IS 300 — Lectures 14/15

- Systems development – why is a formal process important?
- What is the traditional systems development life cycle (SDLC)?
- How does this life cycle begin – the systems investigation stage.
- How are the functional specifications developed – the systems analysis stage.
- How are the system specifications developed – the systems design stage.
- What activities take place during systems implementation and maintenance?
- What is process reengineering and how does it relate to the traditional SDLC?
- How does prototyping differ from the traditional SDLC?

Systems development – why is a formal process important?

Without a formal process, one tends to …

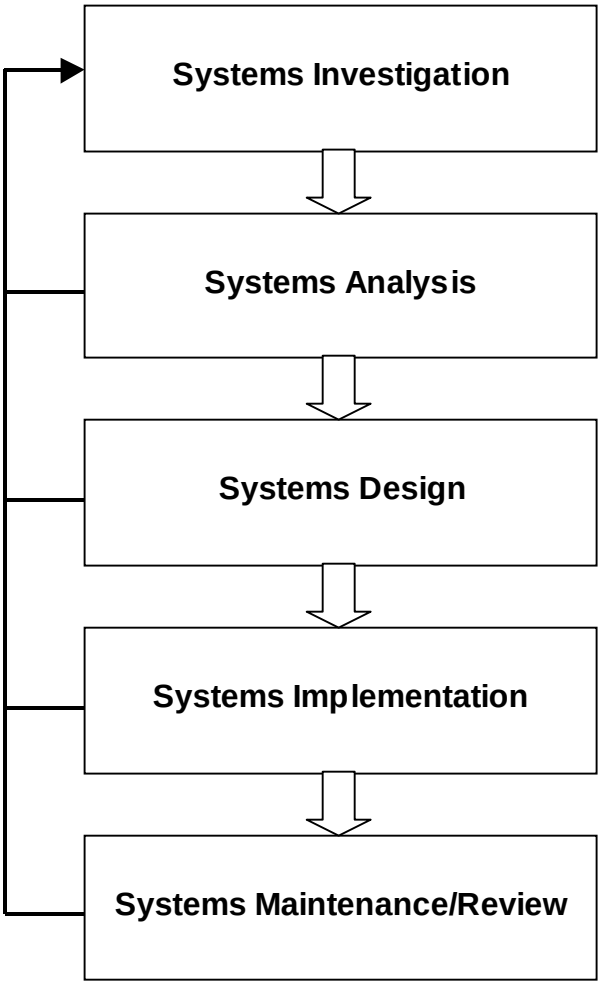
- Skip or hurry through problem investigation and analysis – a natural tendency and the primary reason for system failure

- Put too much reliance on the vendor. So what's wrong with that?

  - They hurry you through the steps to get to the sale
  - They may distort requirements to meet their solution
  - They are reluctant to encourage you to look at other alternatives

What is the traditional systems development life cycle (SDLC)?

See Figure 12.6



How does this life cycle begin – the systems investigation stage.

Who – end users and systems analysts

Why

- Identify areas of concern/opportunities
- Understand the problem (not the symptoms)
- Engage in a high-level discussion of solution alternatives
- Recommend how to proceed

The Feasibility Study – a "quick" attempt to outline solutions to the problem and assess potential feasibility problems.

Organizational (Political) Feasibility. Does the solution "fit" within organization's plan, policies, culture? [Note: not in text]

"Too many scalpels" scenario

Technical Feasibility. Does existing technology exist? Can it be developed? What are the risks?

"Speech recognition and MD" scenario

Economic Feasibility. Cost/benefit analysis (but usually with rough estimates). Efficiency. [Note: how do we justify effectiveness?]

Operational Feasibility. Will the solution work in the specific workplace? A physical consideration. [Note: text combines with organizational]

"Computer on shipboard" scenario

Schedule Feasibility. Can the system be implemented by a specific (real) deadline?

Contraints

- Existing hardware/software/databases
- "Approved" hardware/software lists
  - Support
  - Price (quantity discounts)
  - Compatibility

How are the functional specifications developed – the systems analysis stage.

