

CS 107 Fall 2006

Lecture 14: Systems Analysis Chapter 10

1 The Impetus for Change

A new system is always costly and disruptive; a large, complex, new system is extremely so. However, organizations large and small regularly purchase or create new systems. What motivates them to undertake the trouble and expense?

- The business is starting a new branch or a new factory.
- Someone has a creative new idea about how the business can grow.
- The current system cannot meet the new demands placed upon it.
- The current system is inefficient or inconvenient and many errors happen.
- The current system is breaking down and must be replaced by something.
- There are new government regulations.
- The competition has installed modern systems and the company needs to keep up.

2 The Process of Change

There are common stages in any major development in any area (not just computing). These stages are listed, below, for the process of building the Hamden High School (1990-2000).

- Investigate the need for change. The building is shabby, the roof leaks, many things don't work. Parts of the building are not handicapped-accessible. Employees and students complain.
- Capture the requirements. A citizen-teacher committee establishes a very long list of shortcomings. But but the Town does not want to spend any money. NEASC denies accreditation. The Town agrees to spend the money, and lets the Board of Education hire an architect.
- Analyze the situation. The architect takes the list of needs (40 pages) and refines the list. Major options are presented, and one is chosen. (Rebuild on the same site.) An approximate price and time-line for the project is established.
- Design a solution. The architect submits preliminary plans. A construction management firm is hired, and works with the architect to improve building function and reduce the cost of the facility. A series of plan modifications follows, until one plan is ready for bidding. After the exterior design was largely settled, an interior designer began to select patterns, tiles, finishes, colors, etc..
- Implement the solution. The plan is put out to bid and a general contractor is interviewed and selected. The construction contract is signed, allowing 10% of the total price for changes and unexpected problems. The plan is built over a period of three years. During construction, many small changes happen.

- Install the product and train workers. Over a two-year period new parts of the building are occupied. Then the portable classrooms (in use for 25-years) are demolished. Students and faculty have orientation sessions. A “punch-list” of incomplete details is assembled by school personnel and fixed by the builder.
- Maintain the system. Ah.... we are back to normal; maintenance is a local responsibility.

In this process, no one stage was completely finished when the next stage started. The list of requirements was refined and modified when the architect and construction manager worked with the school staff to address problems that they identified. The architect worked closely with the construction manager, who worked closely with the builder, who worked closely with subcontractors. Problems arose and were addressed by changes in the plan.

3 Software Systems Development

Change happens. That is true in building a school and in building a software project. The larger the project is, the longer it will take, and the more likely change becomes. Any of the reasons given above for *starting* a project can become a reason for *changing* the specification six months or a year later. The old traditional SDLC does not allow for change; it is no longer an acceptable or reasonable way to develop a software product.

Consider the phases of system development again, in the context of a software project.

- Investigate the need for change. Management decisions in the software systems area are much like the decisions made in any other area: the manager needs to know that the expected benefit of change is greater than the expected cost. There must be a first exploratory stage to establish this fact, during which some systems expert provides information about cost and benefit. The result will be a vision statement that forms the first understanding between client and developer, and a contract to continue with the first stage of development.
- Capture the requirements. The initial vision statement will list features that make sense and others that do not make sense. However, until further exploration is done, you cannot tell the difference. The development team uses experience, brainstorming, and use cases to produce a complete requirements document.
 - Functionality.
Local functionality: use cases (documents that tell a story of how you use one piece of the system). Global functionality: maintaining proper system security and availability.
 - Useability.
Appearance of the interface, Information available, Speed of data entry, Hand movement, Colors appropriate to the uses.
 - Reliability.
What happens when power goes out in the middle of a transaction. Backups, Disaster recovery, What will happen when the system malfunctions? How long will it take to fix the problem? Availability – what about down-time?
 - Performance.
Scale: How much data must be handled? Speed: How fast must the system respond?
 - Supportability.
How long to find and fix errors? Extensibility – how hard will it be to add features: Can the system handle the growth of traffic and data? Is later optimizations possible? Will the system still work the next time Windows changes?

- Analyze the situation. Give the user feedback.
- Design a solution. Give the user feedback.
- Implement the solution. Give the user feedback.
- Install the product and train workers. Give the user feedback.
- Maintain the system. Ah... we are back to normal; maintenance is a local responsibility.

Communication and Agile Development In the beginning, the client has an idea and

4 Government Projects are Distorted by Red Tape

- The RFP process and secret bidding can award the contract to an unqualified business.
- Secrecy in the bidding process makes public scrutiny impossible. Lots of “deals” can happen.
- Massive documentation requirements are burdensome.
- Back-propagating changes to the original plan is expensive and impedes needed change.