A SURVEY OF EDUCATIONAL SYSTEMS DEVELOPMENT METHODOLOGIES

Honors Thesis

Carly Walsh

April 4, 2004

Introduction

What is the SDLC

Information systems are evident in many aspects of life. From swiping your card at the ATM to turning on your television set to connect to digital cable, people everywhere interact with systems without thinking anything of it. If all of the information systems were somehow taken away, most people would become utterly bored after a few days. Consider moving into a new house before anything is hooked up – no television, no Internet, no phone. Think of how many times you would walk around the house and look for new things to do, even more so when it is raining and you cannot go outside.

What is a system? Hyperdictionary.com defines a system as "instrumentality that combines interrelated interacting artifacts designed to work as a coherent entity." Basically, a system is two or more parts working together as a whole. Variations of the definition exist for different disciplines such as biology, chemistry, and computing. Each element of a system is involved in design and analysis. The definition of analysis is similar – an investigation of the component parts of a whole – whereas design is the act of working out the form of something. An example of design would simply be making a sketch, outline, or plan.

The Systems Development Life Cycle (SDLC) is a management process that utilizes both design and analysis to plan, build and control systems and applications (Hoffer, 2002). The SDLC is most often referred to as the Waterfall model or method. It is an iterative process used to control costs and time, work with project timelines and deadlines, and stay on budget. Most

large development business projects use this traditional method when the system is well understood. However, this tool can be helpful when used in any type of project if used correctly.

The need for a reliable repeatable process (Why use SDLC)

(Pitagorsky, 1998)

Some professional projects are extremely large and time-consuming. Because they are so large, management needs to be able to control these projects. This is where the SDLC is applied; helping to keep the project monitored and aligned.

Nearly everyone will need to plan a project in his or her life. Intuitively, people apply the SDLC with high frequency. For example, a student may approach a writing assignment by either typing the project in one sitting, or by creating an outline or taking notes first. This would be a way of planning the direction using one or two of the steps in the SDLC. Even if the student plans the assignment in their mind, they are still using the same kind of process.

Some would say that one approach is better, and they may be right. However, the point is merely that any approach will work, but it could be helpful to have a detailed, repeatable process to use in every project attempted. This allows for structure and for those who have no idea where to start, the SDLC gives them options.

Key issues in systems development and support

Managers rely on employee knowledge and ability to complete tasks. Project importance is usually rated so those working on it know the timeframe and costs associated with the project. Information systems are built to maximize efficiency, communication, and knowledge. However, the system can only be as good as the people who build it. These designers must also keep in

mind any ethical and legal issues, while satisfying the needs outlined in the original plan (Whitten, 2004).

