Systems Analysis & Design: Modeling the Real World
Systems Analysis and Design (SAD)

• When candidate IT applications are proposed and approved, how do you go about making them a reality?

• SAD is all about going from “Yes, we need an IT” to a completed IT application that meets the needs of the business.

• SAD is the complex organizational process whereby computer-based information systems are designed, developed, deployed, and maintained.
Why SAD?

Assume: We have lots and lots of resources (e.g., $) at our disposal.
Systems Analysis and Design (SAD)

• Achieving successful IT applications involves more than just “let’s just go build/buy it”
  – Many questions to be addressed
    • Why is the system needed?
    • What are the needs and problems to be addressed?
    • Who are the stakeholders?
    • What processes are involved?
    • ....

  – SAD begins with these questions in mind and tries to answer them before proceeding with application development
• “…I think people are focusing too much on the IT technical aspect and sometimes they forget to spend time with the user, for this case, the system was not tested with the front line operator until quite the last minute”
  – Sauder MBA alumni

• “Building the workflows behind each [IT] request also proved challenging – in essence, this was one of the biggest benefits -and the start of having teams work consistently, share knowledge, and improve the efficiency of their team processes.”
  – Sauder MBA Alumni
• “…gathering the buy-in of all users and data contributors of this new system is essential, so I was required to understand the needs and concerns of all relevant parties involved.”
  – Alex Lai

• “Adopting this new software and integrating this to our current workflow pose challenges most of which are focused on setting it up and getting the right information on it in order to have accurate output.
  – Lynne Gonzaga
“Although there are numerous reasons why IS projects fail, poorly defined requirements are prominent for IS failures. Systems Analysis and Design books…indicate that defining requirements is one of the first activities that occurs…to provide a systematic approach for developing an IS. Poorly defined requirements introduce error in the IS project at a very early point in the project’s life. This early introduction of error is cascaded through the rest of the development process until corrected. Unfortunately, any decisions made on poorly defined requirements are costly to change at later stages of the IS project, particularly as the project is closer to completion. One study reports that poorly defined requirements are responsible for over one-third of excessive spending during IS projects (Ellis 2009). Hence, the activities that occur during requirements definition of IS projects need careful attention.”

Binny Samuel, Assistant Professor, U. of Cincinnati
(Doctoral Dissertation 2012)
Systems Analysis and Design (SAD)

- SAD typically relies on various *formal* methodologies, techniques, and tools
  - METHODOLOGIES: Comprehensive, multiple step approaches to systems development that gather requirements, guide work and influence quality of the final product
  - TECHNIQUES: Particular processes that analysts follow to ensure that their work is complete and comprehensive to others on the project team (EX: Entity Relationship Diagrams)
  - TOOLS: computer programs that make it easy to use and benefit from the techniques of the development methodologies (Ex: CASE)
Systems Analysis and Design (SAD)

• Formal approaches to SAD
  – There are many!
    • We’ll cover a few types today, but knowing these specifically (e.g. Unified Modeling Language) is not necessarily important
  – Many formal approaches share the “life cycle” framework: Plan, Analyze, Design, Implement, Maintain…
## Systems Development Life Cycle (SDLC)

<table>
<thead>
<tr>
<th>SDLC Phase</th>
<th>Activities</th>
</tr>
</thead>
</table>
| 1. Plan    | • Define the system to be developed  
• Set the project scope  
• Develop the project plan including tasks, resources, and time frames |
| 2. Analysis| • Gather the business requirements for the system |
| 3. Design  | • Design the technical architecture required to support the system  
• Design system models |
| 4. Develop | • Build the technical architecture  
• Build the database and programs |
| 5. Test    | • Write the test conditions  
• Perform the testing of the system |
| 6. Implement| • Provide training for the system users  
• Write detailed user documentation |
| 7. Maintain| • Build a help desk to support the system users  
• Provide an environment to support system changes |
Another view of SDLC