Who – end users and systems analysts

What – an in-depth study of the system and development of functional specifications

Overview – build system models

	Current System	New System
Physical Model	Step 1	Step 4
Logical Model	Step 2	Step 3

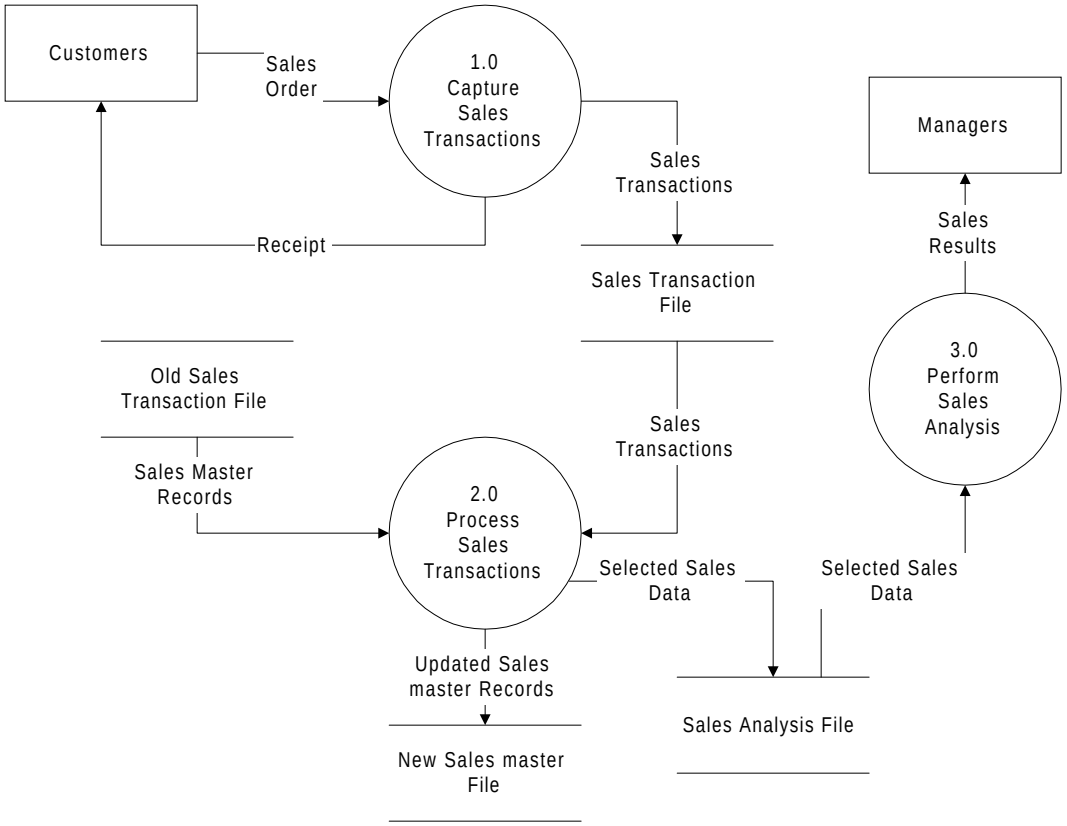
Physical Model – How (with technology)
Logical Model – What (regardless of the chosen implementation technology)

Systems analysis stops after Step 3 above.

Model Captures

- Process that transform the data (Dataflow Diagram) – see next page.
- Data used by the system (Data Dictionary)
- Data relationships (Entity-Relationship Diagram)
- Rules or policies for performing the transformations
- Control issues
- Interfaces with other systems

Dataflow Diagram:



How are the system specifications developed – the systems design stage.

Who – IS system specialists (large systems)
Users (medium and small systems)

What – transform new logical model into a new physical model (add technology)

System Specifications – used for

- software development/acquisition
- hardware acquisition
- system testing

Deals with

- User interface design – screens, forms, reports
- Data design – Record Structure Diagrams and integrity rules
- Process design – detailed specifications for software modules

What activities take place during systems implementation and maintenance?

Modify purchased SW if necessary (source code required)

Integrate components

Train users

Test system

- unit tests
- system (integration) tests
- procedure tests (backup/recovery)
- documentation

Conversion – the final implementation step

Parallel – low risk, high cost

- may not be feasible (STAR)
- "attitude" that you can fall back on old system may be detrimental

Direct (plunge) – low cost, high risk

- users forced to make system work
- benefits available immediately

Pilot – a subset of organization/whole system

- provides experience when whole organization gets system
- may not detect some area-specific problems

"Pay 'n Save" example

Phased – a subset of system/whole organization

- easier to test subsystem
- real benefits delayed

"Lamonts" example

Note: Within phased or pilot, you can use parallel or direct.

Systems Maintenance

Defined as

- Corrective – fix errors (20%)
- Adaptive – adapt to changes brought on by changes in other systems (20%)
- Perfective – user enhancements (can cause adaptive maintenance in other systems) (60%)

Maintenance typically accounts for 50% - 75% of the total life cycle costs.

