, this one required little oversight or outside interaction/intervention to ensure progress. This project represented a combination of research and personal experiences gained by the author over the past five years to establish a model of architectural planning and design relative to a specified curriculum set. The simplest analysis of the situation is that, without having gained the personal experiences over these past years to understand what is involved in providing this type of instruction, it would have proved exceeding difficult to achieve results beyond the guesswork used when the principle was first applied.

Significant Events/Milestones in the Project

Identifying a basis for curriculum selection. The Association for Computing Machinery had not formally developed an Information Technology component to its curriculum recommendations when this project was first proposed; bits and pieces had been pulled together from various resources in 2001 to create a draft concept, but it was not until three years later than the vision was realized. Using this evolving standard, however, provided an excellent foundation on which high school, community college, and four-year institutions alike could find some common ground. The only alternative would have been to select courses based upon the author's personal perceptions, but this method would have almost certainly resulted in natural biases towards his own strengths, instead of ensuring a well-rounded curriculum.

Identifying and selecting diverse curriculum. In any program of study, a program director attempts to achieve a balance between the theoretical and practical knowledge a student must acquire in order to be successful in an operational environment. The choice for using certification-based training was difficult due to the conflicting attitudes regarding its use in academia, versus its exceptionally well-defined knowledgebases and stable learning objectives. Nonetheless, until the curriculum was selected, there was little opportunity to move the project forward.

Development of a curriculum matrix. The matrix looks relatively straightforward, but it required using a combination of the author's personal experiences and his use of resources from various textbooks, lab manuals, and training programs to understand what was expected in terms of equipment and facilities. Once the matrix was done, it was much easier to identify how many laboratories would be minimally required, and where the courses would best align within those facilities.

Discovery of the Operations Maturity Model. The author's initial advisor, Mr. Sam Conn, first pointed out the Capability Maturity Model in 2004, well after this project had begun. One of the challenges the author had was in identifying how to 'rank' different approaches to facilities design, but the offshoot of the CMM, the Operations Maturity Model (OMM), presented the perfect approach.

Project Changes

The greatest change to the initial proposal is the elimination of the deliverable that was to be "a chart, with recommendations, that identify popular, common, or affordable technologies for establishing a facility". The hard reality is that the chart would have become obsolete the moment it was created, since changes in technology, coupled with often dramatic changes in technology costs, would have rendered the suggestions meaningless in short order. Instead, the idea of using a maturity model was substituted, representing not so much a specific set of technologies, but rather a true *methodology* for using technologies to properly management a facility and implement the most current technologies (whatever they may be). This was done in a way that can hopefully provide architectural stability (both physically and technologically), long-term instructional viability, and consistent maintenance.

Of course, it was still important to at least provide some technological context, both for the laboratory examples and the maturity model. Even though the technologies may eventually be outdated for a reader, they nonetheless provide an example of how the author connected the abstract matrix approaches to a concrete implementation in a laboratory room. Likewise, it becomes important for a reader reviewing the operations maturity model to have some sense, and some examples, as to how one might progress from one level to another. The author attempted to keep most of the examples grounded in technologies that (although they may evolve) will have a strong likelihood of existing for many years to come.

Did the Project Meet Its Goals?

As an initial attempt to provide some structure to developing academic IT laboratory facilities, the project successfully met its goals. This project demonstrated that there are methods and mechanisms that exist that can allow a designer to develop such facilities without relying on guesswork or the inherently imperfect examples of another institution's particular facilities. Moreover, this project was able to successfully demonstrate that continuous improvement in those facilities is possible when good business models are used.

In so far as the author's personal goal of completing the project at the conclusion of the author's academic coursework, the result was not achieved as readily. A combination of personal incidents and other professional expectations necessitated that the project be put on hold for several years, but the outcome was a positive one nonetheless. During that time the author was able to refine his skills and knowledge regarding the design and management of a laboratory facility, and thus use that information to better establish the curriculum and course requirement matrices.

What Went Right, and Wrong

In general, because this process was less of a 'project', and more of a 'research/thesis' paper, very little technically went wrong. While it is true that the document and its deliverables evolved over time, these changes are considered to be positive movements over the course of the project.

The continuous development of the Information Technology Computing Curricula, for example, provided the author with an opportunity to see the evolution of academia in relationship to this field of study. As much as the technology has rapidly evolved in recent years, so too have the perceptions about the mechanisms and content of its instruction in academic environments. Although the most recent draft of the Curricula stills maintains a strong resemblance to its initial roots, it still exists as exactly that – a *draft*.

The author also strongly holds to the belief that this project is superior to its original proposal as a result of the inclusion of the operations maturity model, even at the cost of the technology recommendations chart. Specific technologies will inevitably evolve and/or fade over time, but proper methodologies and best practices are timeless.

Project Variables and Their Impact

There were three sets of variables that presented the greatest challenge to this project:

Technologies. Changes in technology, including those that occurred between the project's initial proposal and its conclusion, presented a constant threat to the value of the project and its applicability to future laboratory implementations. The author chose to circumvent this uncontrollable variable as much as possible by focusing on a methodology of design, rather that a project which would specify a particular implementation activity.

Curriculum. As has been pointed out in numerous instances throughout this document, curriculum plays a vital role in defining the context in which a laboratory facility will be designed and maintained. Constant revisions to the ACM Computing Curricula posed a real possibility of having to continuously modify the project matrices to match this model. However, the core subject matter including in the model has remained surprisingly stable over the years, eliminating a need to continuously redraft the results of the project.