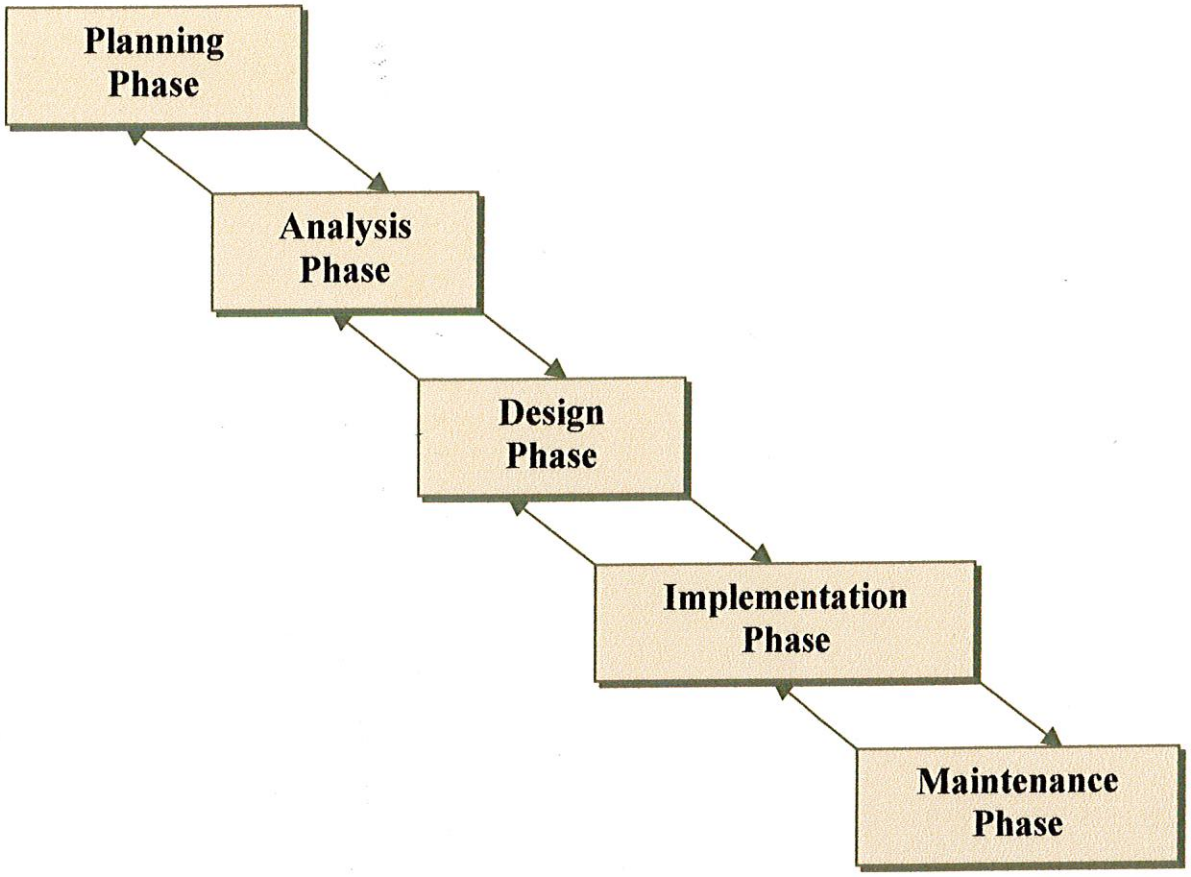
Continuous improvement/adjustment

A good working system has certain attributes associated with it. It should be flexible to allow for future changing needs, require low maintenance, and be scalable. It is also important that a system be able to work well within another system or alongside other similar systems. Integration can be made easier if planners remember to account for the future (Lewis, 2001). Users of a system rely on a system to do what it is supposed to do. Problems result in delays, delays mean loss in productivity, and any loss in productivity may lead to an unsatisfactory results. Unfortunately, when problems do arise, they may also be blamed on planners, further pushing them to make sure that their system is perfect before it is widely used. Most planners would much rather use their time improving a system than fixing problems they should have caught earlier.

The Systems Development Life Cycle

Traditional life cycle – the Waterfall Model

The Waterfall model (or traditional SDLC) is perhaps the most widely used development methodology (Hoffer, 2002). Each phase is clearly defined, but carries over into the next. The project may not have a set ending date, as the phases may be revisited and maintenance normally continues throughout the useful life of the system.



(Hoffer, 18, 2002)

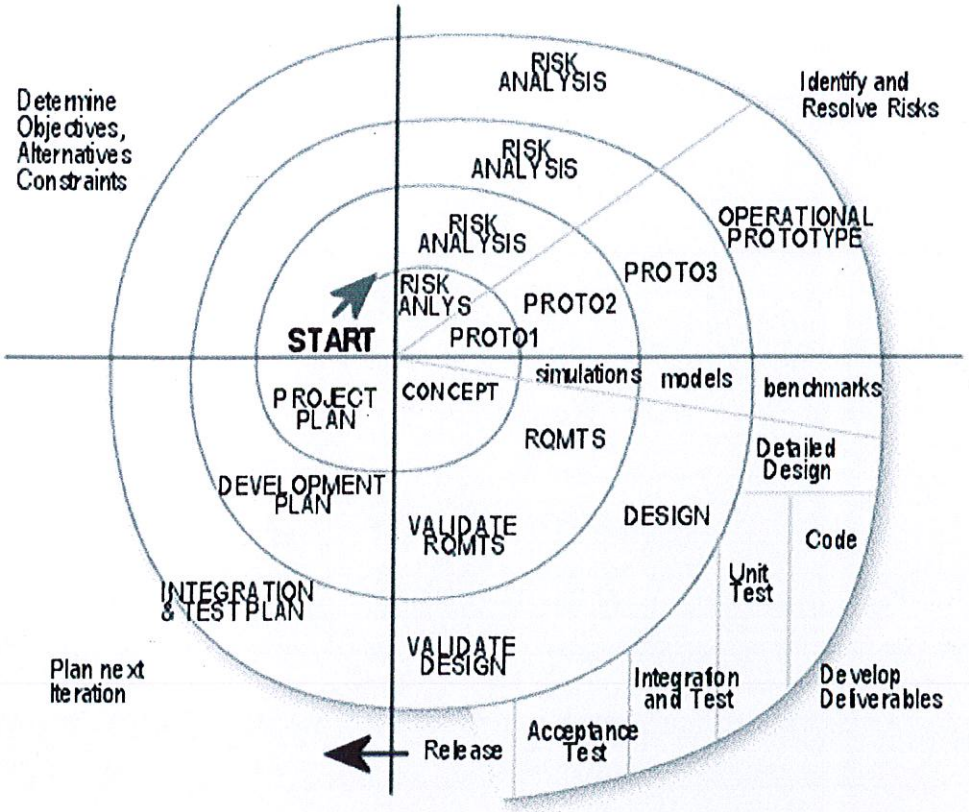
The need for variations on the basic theme (SDLC Alternatives)

Choosing a methodology is important for any system. Each one has its advantages and disadvantages, but it is important to choose the methodology that would work best for each individual system because each is time consuming and comes with its own costs. Aside from the SDLC, other methodologies like Rapid Application Development, the Spiral method, and the Incremental method can also be used to develop a new system.