Designing for maintainability is very important

What is process reengineering and how does it relate to the traditional SDLC?

Reengineering (Hammer and Champy)

"The fundamental rethinking and radical redesign of business processes to achieve dramatic improvements"

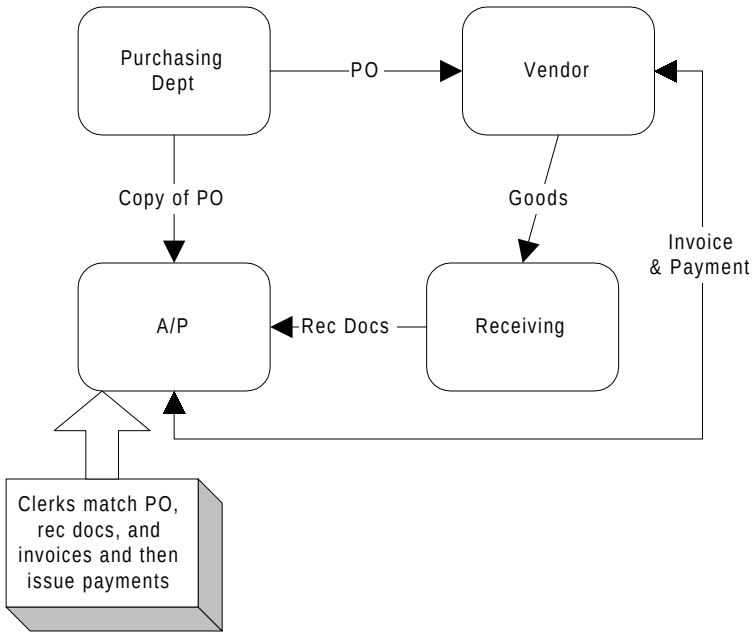
Fundamental – Why do we do what we do?

> "How can we perform credit checks more efficiently?" This assumes that we should be doing it in the first place.

Radical – Reinvention

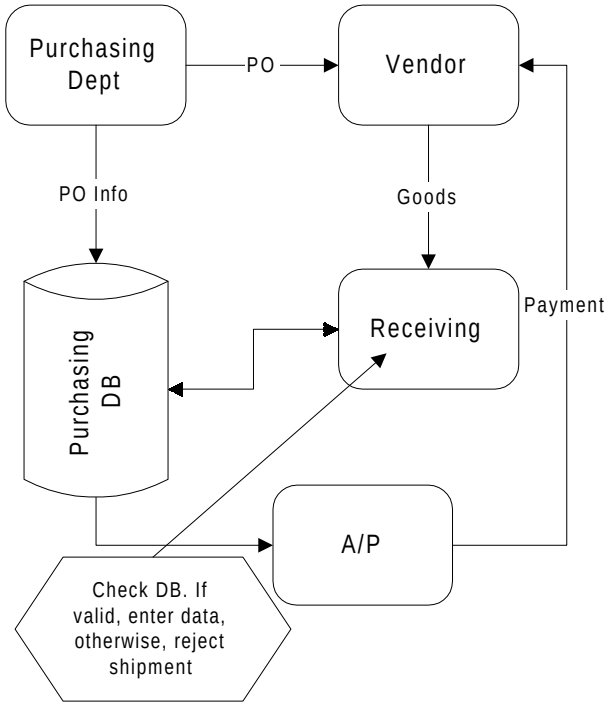
Dramatic – Not marginal

Ford's traditional A/P approach (very common)



Adding better technology allowed Ford to reduce A/P staff from 500 to 400 (20% reduction).

The “reengineered” solution (invoiceless processing):



By reengineering and adding technology, Ford reduced staff size to 125 (75% reduction).

How does prototyping differ from the traditional SDLC?

An alternative to traditional SDLC – see Figure 12.8

Traditional SDLC – characterized by

- Clear stages/responsibilities
- Good for
  - complex problems
  - problems where requirements can be prespecified
  - stable requirements
- Not all problems are characterized like this (which aren't?)
- Formal methodologies (to handle complexity)

Prototyping

Characteristics

- Requirements determined dynamically
- More interaction with users
- Good when requirements are (initially) hard to define
- Requires special "software engineering" tools to create prototypes quickly and easily

Used to deal with uncertainty

- Demonstrating concept feasibility (systems investigation)
- User requirements not clear (systems analysis phase)
- Design approaches need further evaluation (systems design phase)
- EIS demo from earlier lecture was an example of a prototype