Institutional Uniqueness. In writing a document specifically intended to serve high schools and two-year colleges on a limited budget, there existed the possibility that the end result would be the opposite: a series of unique laboratories that would be expensive and difficult to justify. In other words, the possibility existed that the proposal could fail. The variability thus came in the form of what each institution could possibly afford, versus what the end result of the project might recommend. In the end, this variability became a non-issue, as it was demonstrated that numerous information technology-related courses could be held within a relatively small number of laboratory facilities.

Findings/Analysis of Results

Because this undertaking did not have a terminus more common a traditional project, there is no *conclusion* to the problem. This project attempted to demonstrate that there are technologies and business models that, when used together, can create a platform for designing successful laboratory systems for information technology instruction. The only answer is that such an endeavor *is possible*, but it requires carefully planning and consideration of the curriculum in order for it to be successful. An actual implementation of this method, combined with a longitudinal study of the facility's functionality, would be necessary to truly prove or disprove this methods' viability.

All of this not withstanding, the methods, curriculum, processes, and maturity model are all grounded in the sound principles of education and business today, suggesting that their merging has produced an equally sound model of design. The technologies suggested in this project as means of achieving progressive maturity are also consistently considered to be reliable, proven, and – most importantly – possible in a laboratory environment. Although there are many technologies that are available that are either prohibitively expensive or unwieldy to manage, those outlined in this project are technologies which any faculty member or modestly proficient student should be able to master.

Summary

Although the history of this project proposal originates several years ago, the timeliness and applicability of its goals has gone unchanged. Educational facilities are inherent cumbersome facilities to manage, and without the proper initial design can create unending frustration for both its managers, and the students who utilize them.

The project's progression, although delayed, went relatively smoothly, with only nominal concerns regarding external changes that could have required updates to the document. As each significant milestone was achieved, the project continued to take shape and suggest that a positive result was at hand. In the end, the author believes that the project was very successful, and can be a useful tool for any organization or facility preparing its laboratory design.

Chapter Five: Lessons Learned and Next Evolution of the Project

Experiences and Lessons Learned

In some sense the project itself was a culmination of the experiences and lessons the author learned from working as a program coordinator for an Information Technology program. The project represents a blueprint for what should have been done nearly six years ago when the author developed information technology facilities for his own institution, and it is a document which can be relied upon for making future decisions when new academic programs are developed, or existing ones are expanded.

This project also afforded the author an opportunity to unite various aspects of his background: prior experiences and education in business, recent studies in information technology and enterprise architecture at Regis University, and current employment as a professor of information technology all came together to develop this project.

Some of the specific experiences and lessons learned during this project include:

- Certification in academic curriculum will *remain* a hotly debated topic, but it is a topic worthy of debate given its importance and relevance in the marketplace, and its impacts on current students and graduates alike.
- Technological change is never ending, but there *are* ways to prepare for the technological unknown.
- There *are* better ways to both design and run an academic laboratory than those set by one's predecessors, but it requires using techniques and ideas that sometimes have little to do with information technology itself.
- Technology can be both a *burden* and a *salvation*. The key to implementing

technologies in an academic setting are choosing ones that are manageable for both instructors *and* students, and that move learning forward without generating frustration or defeatism.

Possible Alternative Approaches

This project was developed with some specific decisions that intentionally narrowed the scope of the project, and in this way there were several alternatives that could have been considered. For example, the decision to use the ACM Information Technology Computing Curricula as the basis for the subject areas to be included in the curriculum could have just as easily been substituted with state-university curriculum (for the purposes of students transferring to other institutions), alternative accreditation standards (such as the Computing Accreditation Commission of ABET), or even instructor-driven subject matter expertise (in the case of institutions with limited faculty support).

Another significant alternative approach rested in the use of professional industry certification as the *guidelines* for defining specific course content. This decision was driven by several personal and professional factors, but course content could have been just as easily driven by transferability guidelines to other colleges and universities, a decision to use pre-selected textbooks and other course materials that define the objectives of a course, or even regionalized and/or industry-requested inclusions at a particular institution. High schools may be driven to use only curriculum that is readily available via online availability (i.e. Adobe PDF files and online multimedia training), in order to minimize expenses associated with textbooks and other tools that usually have a short shelf-life in this field of study.

The last major alternative option was the use of alternative business modeling techniques. Classic business models like Just-In-Time (JIT) management and Continuous Quality Improvement (CQI) efforts may have served certain niche institutions or individuals already intimately familiar with these processes. The author selected the Operations Maturity Model because it seemed to best define *levels* of continuous improvement, as opposed to other methods that simply identify cycles of a process. With the understanding that most academic laboratory environments seldom have the time or resources at their disposal to employee the other more aggressive corporate strategies, this solution seemed best; some institutions, however, may have the manpower and resources to be more aggressive, and thus prefer to implement an alternative evaluative model.

Correspondence with Initial Project Expectations

The project both met and exceeded the initial expectations. As with many projects that are presented through the Regis University's Information Technology program, initial anticipation was that, although the project would fulfill the requirements of the degree program, the useful life of the end-product would be inherently limited as a result of the rapid changes in technology itself. The modest changes to the project, however, have yielded a product which is much more universally applicable to good information technology laboratory design, and – more importantly – much more timeless because of its ability to define the methodology to a greater degree, while using examples of modern technologies that are more likely to survive the tests of time.