Spiral Method

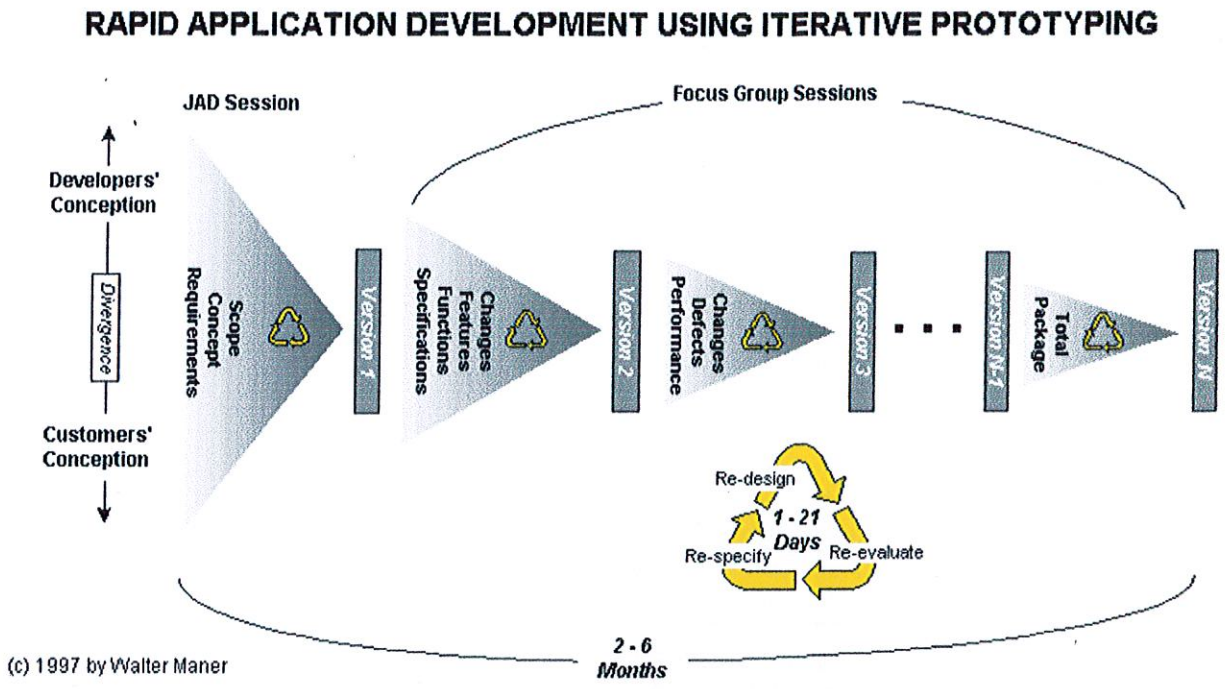
When a project is driven mainly by risk, the Spiral method may be used (Boehm, 2000; Sews, 2000; Knight, 2001). Risky projects are usually very intensive, large, and time-consuming. They contain individual project risks, or difficulties with scheduling, personnel, and budgeting, technical risks, including the scope of the project and actual complexity in design and implementation, and business risks, including the market, management, and sales risk.

There are two main features of the Spiral method – the cyclical nature, which allows for incremental change in the project definition and implementation, and milestones, which help to ensure commitment to feasibility and completion. This methodology is useful when managers "expect the unexpected" because it can adapt and accept these surprises. The prototypes made with each turn of the spiral allow for additional testing and evaluation.