Next Evolutionary Steps of the Project

As the author has progressed through the project, the next area of focus that has most naturally presented itself is in the practical implementation of the methodology, and the education of instructors and program coordinators on the usability and practicality of implementing the OMM model in programmatic and facilities management. Several segments of this project may serve as ideal candidates for a set of lecture and poster presentations for information technology-related conferences, such as the ACM Special Interest Group for Information Technology Education (SIGITE), or the League for Innovation's annual Conference on Information Technology (CIT).

Academic instructors can become entrenched in the theory that they are subject matter experts who should not have to be bothered with the managerial aspects of classroom instruction. While this may be much more true for humanities, mathematics, and social sciences, the hard sciences – including information technology – can always benefit from learning new ways to mature both the facilities and the classroom experience, and how to apply new or existing technologies to advancing the sophistication of classroom management solutions, not just industry ones. The author has found in his own experiences that there is a seductive trap of focusing on ‘teaching the technology’ that is easy to fall into, without noticing whether best practices are truly being applied to the operations.

From a more pragmatic perspective, it would also be useful to expand the matrix to include greater specificity of equipment, systems, and supplies, as well as a greater inclusion of additional certification courses and traditional academic coursework that is relatively consistent across institutional types and/or state-by-state standards. As the

ACM Computing Curricula for IT matures, this option will become increasingly feasible. Of course, the curriculum is more likely to be implemented by traditional four-year colleges and universities rather than high schools and community colleges, but the gap between the divergent missions of K-12, two-year, and four-year institutions is unlikely to change any time soon.

Conclusions and Recommendations

This project has hopefully advanced the way in which it is possible to develop information technology laboratory facilities for educational institutions, without defaulting to casual guesswork and expectations that potential problems can be resolved after-the-fact. The project can be considered a success if it encourages technologist, and especially technology educators, to think more dynamically about their facilities, and the possibilities that technology and good business sense can bring to the classroom.

Hopefully this project can also spark additional dialog about the various topics that have been broached in this paper – curriculum design, certification, business modeling – and the potential they have to transform educational processes. Although many of these items are controversial, it is a healthy process to invite that controversy, and ensure that educators, particularly those in technology, do not become complacent in their ideologies or theories about what constitute the *best* solutions to a particular problem.

Summary

This project presented a series of progressive approaches to developing a training laboratory for educators. It begins with selecting the basis for an academic curriculum, defining how and what courses will fulfill those areas of learning, identifying the technological requirements necessary to carry out those training sessions, and finally identifying and segregating those that have common facilities requirements.

This project also demonstrated that through the creative combination of existing stable and affordable technologies, facilities can be designed that are functional, yet modular in their design. Facilities created with the broadest and simplest platforms are most likely to be the most cost-effective solutions over the span of their utilization, given that the future development of information technologies is exceptionally difficult to predict.

Finally, this project demonstrated that, once a facility is in place, opportunities will always exist to continuously improve and mature the facility's operations for the purpose of better serving students' educational needs. By using business approaches like the Operations Maturity Model, a facility manager can assess the current state of operations, as well as discover new goals and objectives that can be targeted for improving the system. With the plethora of new technologies that are constantly evolving to enhance automation, maintenance, and disaster recovery efforts, the ability to achieve progressively higher maturity levels is becoming increasingly more possible.

With careful planning and a detailed analysis of the goals one wishes to achieve, information technology facilities can become a reality – even when budgets, space, and manpower are at a premium. It is hoped that this project can serve alas a guideline for others who are struggling to make information technology education in their institution a reality.

References

ACM (September 2005). Computing curricula 2005: the overview report. http://www.acm.org/education/curric_vols/CC2005_Final_Report2.pdf

ACM (October 2005). Computing curricula 2005: information technology volume. http://www.acm.org/education/curric_vols/IT_October_2005.pdf

Belles, R. & Miller, J. (February 2002). Multi-purpose specialty labs: providing specialized computer labs both cost and space effectively. Journal of Computing Sciences in Colleges, 17, 3, 46-52. http://portal.acm.org/citation.cfm?id=772648

Carey, K. (May 2004). A matter of degrees: improving graduation rates in four-year colleges and universities. Education Trust, The. http://www.cherrycommission.org/docs/Resources/Completion/A%20Matter%20of%20Degrees.pdf

Chabrow, E. (August 16, 2004). By the book: declining computer-science enrollments should worry anyone interested in the future of the U.S. IT industry. http://www.informationweek.com/story/showArticle.jhtml?articleID=29100069

Cherry, D., Phillabaum, P. & Valero, P. (2000). Living on the bleeding edge: creating and managing highly specialized student labs. Proceedings of the 28th Annual ACM SIGUCCS Conference on User Services: Building the Future, 40-49. http://portal.acm.org/citation.cfm?id=354918

CompTIA (2006). CompTIA website: http://www.comtia.org

ExtraQuest Corporation (2002). Process-driven delivery: ExtraQuest's adaptive implementation of OMM and SDD.

http://www.extraquest.com/EQOMMSDD.pdf

(i)Structure (2002). The operations maturity model: a new paradigm for optimizing IT's value. http://www.i-structure.com/resources_omm_whitepaper.asp

LeBlanc, C. & Stiller, E. (2004). Teaching computer security at a small college. Proceedings of the 35th SIGCSE Technical Symposium on Computer Science Education, 407-411. http://portal.acm.org/citation.cfm?id=971439

Madison, D. (February 2002). A dynamic lab approach to reinforcing computing concepts. Journal of Computing Sciences in Colleges, 17, 3, 60-66. http://portal.acm.org/citation.cfm?id=772650

Mason, J. (June 2003). Certifications – beat 'em, join 'em (or lose 'em)? Journal of Computing Sciences in Colleges, 18, 6, 36-45. http://portal.acm.org/citation.cfm?id=770827

Nelson, M. & Rice, D. (2001). Integrating third party-certification with traditional computer education. Journal of Computing Sciences in Colleges, 17, 2, 280-287. http://portal.acm.org/citation.cfm?id=775385

Patterson, D. (April 2005). The state of funding for new initiatives in computer science and engineering. Communications of the ACM, 48, 4, 21-25. http://portal.acm.org/citation.cfm?id=1053307

U.S. Bureau of Labor Statistics – Office of Occupational Statistics and Employment Projections (2006). Occupational outlook handbook (OOH), 2006-07 edition. http://www.bls.gov/oco/home.htm

Weeden, E., Scarborough, G. & Bills, D. (2003). Lab management strategies for IT database curriculum. Proceedings of the 4th Conference on Information

Technology Curriculum, 62-66. http://portal.acm.org/citation.cfm?id=947136

Yoo, S. & Hovis, S. (2004) Remote access internetworking laboratory. Proceedings of the 35th SIGCSE Technical Symposium on Computer Science Education, 311-314. http://portal.acm.org/citation.cfm?id=971409

Zweben, S. (May 2005). 2003-2004 Taulbee survey: record Ph.D. production on the horizon; undergraduate enrollments continue in decline. Computing Research News, vol. 17, no. 3. http://www.cra.org/CRN/articles/may05/taulbee.html

Appendix A

Computing Curricula 2005, DRAFT – I.T. Excerpts

The following are excerpts from the May 11, 2005 draft release of the 2005 Computing Curricula - Information Technology Volume document:

"***1.3 Definition of Information Technology as an Academic Discipline***

Information Technology (IT) in its broadest sense encompasses all aspects of computing technology. IT, as an academic discipline, focuses on meeting the needs of users within an organizational and societal context through the selection, creation, application, integration and administration of computing technologies.

1.4 Broad Goals of an IT Program

IT programs aim to provide IT graduates with the skills and knowledge to take on appropriate professional positions in Information Technology upon graduation and grow into leadership positions or pursue research or graduate studies in the field. Specifically, within five years of graduation a student must be able to:

1. Explain and apply appropriate information technologies and employ appropriate methodologies to help an individual or organization achieve its goals and objectives;
2. Manage the information technology resources of an individual or organization;
3. Anticipate the changing direction of information technology and evaluate and communicate the likely utility of new technologies to an individual or organization;
4. Understand and for some to contribute to the scientific, mathematical and theoretical foundations on which information technologies are built;
5. Live and work as a contributing, well-rounded member of society.

1.5 Program Outcomes

To enable IT graduates to achieve the above goals, they must possess the following skills upon graduation, namely the ability to:

(a) Use and apply current technical concepts and practices in the core information technologies;

(b) Analyze, identify and define the requirements that must be satisfied to address problems or opportunities faced by organizations or individuals;

(c) Design effective and usable IT-based solutions and integrate them into the user environment

(d) Assist in the creation of an effective project plan;

(e) Identify and evaluate current and emerging technologies and assess their applicability to address the users' needs;

(f) Analyze the impact of technology on individuals, organizations and society, including ethical, legal and policy issues;

(g) Demonstrate an understanding of best practices and standards and their application;

(h) Demonstrate independent critical thinking and problem solving skills;

(i) Collaborate in teams to accomplish a common goal by integrating personal initiative and group cooperation;

(j) Communicate effectively and efficiently with clients, users and peers both verbally and in writing, using appropriate terminology;

(k) Recognize the need for continued learning throughout their career.

1.6 Purpose and Structure of This Document

The primary purpose of this document is to set out a model curriculum that enables students to acquire the skills outlined above. It is intended as a guide for educational institutions of higher education in the creation and/or revision of baccalaureate programs in IT. It is expected that it will also be useful in the creation and/or revision of associate programs in IT." (ACM, 2005).

ITF. Information Technology Fundamentals (33 core hours)	Hours
ITF1. Pervasive Themes in IT (17)	17
ITF2. Organizational Issues (6)	6
ITF3. History of IT (3)	3
ITF4. IT and Its Related and Informing Disciplines (3)	3
ITF5. Application Domains (2)	2
ITF6. Applications of Math and Statistics to IT (2)	2
HCI. Human Computer Interaction (20 core hours)	
HCI1. Human Factors (6)	6
HCI2. HCI Aspects of Application Domains (3)	3
HCI3. Human-Centered Evaluation (3)	3
HCI4. Developing Effective Interfaces (3)	3
HCI5. Accessibility (2)	2
HCI6. Emerging Technologies (2)	2
HCI7. Human-Centered Software (1)	1
IAS. Information Assurance and Security (23 core hours)	
IAS1. Fundamental Aspects (3)	3
IAS2. Security Mechanisms (Countermeasures) (5)	5
IAS3. Operational Issues (3)	3
IAS4. Policy (3)	3
IAS5. Attacks (2)	2
IAS6. Security Domains (2)	2
IAS7. Forensics (1)	1
IAS8. Information States (1)	1
IAS9. Security Services (1)	1
IAS10. Threat Analysis Model (1)	1
IAS11. Vulnerabilities (1)	1
IM. Information Management (34 core hours)	
IM1. IM Concepts and Fundamentals (8)	8
IM2. Database Query Languages (9)	9
IM3. Data Organization Architecture (7)	7
IM4. Data Modeling (6)	6
IM5. Managing the Database Environment (3)	3
IM6. Special-Purpose Databases (1)	1
IPT. Integrative Programming & Technologies (23 core hours)	
IPT1. Intersystems Communications (5)	5
IPT2. Data Mapping and Exchange (4)	4
IPT3. Integrative Coding (4)	4
IPT4. Scripting Techniques (4)	4
IPT5. Software Security Practices (4)	4
IPT6. Miscellaneous Issues (1)	1
IPT7. Overview of programming languages (1)	1
NET. Networking (20 core hours)	
NET1. Foundations of Networking (3).	3
NET2. Routing and Switching (8)	8
NET3. Physical Layer (6)	6
NET4. Security (2)	2
NET5. Application Areas (1)	1
NET6. Network Management	
PF. Programming Fundamentals (38 core hours)	
PF1. Fundamental Data Structures (10)	10
PF2. Fundamental Programming Constructs (9)	9
PF3. Object-Oriented Programming (9)	9
PF4. Algorithms and Problem-Solving (6)	6
PF5. Event-Driven Programming (3)	3
PF6. Recursion (1)	1