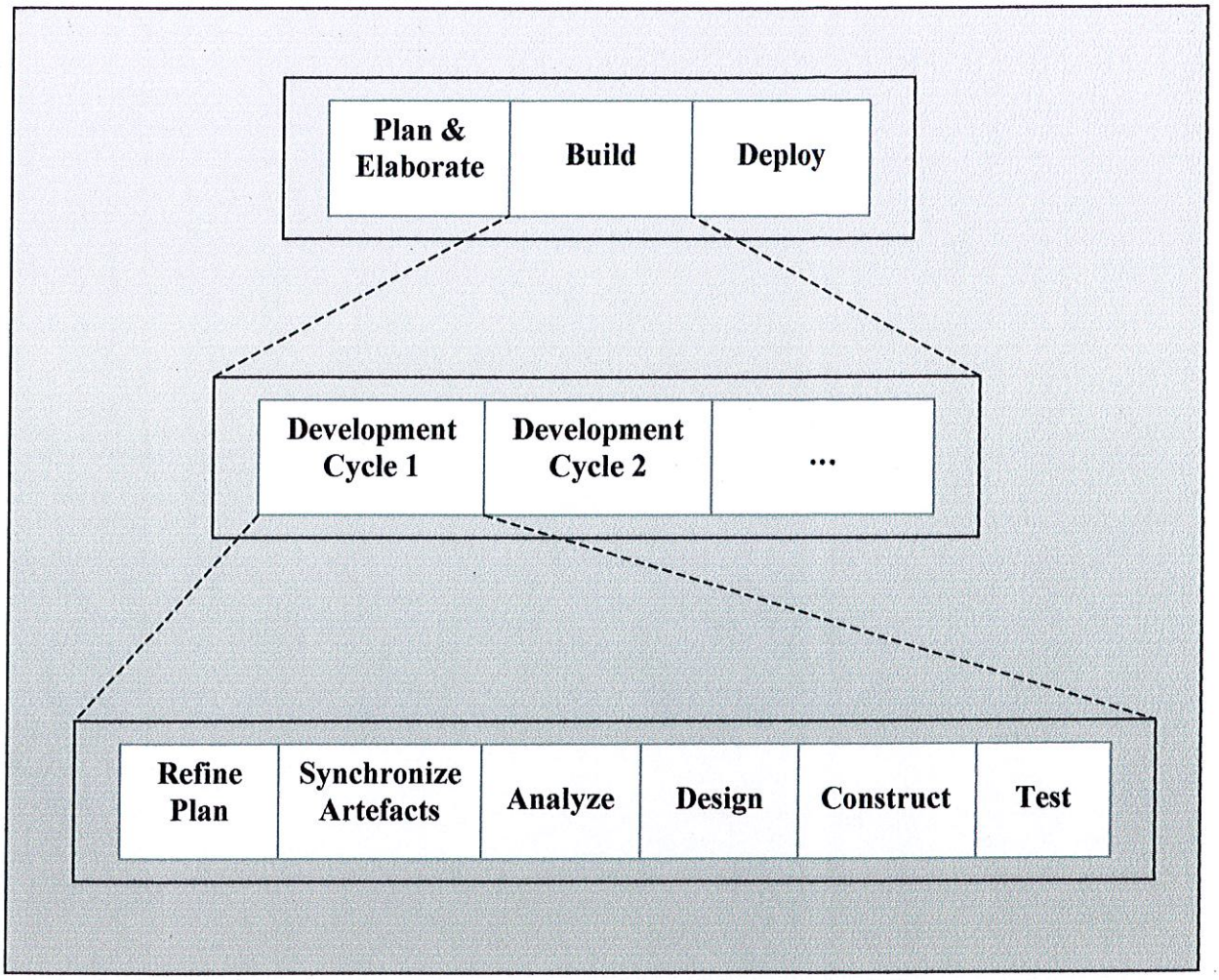
RAD

The Rapid Application Development method is a high-speed version of the Waterfall method (Eliott, 1997; Knight, 2001). By using a component-based construction, the development cycle is shortened considerably. Managers use this approach when the system is widely understood, low risk, and needs to be completed in a short time frame.



Incremental

The Incremental model, similar to the waterfall model, is done using increments, or sections of the project (Hunger, 2000). Each section is treated as its own project, with the waterfall method applied to fulfill individual requirements and objectives. Thus, multiple cycles run through the SDLC separately and then are brought back into the whole.



(Hunger, 2000)

Project Life Cycle issues/criticisms

The SDLC (along with many other methodologies) has many criticisms associated with it (Okon, 2002). The traditional linear waterfall design may be too rigid for most systems because everything is done chronologically. Some believe the SDLC is inflexible, lengthy, and costly. If the requirements are not clearly and thoroughly defined, the system can be difficult to modify and add these newly found necessities. Also, testing must ensure that the components work together within the system. This should not be overlooked.

If planners do not keep in mind end-user involvement, they may be faced with the frustrations and confusion these users acquire. The client must be able to use and understand a system for optimal benefit. Also, for smaller scale systems, a shorter process may work better with less resource costs.

Project Success/Failure Factors

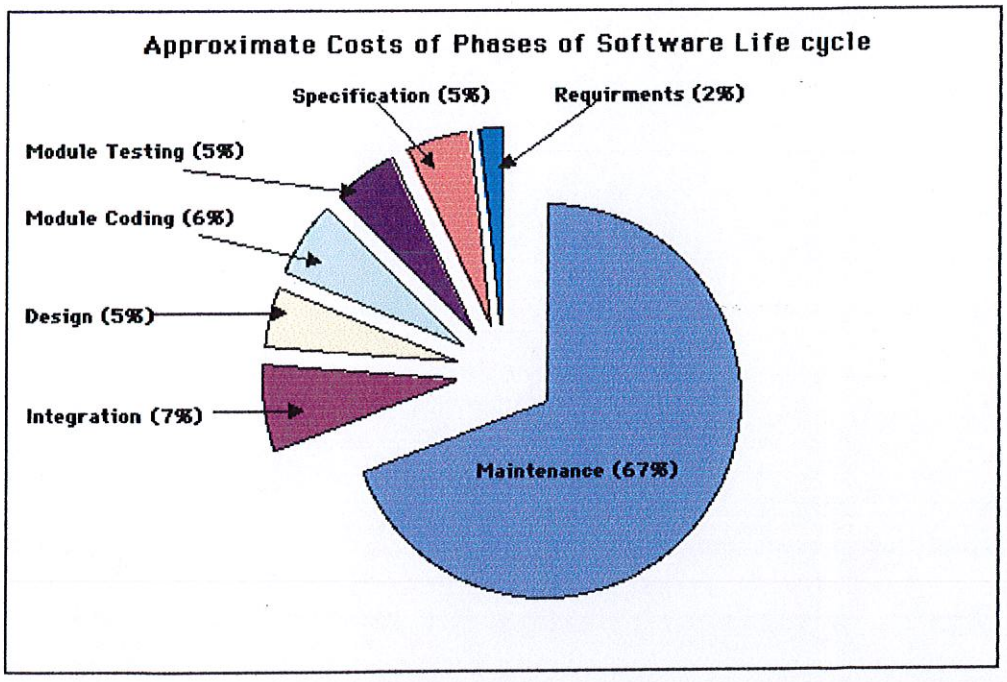
Successful projects have many similarities, beginning with management support (Ahituv, 1984; Knight, 2001). Sufficient support improves the confidence of the designers, implementation team, and users because they know that the project is needed and important to the organization. Support also helps when problems arise, because users know that management will be there to make sure that problems are resolved quickly and effectively. Management that gives realistic milestones and work schedules will be better respected and keep those involved informed. Teams will be more likely to finish a project on time and on budget when they are given this support.

Successful projects also need to have clearly defined system requirements and detailed plans (Ahituv, 1984). These requirements help to make sure that the system will do everything it is supposed to do and all aspects within the system are thoroughly designed. Additionally, successful projects have users who have been involved with the process. This helps them to use the system correctly and to use the components in the most efficient way.

A successful project should be fully functional and accepted by most users when completed. Additions and enhancements may be needed within the maintenance of the project, but the system should never be interrupted. Finally, the benefits will usually outweigh the costs in the long run for a proper and successful project.