PT. Platform Technologies (14 core hours)	
PT1. Operating Systems (10)	10
PT2. Architecture and Organization (3)	3
PT3. Computing infrastructures (1)	1
PT4. Enterprise Deployment Software	
PT5. Firmware	
PT6. Hardware	
SA. System Administration and Maintenance (11 core hours)	
SA1. Operating Systems (4)	4
SA2. Applications (3)	3
SA3. Administrative Activities (2)	2
SA4. Administrative Domains (2)	2
SIA. System Integration and Architecture (21 core hours)	
SIA1. Requirements (6)	6
SIA2. Acquisition/Sourcing (4)	4
SIA3. Integration (3)	3
SIA4. Project Management (3)	3
SIA5. Testing and QA (3)	3
SIA6. Organizational Context (1)	1
SIA7. Architecture (1)	1
SP. Social and Professional Issues (23 core hours)	
SP1. Professional Communications (5)	5
SP2. History of Computing (3)	3
SP3. Social Context of Computing (3)	3
SP4. Teamwork Concepts and Issues (3)	3
SP5. Intellectual Properties (2)	2
SP6. Legal Issues in Computing (2)	2
SP7. Organizational Context (2)	2
SP8. Professional and Ethical Issues and Responsibilities (2)	2
SP9. Privacy and Civil Liberties (1)	1
WS. Web Systems and Technologies (21 core hours)	
WS1. Web Technologies (10)	10
WS2. Information Architecture (4)	4
WS3. Digital Media (3)	3
WS4. Web Development (3)	3
WS5. Vulnerabilities (1)	1
WS6. Social Software	
Total Hours	**281**

Notes:

1. Order of Knowledge Areas: Fundamentals first, then ordered alphabetically.
2. Order of Units under each Knowledge Area: Fundamentals first (if present), then ordered by number of core hours.

Appendix B
Certifications and Courses Selected for Inclusion

Certification Preparation Courses

- Cisco CCNA – Semesters 1 through 4
- CIW Fundamentals
- CIW Site Building
- CIW E-Commerce
- CIW Internet Security and CompTIA Security+ (comparable coverage)
- CompTIA A+
- CompTIA Linux+
- CompTIA Network+
- CompTIA Security+
- CompTIA Server+
- Oracle Certified Professional (OCP), Database Administration
- Microsoft MCSA and MCSE, for Server 2003
- Microsoft Office Specialist
- Planet 3 Certified Wireless Network Associate (CWNA)

General Courses

- Introduction to Information Technology
- Programming (C++, Java, Visual Basic)

Appendix C
Certification Knowledge Domains

Cisco CCNA (Academy Semesters 1 through 4)

The Open Systems Interconnect Model (OSI)
Hardware and the OSI Reference Model
Wide Area Network Protocols
TCP/IP Technologies
Understanding Layer 2 Switching Technology
Understanding Layer 3 Routing Technologies
Configuring a Cisco Switch
Configuring a Cisco Router
Creating and Applying Additional Router Configurations

CIW Foundations

The Internet and the World Wide Web
Internet Addressing and Servers
Scripting, Connectivity, and Security
E-Mail, FTP, Newsgroups, and Telnet
Objects, Plug-Ins, Viewers, and Security
Search Engines and E-Commerce
Web Page Authoring and HTML Coding
Graphical Elements, Hyperlinks, and Tables
Forms, Images, and Frames
HTML Editors and Extensions

CIW Site Designer

Web Design Concepts and Site Development
Basic Web Technologies
Using Paint Shop Pro, Flash, JavaScript & DHTML
Using FrontPage, Dreamweaver, and Homesite
Advanced Web Technologies and Website Publishing

CIW E-Commerce

E-Commerce Overview
Legal Issues and the Internet
Marketing to the Web
Online Promotion Techniques
Building Usable Web Sites
E-Service for E-Customers
Supporting Business-to-Business Activities Online
Entry-Level E-Commerce Outsourcing
Mid-Level Online Storefront Packages
High-Level Online Storefront Packages
Working with Internet Information Server
Working with E-Commerce Software

Customizing an E-Commerce Site
Creating Online Catalogs
Processing Online Transactions
Supporting E-Services
E-Commerce Transaction Security
Managing E-Business Information

CIW Security Professional

Security Fundamentals
Security Auditing and Log Analysis
Security Attack Types and Encryption
Security Protocol Layers and Security
Security Firewalls
Security Operating System Security
Security Assessing and Reducing Risk
Security Auditing
Security Auditing and the Control Phase
Security Attack Detection and Response

CompTIA A+ Core

Installation, Configuration and Upgrading
Diagnosing and Troubleshooting
Motherboard/Processors/Memory
Preventive Maintenance
Printers
Basic Networking

CompTIA A+ Operating Systems

OS Fundamentals
Installation, Configuration and Upgrading
Diagnosing and Troubleshooting Networks

CompTIA Linux+

Planning the Implementation
Installation
Configuration
Administration
System Maintenance
Troubleshooting
Identify, Install, and Maintain System Hardware

CompTIA Network+

Media & Topologies
Protocols & Standards
Network Implementation
Network Support

CompTIA Project+

IT Project Initiation and Scope Definition
IT Project Planning
IT Project Execution, Control and Coordination
IT Project Closure, Acceptance and Support

CompTIA Server+

Installation
Configuration
Upgrading
Proactive Maintenance
Environment
Troubleshooting and Problem Determination
Disaster Recovery

CompTIA Security+

General Security Concepts
Communication Security
Infrastructure Security
Basics of Cryptography
Operational/Organizational Security

Microsoft MCSA/MCSE for Server 2003

70-270

Installing Windows XP Professional
Implementing and Conducting Administration of Resources
Implementing, Managing, Monitoring, and Troubleshooting Hardware Devices and Drivers
Monitoring and Optimizing System Performance and Reliability
Configuring and Troubleshooting the Desktop Environment
Implementing, Managing, and Troubleshooting Network Protocols and Services
Configuring, Managing, and Troubleshooting Security

70-290

Managing and Maintaining Physical and Logical Devices
Managing Users, Computers, and Groups
Managing and Maintaining Access to Resources
Managing and Maintaining a Server Environment
Managing and Implementing Disaster Recovery

70-291

Implementing, Managing, and Maintaining IP Addressing
Implementing, Managing, and Maintaining Name Resolution
Implementing, Managing, and Maintaining Network Security
Implementing, Managing, and Maintaining Routing and Remote Access
Maintaining a Network Infrastructure

70-293

Plan and Implement Server Roles and Server Security
Plan, Implement, and Maintain a Network Infrastructure
Plan, Implement, and Maintain Routing and Remote Access
Plan, Implement, and Maintain Server Availability
Plan and Maintain Network Security
Plan, Implement, and Maintain Security Infrastructure

70-294

Planning and Implementing an Active Directory Infrastructure
Managing and Maintaining an Active Directory Infrastructure
Planning and Implementing User, Computer, and Group Strategies
Planning and Implementing Group Policy
Managing and Maintaining Group Policy

70-296

Planning and Implementing Server Roles and Server Security
Planning, Implementing, and Maintaining a Network Infrastructure
Planning, Implementing, and Maintaining Server Availability
Planning and Maintaining Network Security
Planning, Implementing, and Maintaining Security Infrastructure
Planning and Implementing an Active Directory Infrastructure
Managing and Maintaining an Active Directory Infrastructure
Planning and Implementing User, Computer, and Group Strategies
Planning and Implementing Group Policy
Managing and Maintaining Group Policy

70-297

Creating the Conceptual Design through the Gathering and Analyzing of Business and Technical Requirements
Creating the Logical Design for an Active Directory Infrastructure
Creating the Logical Design for a Network Services Infrastructure
Creating the Physical Design for an Active Directory and Network Infrastructure

70-298

Creating the Conceptual Design for Network Infrastructure Security by Gathering and Analyzing Business and Technical Requirements
Creating the Logical Design for Network Infrastructure Security
Creating the Physical Design for Network Infrastructure Security
Designing an Access Control Strategy for Data
Creating the Physical Design for Client Infrastructure Security

70-299

Implementing, Managing, and Troubleshooting Security Policies
Implementing, Managing, and Troubleshooting Patch Management Infrastructure
Implementing, Managing, and Troubleshooting Security for Network Communications
Configuring and Troubleshooting Authentication, Authorization and PKI

Planet3 CWNA

Radio Frequency (RF) Technologies
Wireless LAN Technologies

Wireless LAN Implementation and Management
Wireless LAN Security
Wireless LAN Industry and Standards

Programming (C++, Java, Visual Basic)

Oracle Certified Professional, Database Administration

SQL

Writing Basic SQL Select Statements
Restricting and Sorting Data
Single-Row Functions
Displaying Data from Multiple Tables
Aggregating Data using Group Functions
Subqueries
Producing Readable Output with iSQL*Plus
Manipulating Data
Creating and Managing Tables
Including Constraints
Creating Views
Creating Other Database Objects
Controlling User Access

Database Administration Fundamentals I

Understanding Oracle architecture and its main components
Setting up password file authentications
Creating and Managing Initialization Parameter Files
Create a database using Oracle Database Configuration Assistant
Creating a database manually
Configuring OMF
Identifying DBA administrative tools

Database Administration Fundamentals II

Using Oracle Net Services and Client-Server connections
Establishing web client connections through Oracle networking products
Troubleshooting database connections
Using and Configuring the Oracle Shared Server
Backing up, restoring, and recovering databases
Configuring the database archiving mode
Using Recovery Manager (RMAN)
Loading data into, and transporting data between databases

Database Performance Tuning

Overview of Oracle9i Performance Tuning
Diagnostic and Tuning Tools
Sizing the Shared Pool
Sizing the Buffer Cache
Sizing other SGA Structures
Database Configuration and I/O Issues
Optimize Sort Operations

Diagnosing Contention For Latches
Monitoring and Detecting Lock Contention
Tuning Oracle Shared Server
Application Tuning
Using Oracle Blocks Efficiently
SQL Statement Tuning
Tuning the Operating System and Using Resource Manager