Conversely, some projects fail. These projects usually have lacked support either technically or executively. From there, planners may have begun with unclear goals and objectives. This could lead to poor planning and not enough time allotted for the project completion. The lack of support could mean that too few people have been assigned and not enough additional resources have been given to the project. Furthermore, analysis may have been done quickly, omitting much-needed steps. The project or old system may not have been understood prior to finishing the new system as well. Finally, the project risks may have been overlooked, the project may have been inflexible and difficult to use, or the system may not have the ability to expand or be enhanced.

Sometimes projects can go over budget. Costs of professional projects need to be determined before the initial inception, and reevaluated when needed throughout the process. The following chart shows an example of the division of costs for a normal software project.



(Upchurch, 1997)

The Major SDLC Phases

Planning

Most of the preliminary work should be done during the Planning stage of the SDLC. The planner must first define the problem or project according to its assignment. It is also extremely important to define the scope of the project. If done correctly, it may be decided that the project will take too much time and effort to build versus the benefits it may bring, causing the planners to scrap the project. This is also called a feasibility study. They will ask if a project is feasible financially, operationally, technically, and timely (Lewis, 2001; Hoffer, 2002). If the benefits outweigh the costs, then the project will continue through the SDLC.

After deciding whether a project is feasible or not, uses of the system need to be identified. This allows for better detail in configuring the system and better use of individual components within the system. How to staff the project, if needed, or how the project should be divided, should determine the path to completion. Finally, the project should be initiated and launched so that through the rest of the SDLC additions and improvements are made.

Analysis

The Analysis stage of the SDLC is mainly comprised of the discovery and understanding of system needs. The problem domain must be evaluated, information gathered, and system requirements diagramed, prioritized, and documented. Alternative approaches to the fulfillment of a project must be generated and evaluated. This allows for a better high-level view of needs, and ensures that the alternative with the best fit for the project is selected. If needed, an analyst may review his/her recommendations with management to obtain crucial feedback for the project.

Once a direction for the project has been selected, or possibly a few comparable choices, the analysts may decide to build a prototype based on the needs and direction of the project. This prototype will be used to discover added requirements, and will be modified throughout the entire development process. If the prototype is good enough, it may be used in the finished product.

Another element of the Analysis phase is evaluation of the current system. This can help determine the needs of the new system and the needs that were not previously fulfilled by the older system. A decision may be made to enhance the old system, rebuild the old system, or come up with a completely different new system.

Design

The Design stage of the SDLC details the design for many different aspects of the system or project. Design and integration of the network, application architecture or processing functions, and database must be prepared for integration. User interfaces, including forms, screens, and reports, and system interfaces, which provide communication with other systems, are extremely important in order to make sure that the system will be used correctly.

Analysts must construct high-level, functional descriptions of the logical and physical (data capture, processing, and information output) design of the system. This may be illustrated in a model or detailed documentation. Additions to the prototype may be made to ensure correctness and workability. Finally, programming languages must be assessed and chosen for the system, keeping expandability, flexibility, and ease of use in mind.

Implementation

After the design is completed, the system must be implemented. All coding for programs is completed, software components are constructed, and data from the old system is converted to fit into the new system. The users of the system should be thoroughly trained and the system must be documented to include all aspects detailed in the Design and Analysis phases.

Conversion during the implementation phase can be done in different ways (Oz, 2004). A manager may decide that the old system must be used along side the new system for a period of time before switching completely to the new system. This process is called parallel conversion. Parallel conversion minimizes the amount of risk associated with the project because if the new system fails, the old system is still up and running. However, parallel conversion can be an expensive choice because of the costs of running both systems at the same time along with the added support required by each system.

The second type of conversion is called phased conversion. Managers may choose to use the phased conversion method if the system is large and would be easier to implement in smaller sections in separate phases. This approach reduces risk because one module can fail without the others also failing. It is also easier for the users of the system to be trained on smaller sections instead of one large system. Still, this approach can take a longer time to implement, meaning that users will not be enjoying the benefits of the new system sooner.

Another conversion method sometimes used is called the cut over conversion, or flash cut conversion. The old system is scrapped and the new system takes over all operations completely. This method can be very risky, but if the new system is implemented without error, this method can save money on resources that might not be needed.

The last approach is called the piloted conversion method. The new system is implemented in only one business area, allowing for the rest to operate normally without interruption using the old system. The new system is tested, fixed, and finally applied to the rest of the business. Risk is minimized to the test area, and helps determine how users will react to the new system. As with the parallel conversion method, the piloted method can delay the benefits of the new system for the areas outside of the test.

The system should be completely installed and running during the Implementation phase. Verification and testing should be started to help guarantee that the system works. Once most of the bugs (errors) are resolved, an analyst can determine that the system is productive and consider the project complete.

Evaluation/Support/Maintenance

Completed projects require maintenance. The Maintenance phase is usually the lengthiest and costliest phase of the systems development life cycle. It takes time for users to access all areas of the system and will usually run into snags along the way. Changes will need to be made and upgrades will help improve the usability and lifespan of the system. Sometimes a help desk may be used to help repair these issues and help improve what needs enhancement. All of this can cost a lot for the project and possibly make a project go over budget.

Project Types

There are many different types of projects, each with its own requirements. Business people, students, and almost every other person will complete some type of project in their lifetime. Some even see their everyday decisions as small projects – applying and adapting

methodologies to the outcome. Others carry out their projects more professionally, using more time and effort to satisfy objectives.