Appendix D
Matrix 1 – Certification and Curricula Mapping

ACM Computing Curricula, 2005 Mapping	Introduction to IT	Database Concepts & Principles	Cisco CCNA	Certified Internet Webmaster (CIW)	CIW Security & CompTIA Security+	CompTIA A+	CompTIA i-Net+	CompTIA Linux+	CompTIA Network+	CompTIA Server+	Microsoft MCSA/MCSE	Microsoft Office Specialist	Oracle DBA, Certified Professional	Planet3 CWNA	Programming (C++, Java)
ITF. Information Technology Fundamentals															
ITF1. Pervasive Themes in IT (17)	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
ITF2. Organizational Issues (6)	x														
ITF3. History of IT (3)	x														
ITF4. IT and Related/Informing Disciplines (3)	x											x			
ITF5. Application Domains (2)	x											x			
ITF6. Applications of Math and Statistics to IT (2)	x	x										x	x		
IAS. Information Assurance and Security															
IAS1. Fundamental Aspects (3)			x	x	x			x	x	x	x				
IAS2. Security Mechanisms (5)					x						x			x	
IAS3. Operational Issues (3)					x						x				
IAS4. Policy (3)					x						x				
IAS5. Attacks (2)					x									x	
IAS6. Security Domains (2)					x										
IAS7. Forensics (1)					x										
IAS8. Information States (1)					x										
IAS9. Security Services (1)					x				x		x			x	
IAS10. Threat Analysis Model (1)					x										
IAS11. Vulnerabilities (1)					x						x			x	
IM. Information Management															
IM1. IM Concepts and Fundamentals (8)		x										x			
IM2. Database Query Languages (9)		x										x	x		
IM3. Data Organization Architecture (7)		x											x		
IM4. Data Modeling (6)		x													
IM5. Managing the Database Environment (3)													x		
IM6. Special-Purpose Databases (1)													x		
IPT. Integrative Programming/Technologies															
IPT1. Intersystems Communications (5)	x				x			x	x	x	x				x
IPT2. Data Mapping and Exchange (4)															x
IPT3. Integrative Coding (4)															x
IPT4. Scripting Techniques (4)											x				x
IPT5. Software Security Practices (4)					x						x				x
IPT6. Miscellaneous Issues (1)															x
IPT7. Overview of programming languages (1)	x														x
NET. Networking															
NET1. Foundations of Networking (3).	x		x		x	x	x		x					x	
NET2. Routing and Switching (8)			x						x		x				
NET3. Physical Layer (6)			x						x					x	
NET4. Security (2)			x						x		x			x	
NET5. Application Areas (1)			x				x		x		x			x	
NET6. Network Management			x								x			x	

PF. Programming Fundamentals															
PF1. Fundamental Data Structures (10)															x
PF2. Fundamental Programming Constructs (9)															x
PF3. Object-Oriented Programming (9)															x
PF4. Algorithms and Problem-Solving (6)															x
PF5. Event-Driven Programming (3)															x
PF6. Recursion (1)															x
PT. Platform Technologies															
PT1. Operating Systems (10)					x	x		x		x	x				
PT2. Architecture and Organization (3)						x				x					
PT3. Computing infrastructures (1)						x		x		x	x				
PT4. Enterprise Deployment Software								x		x	x				
PT5. Firmware						x				x					
PT6. Hardware						x				x				x	
SA. System Administration/Maintenance															
SA1. Operating Systems (4)					x	x	x	x			x				
SA2. Applications (3)						x		x			x	x			
SA3. Administrative Activities (2)								x			x				
SA4. Administrative Domains (2)								x			x				
SIA. System Integration and Architecture															
SIA1. Requirements (6)	x						x			x				x	
SIA2. Acquisition/Sourcing (4)	x									x					
SIA3. Integration (3)			x				x	x		x	x			x	
SIA4. Project Management (3)	x				x					x					
SIA5. Testing and QA (3)			x					x			x			x	
SIA6. Organizational Context (1)								x			x				
SIA7. Architecture (1)			x					x			x				
SP. Social and Professional Issues															
~~SP1. Professional Communications (5)~~															
SP2. History of Computing (3)	x														
SP3. Social Context of Computing (3)	x														
SP4. Teamwork Concepts and Issues (3)	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
SP5. Intellectual Properties (2)					x										
SP6. Legal Issues in Computing (2)					x										
SP7. Organizational Context (2)					x										
SP8. Professional/Ethical Issues (2)					x										
SP9. Privacy and Civil Liberties (1)					x										
WS. Web Systems and Technologies															
WS1. Web Technologies (10)			x	x	x		x	x	x	x	x		x		
WS2. Information Architecture (4)	x			x									x		
WS3. Digital Media (3)				x			x								
WS4. Web Development (3)				x			x								
WS5. Vulnerabilities (1)				x	x			x	x		x				
~~WS6. Social Software~~															
~~**HCI. Human Computer Interaction**~~															
~~HCI1. Human Factors (6)~~															
~~HCI2. HCI Aspects of Application Domains (3)~~															
~~HCI3. Human-Centered Evaluation (3)~~															
~~HCI4. Developing Effective Interfaces (3)~~															
~~HCI5. Accessibility (2)~~															
~~HCI6. Emerging Technologies (2)~~															
~~HCI7. Human-Centered Software (1)~~															