Types of business processes

Projects completed in businesses differ greatly from other projects. Risks are greater, schedules are tighter, and the benefits are numerous. Projects are decided upon by managers and assigned to the appropriate personnel. It is important that the most qualified employees are utilized for their knowledge and expertise. There are three types of business processes – business process automation (BPA), business process improvement (BPI), and business process reengineering (BPR) (Green, 2004). Evaluating goals of a project can also help to choose the appropriate strategy.

Business process automation (BPA)

Business process automation is a process used when a computer can perform simple manual tasks in a system. This allows for employees to use their time doing other more important tasks, making them more efficient. The ways things are done are not significantly changed, only shortened. The scope of BPA is narrow, keeping the objective clear and simple. Additionally, BPA can add a small amount of value to the business at little cost by improving or enhancing their existing processes.

Business process improvement (BPI)

BPA projects sometimes slip into business process improvement when a system is evaluated for automation and other small to moderate improvements are needed. Because of this slip, BPI is probably the most commonly used technique, allowing for improvement without a complete overhaul of a system, usually on time and on budget. Sometimes BPI is used when a

process is outdated. This works sometimes, but the system should usually be completely changed to bring it to the present.

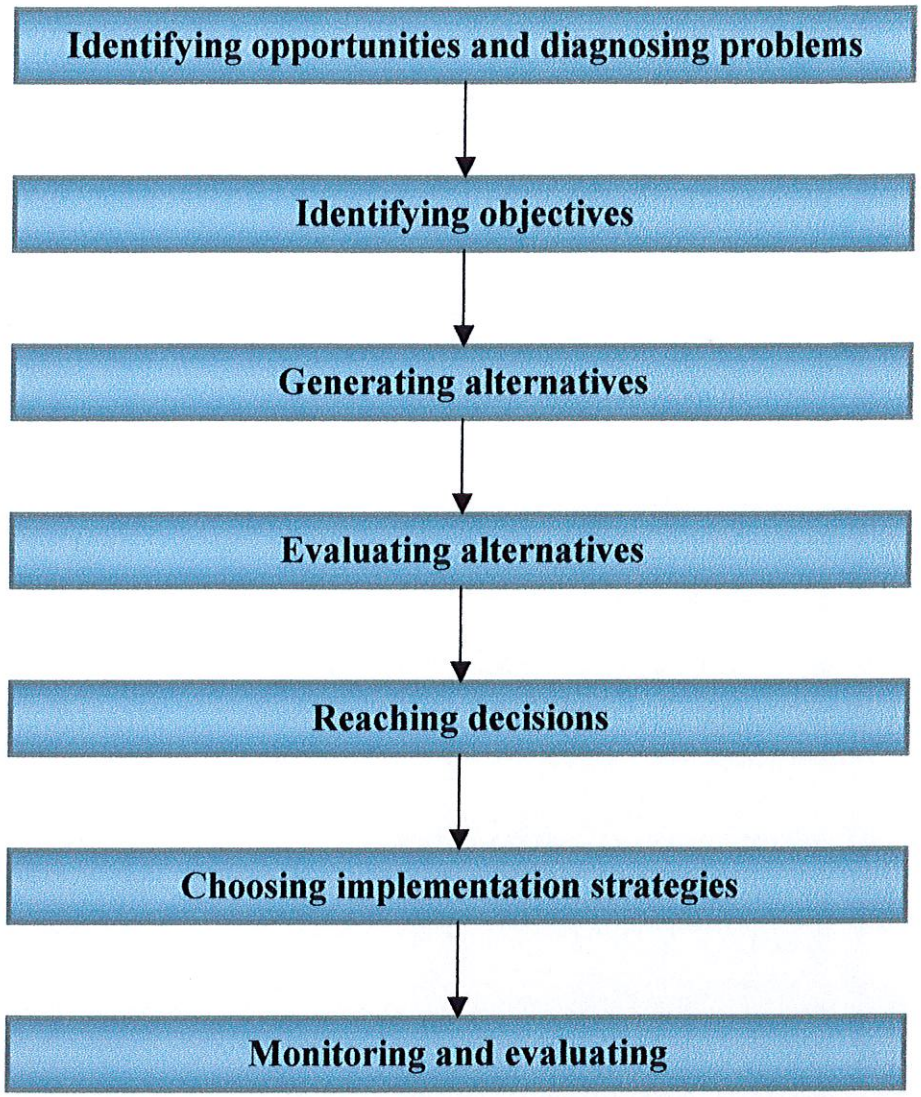
Business process reengineering (BPR)

Business process reengineering is used when a completely new system is needed to replace an outdated system. This requires an ultimate rethinking and drastic redesign of broad business processes. Major revisions are made, especially in the way that work is performed. Time is spent more on designing the new system than evaluating the old as-is system. Although usually quite costly and lengthy, BPR has the potential to significantly enhance and add value to the business with the changes it makes. Nevertheless, it is critical that steps not be overlooked in order for BPR to succeed. The massive amounts of detail needed for BPR can also cause failure, as businesses often get lost. BPR is also much more risky as so much effort is spent in completing the project. Businesses that can tolerate this risk and have the budget for it are much more likely to use BPR to increase their efficiency. In other businesses, however, a combination of BPR and BPI may be used.

Roles and Responsibilities at BSU

Modified SDLC (Decision-Making Process) for non-systems projects

The traditional SDLC is used mainly for software projects; however, many other projects exist that need a similar tool to plan and execute. The following graphic depicts a modified SDLC that can be used for non-systems projects:



(Lewis, 186, 2001)

SDLC in software project

There are many different types of information systems projects. Each is specifically used in certain instances, and each usually utilizes one or more development methodology. The following table outlines these systems:

Information System	Definition
Executive Support System or Executive Information System (ESS/EIS)	**Information system used to help executives evaluate long-term planning for the organization**
Decision Support System or Group Decision Support System (DSS/GDSS)	**Information system specifically built to help managers or groups of managers make decisions without having to study lengthy data**
Transaction Processing System (TPS)	**Information system used to record transactional (sale, purchase, etc.) processes**
Expert System or Knowledge Base System (ES/KBS)	**Information that includes human expertise to help managers make decisions**
Management Information System (MIS)	**Information systems used for planning, decision-making, control, recording transactions, and problem solving**
Geographical Information System (GIS)	**Information system used to make geographical decisions**

(Oz, 2004)

SDLC in software project vs. other project

The systems development life cycle is a process used mostly by computer analysts, programmers, and other professionals; however, the modified SDLC can also be used in a more general form for anyone attempting to plan, execute, or control a project. Professionals usually need an approach like the SDLC to ensure a project is appropriately planned, timely, and obtains the necessary result. Whereas others may just need the SDLC as a basic outline or starting point to complete a project.

Some undergraduates will begin and finish a project in one sitting, neglecting to plan or design much more than what they have mapped out in their minds. On the other hand, many of their fellow students may apply the SDLC or a similar approach to an assigned project. These students may not receive better grades than their peers, however, they may have a better understanding of what they wanted their project to display. If asked to explain exactly what their

project is, a more prepared student may impress their professors more than an unprepared student.

Who's who in educational systems development

In systems development, it is the responsibility of an educator to assign a project to his/her class, expect the class to complete the project, and assess the quality of the project. It is not necessarily an educator's duty to expect a student to abide by strict guidelines, such as the SDLC. However, if the SDLC is seen only as a basis, with changes made as needed, it may become a useful tool for teachers and their students. Teachers may have an easier way to evaluate the progress and overall quality of a project completed by a student. They may also have an easier way to explain to their students how a project may be completed when they enter the business world.

Students capture benefits from the SDLC when they struggle with the direction they wish to go in attempting an assigned project. Students must keep in mind that they will most likely not be required to use the SDLC or a similar tool, but they may receive a better evaluation if their teacher sees the effort and planning that went into their work. Their responsibility to their teacher is to complete the project; while they have the responsibility to themselves to achieve the best marks they are capable of receiving.

What is expected of BSU students

Expectations of students at Bemidji State University vary. Some professors may require students to follow a set of steps to complete a project, with evaluation during and after completion. Others may give full freedom to their students, with the expectation only that their

students will try their best. Still some professor will allow some freedom, with stricter guidelines on certain aspects of a project.

Bemidji State University faculty surveys

In April 2004, a few Bemidji State University professors were surveyed anonymously. Faculties were chosen to participate if they taught a class involving computer systems or other project driven course. After giving their consent, they filled out the survey to the best of their knowledge.

FACULTY #1
What are some ways computers are used to aid students in your course(s)?
Computers are essential in the learning, understanding, and implementation of business solutions in all of my classes.

What types of projects are students required to plan out and assess in your course(s)?
Decision Support Systems, Profit/Product Mix projects, Least Cost projects, Shortest path projects, Programming Projects etc...

How is the SDLC (or a similar process) used in your classroom?
All projects, even those non-system based, must be developed through a multi-step process from investigation of initial requirements through analysis, design, implementation and maintenance. 1) Determine what the project requirements and needs are; 2) Determine how the Data, dialogue, and processes will be implemented, 3) Physically complete the project; and 4) Maintain it's accuracy and integrity by checking for errors and updating information.

How would you change or modify the basic SDLC to gear toward your course (i.e. add, delete or change steps)?
The SDLC is a very good step-by-step life cycle process. I use it as is.

Do you believe that most students use some kind of system (similar to the SDLC) to complete a project or task? Why or why not?
Yes... I believe they just are not aware that they are using a system.

FACULTY #2
What are some ways computers are used to aid students in your course(s)?

Mainly for writing, although students in Web Design and Content writing design and write web sites.

Students in one of my classes design a 10-week writing project involving either a wiki or a blog, but they are not required to follow a structured procedure.

What types of projects are students required to plan out and assess in your course(s)?
Multiple web design projects in a few of my classes. In early projects, I handle the procedure, giving set tasks for students to address, explaining why and where they fit into the procedure. In later tasks, I use a general procedure, drawn in part from web design shops but I've adapted it for use in teaching and learning.

How is the SDLC (or a similar process) used in your classroom?
The stages of the procedure I use resemble the SDLC, but I keep the focus on teaching and learning by designing the product rather than the product itself.

Analysis – do some research and create a rough plan for the project

Design – create the site plan on paper

Implementation – This is done over the course of a few weeks. I refer students to previous work they've done, and a page of site specs.

Evaluation – students write a page in which they consider what they have done. In this stage, students are to reconsider Analysis to see where they've come from and reconsider what they did.

We also do some exercises in class at each stage that addresses that stage as a way of demonstrating how to go about handling the larger task.

How would you change or modify the basic SDLC to gear toward your course (i.e. add, delete or change steps)?
I modify the process every semester, tweaking it to focus student on issues and principles at each stage, and incorporating new techniques as they are developed. I've introduced, for instance, more info architecture techniques into class exercises, and will be formally implementing user testing in the process this semester as part of Evaluation and revision.