Appendix E
Matrix 2 – Certification and Equipment

Certification Requirements Matrix **x** = required **o** = optional		Cisco CCNA	CIW Foundations	CIW E-Commerce	CIW Site Design	CIW Security & CompTIA Security+	CompTIA A+ Core	CompTIA A+ Operating Systems	CompTIA i-Net+	CompTIA Linux+	CompTIA Network+	CompTIA Server+	Microsoft MCSA/MCSE	Microsoft Office Specialist	Oracle DBA, Certified Professional	Planet3 CWNA	Programming (C++, Java)	Key Systems for Consideration
Hardware	PC - General	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
	PC - Internals		x			x	x			x	x							*
	Server - General											x						
	Server - Internals											x						
	Peripherals	x	x			x	x				x	x	x			x		
Networking	Hub/Switch	x	x			x	x	x	x	x	x	x	x			x		
	Router	x				x					x		x			x		
	Wireless	x	o			x	o				x					x		
Infrastructure	Rackmounting	x				x					x	x				x		
	Cabling	x	x			x	x			x	x	x	x			x		
	Support Server(s)	x		x	x	x				x	x	x	x		x	x	x	
Operating System	General	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
	Network Services	x			x	x			x	x	x	x	x			x		*
	Directory Services	x				x					x	x	x			x		*
	Windows	x	x	x	x	x	x	x	x		x	x	x	x	x	x	x	
	Linux/UNIX	x				x		x	x	x	x	x				x	x	*
	Other					x					x	x						*
Software	Major Vendor		x	x	x								x	x	x		x	
	Minor Vendor	x				x												
	Free/Shareware	x	x	x	x	x	x	x	x	x	x	x				x	x	
Lab Accessibility	To Internet	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
	To Intranet	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
	From Internet	o													o			
	From Intranet	o													o			
	Outgoing Filtering	x				x				x	x	x	x		o	x	o	*
	Incoming Filtering	x				x							x			x		*

Sample Textbooks for Reference:

Andrews, J (2005). A+ Guide to Software: Managing, Maintaining, and Troubleshooting, Third Edition Enhanced. Boston, MA: Course Technology.

Brown, J., Thomas, S. & Bruzzese, J. (2002). CIW: Site and E-Commerce Design Study Guide. Berkeley, CA: Sybex

Bott, G. (2004). Implementing, Managing, and Maintaining a Microsoft® Windows Server™ 2003 Network Infrastructure (70-291). Redmond, WA: Microsoft Press.

CCNA 1 and 2 Companion Guide, Revised. (2004). Indianapolis, IN: Cisco Press.

CCNA 3 and 4 Companion Guide, Revised. (2004). Indianapolis, IN: Cisco Press.

Ciampa, M. (2004). Security+ Guide to Networking Security Fundamentals, Second Edition. Boston, MA: Course Technology.

Conrad, J. (2001). Server+ Guide to Advanced Hardware Support. Boston, MA: Course Technology.

Corbin, W. (2004). Planning, Implementing, and Maintaining a Microsoft® Windows Server™ 2003 Active Directory® Infrastructure (70-294). Redmond, WA: Microsoft Press.

Dean, T. (2005). Network+ Guide to Networks, Fourth Edition. Boston, MA: Course Technology.

Eckert, J. & Schitka, M. (2005). Linux+ Guide to Linux Certification, Second Edition. Boston, MA: Course Technology.

Lane, P., Sodeman, W. * Dulaney, E. (2002). CIW: Foundations Study Guide. Berkeley, CA: Sybex

Mackey, D. (2003). Web Security for Network and System Administrators. Boston, MA: Course Technology.

McCann, B. & DiNicolo, D. (2005). 70-290: MCSE Guide to Managing a Microsoft Windows Server 2003 Environment, Enhanced. Boston, MA: Course Technology.

McCann, B. & Wright, B. (2005). 70-293: MCSE Guide to Planning a Microsoft Windows Server 2003 Network, Enhanced. Boston, MA: Course Technology.

Tittel, E. & Stewart, J. (2005). 70-270: MCSE/MCSA Guide to Microsoft Windows XP Professional, Second Edition. Boston, MA: Course Technology.

Zacker, C. (2004). Managing and Maintaining a Microsoft® Windows Server™ 2003 Environment (70-290). Redmond, WA: Microsoft Press.

Zacker, C. (2004). Planning and Maintaining a Microsoft® Windows Server™ 2003 Network Infrastructure (70-293). Redmond, WA: Microsoft Press.

Appendix F
Core Facility Requirements and Courses

Filtered Key Requirements Matrix **x** = required **o** = optional		Cisco CCNA	CIW Foundations	CIW E-Commerce	CIW Site Design	CIW Security & CompTIA Security+	CompTIA A+ Core	CompTIA A+ Operating Systems	CompTIA i-Net+	CompTIA Linux+	CompTIA Network+	CompTIA Server+	Microsoft MCSA/MCSE	Microsoft Office Specialist	Oracle DBA, Certified Professional	Planet3 CWNA	Programming (C++, Java)	Key Systems for Consideration
Infrastructure	Rackmounting	x				x					x	x				x		*
Hardware	PC - Internals		x			x	x			x	x							*
O.S.	Network Services	x			x	x			x	x	x	x	x			x		*
O.S.	Directory Services	x				x					x	x	x			x		*
O.S.	Linux/UNIX	x				x		x	x	x	x	x				x	x	*
O.S.	Other					x					x	x						*
Access	Outgoing Filtering	x				x				x	x	x	x		o	x	o	*
Access	Incoming Filtering	x				x							x			x		*
	Key Courses	X				X				X	X	X	X			X		