I'm always trying to get more recursiveness into the process, because that's where the big breakthroughs in design AND in learning occur. To create recursiveness, we stop at least twice in the process for students to review each other's work and make suggestions for changes. I'm not sure where this stopping for feedback fits in SDLC. I see it as part of Implementation.

Do you believe that most students use some kind of system (similar to the SDLC) to complete a project or task? Why or why not?
No. When I do not require a formal process, students will attempt to design and implement a site without any planning, playing it by ear from the first step. They will go at a project without any

consideration of purpose, audience, and without any consideration of options are alternatives. All, for instance, will create one set of wireframes and stop rather than create alternatives and select.

Additional comments:
SDLC looks like an interesting process. As I mentioned above, I try to keep the focus on the learning and so adapt procedures for learning rather than trying to recreate a procedure for creating a product.

FACULTY #3
What are some ways computers are used to aid students in your course(s)?
I use PowerPoint to help in teaching the concepts. Then I encourage the students to use Visible Analyst to apply SDLC concepts to a case study project.

What types of projects are students required to plan out and assess in your course(s)?
In my course, the students are assigned a multiple part case study on development of a new computer system.

How is the SDLC (or a similar process) used in your classroom?
I teach the process and have the students work through a project that applies those concepts.

How would you change or modify the basic SDLC to gear toward your course (i.e. add, delete or change steps)?
I don't know that I modify the basic SDLC. I do incorporate some of the alternatives like prototyping in the concepts I teach.

Do you believe that most students use some kind of system (similar to the SDLC) to complete a project or task? Why or why not?
I don't think most students use this type of concept in class projects for courses outside of the systems course. I don't think the projects for most classes are perceived, as being that complex and students are pressed for time, so they tend to take less structured approaches for most class projects. I am simply hopeful that they understand the concepts and will apply them on future tasks once they are employed.

FACULTY #4
What are some ways computers are used to aid students in your course(s)?
- *Power point presentations used for some lectures*
- *Computer use required for class projects*
- *Email correspondences between students and faculty*

What types of projects are students required to plan out and assess in your course(s)?
- *Presentations*

Resources

Ahituv, N., et al. *A Flexible Approach to Information Systems Development*. MIS Quarterly, Vol. 8, No. 2, June 1984, pp. 69-77.

Boehm, Barry. *Spiral Development: Experience, Principles, and Refinements*. Spiral Experience Workshop. February 9, 2000. <http://sunset.usc.edu/publications/TECHRPTS/2000/usccse2000-507/usccse2000-507.pdf>

Eliott, Geoffrey. Principal Lecturer. *Rapid Applications Development (RAD): an odyssey of information systems methods, tools and techniques*. U.K. July 1997. Journal of Financial Information Systems, Copyright © 1995.
http://www.shu.ac.uk/schools/fsl/fisjnl/vol1996/pprs1997/gelliot/fis97.htm

Green, Gina. *Analysis Strategies – The Analysis Process*. Baylor University. January 7, 2004. Copyright © 2004 Baylor University.
<http://business.baylor.edu/Gina_Green//teaching/isy3320/lectures/analysis_process.ppt>

Hoffer, Jeffery A., et al. *Modern Systems Analysis and Design*. 3rd ed. New Jersey: Prentice Hall, 2002.

Hunger, Michael. Iterative, Incremental Development. October 25, 2000. < http://www-emw.inf.tu-dresden.de/de/pdai/Forschung/refactoring/refactoring_html/node21.html>

HyperDictionary Search. Copyright © 2000-2003 Webnox Corp.
<http://www.hyperdictionary.com/dictionary/>

Knight, Steinbach, & Kellen. *System Development Methodologies for Web Enabled E-Business: A Customization Paradigm*. November 1, 2001. <http://www.kellen.net/SysDev.htm>

Lewis, Pamela S., et al. *Management – Challenges in the 21st Century*. 3rd ed. Ohio: South-Western College Publishing, 2001.

Maner, Walter. Rapid Application Development Using Iterative Prototyping. Revised March 15, 1997. Copyright © 1997 Boston University.
<http://www.cs.bgsu.edu/maner/domains/RAD.htm>

Okon, Walt, *Computer Engineering - Three Approaches To Software Development*. 16 September 2002.
<http://www.waltokon.com/ComputerEngineerinEvolutionaryApproachToSoftwareDevelopment.html>

Oz, Effy. *Management Information Systems*. 4th ed. Pennsylvania State University, Great Valley: Course Technology, 2004.

Pitagorsky, George. Systems Development Life Cycle – Outline. Copyright © 1998 Pitagorsky Consulting. <http://www.pitagorskyconsulting.com/SDLC.htm>

Sews, C. *"Spiral Development Model:" Candidate Definition.* May 12, 2000. <http://www.sei.cmu.edu/cbs/spiral2000/Boehm/>

Spiral Model. Datascovery. Copyright © 2001, Datascovery. <http://www.datascovery.com/CompSpiral.html>

Whitten, Jeffery L., et al. *Systems Analysis and Design Methods*. 6th ed. Indiana: McGraw-Hill/Irwin, 2004.

Upchurch, Richard. *Software Lifecycles*. January 26, 1997. <http://www2.umassd.edu/CISW3/coursepages/pages/cis311/LectureMat/process/lifecycles/lifecycles.html>