<table>
<thead>
<tr>
<th>Stage</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Identification &amp; selection</td>
<td>Priorities for systems and projects; an architecture for data, networks, hardware, and IS management is the result of associated systems planning activities</td>
</tr>
<tr>
<td>Project initiation and planning</td>
<td>Detailed steps, or work plan, for project; specification of system scope and high-level system requirements or features; assignment of team members and other resources; system justification or business case</td>
</tr>
<tr>
<td>Analysis</td>
<td>Description of current system and where problems or opportunities are with a general recommendation on how to fix, enhance, or replace current systems; explanation or alternative systems and justification for chosen alternative</td>
</tr>
<tr>
<td>Logical Design</td>
<td>Functional, detailed specifications of all system elements (data, processes, inputs, and outputs)</td>
</tr>
<tr>
<td>Physical Design</td>
<td>Technical, detailed specifications of all systems elements (programs, files, network, system software, etc.); acquisition plan for new technology</td>
</tr>
<tr>
<td>Implementation</td>
<td>Code, documentation, training procedures, and support capabilities</td>
</tr>
<tr>
<td>Maintenance</td>
<td>New versions or releases of software with associated updates to documentation, training, and support</td>
</tr>
</tbody>
</table>
Yet another view of SDLC

• Waterfall model

(Also known as “Analysis”, or “Conceptual Modeling”)
Other approaches to SAD

• **Prototyping**: designing and building scaled down but working version of the desired system (iterative rather than linear as in waterfall model)

• **Joint Application Design (JAD)**: Users, managers, and systems developers are brought together for a series of intensive structured meetings to specify and review system requirements
Other approaches to SAD (cont)

- **Participatory Design (PD):** Development that emphasizes the role of the user

- **Rapid Application Development (SDLC)**
  - Requirements planning, user design, construction, cutover

- **Agile Development** (the “Manifesto”, Beck et al. 2001)
  - Individuals and interactions over processes and tools
  - Working software over comprehensive documentation
  - Customer collaboration over contract negotiation
  - Responding to change over following a plan
Other approaches to SAD (cont)

• **YAMA** or Yet Another Modeling Approach.
  – UML, MERISE, SSM, STRADIS, YSM, ISAC…
    • (Don’t worry, you don’t have to know these)
  – The number of formal systems development methods and techniques grow by the day

• The Reality:
  – **What method do you use to develop IT apps?**
  – Many firms do not use formal methods, or use their own “home grown” approaches
The Evolution of Formal SAD

SAD evolves: Agile (new) vs. Waterfall (old)

(Beck et al. 2001)

Courtesy of Dr. Michael Wufka
A Closer Look at SAD: Analysis & Design

<table>
<thead>
<tr>
<th>SDLC Phase</th>
<th>Activities</th>
</tr>
</thead>
</table>
| 1. Plan    | • Define the system to be developed  
• Set the project scope  
• Develop the project plan including tasks, resources, and time frames |
| 2. Analysis| • Gather the business requirements for the system |
| 3. Design  | • Design the technical architecture required to support the system  
• Design system models |
| 4. Develop | • Build the technical architecture  
• Build the database and programs |
| 5. Test    | • Write the test conditions  
• Perform the testing of the system |
| 6. Implement| • Provide training for the system users  
• Write detailed user documentation |
| 7. Maintain| • Build a help desk to support the system users  
• Provide an environment to support system changes |

(Also known as “Analysis”, “Conceptual Modeling”)
An Information System is a Model of the Real World
An Information System is a Model of the Real World

• What is a model?
  – “... a model is a conceptual abstraction of an existing or proposed real system that captures the characteristics of interest of the system.”
    • Taylor & Tofts, HP Laboratories Bristol (HPL-2003-246)

• Models can vary in the degree to which they represent reality. However, the tradeoff to greater accuracy is greater cost and complexity
  – Ex: (Paper) Maps are models of reality
    • A perfectly accurate map of Canada would be the size of Canada
An Information System is a Model of the Real World

• Representation theory (Wand & Weber*)
  – Information systems [as a model of reality] exist because "it is the human condition to seek better ways to understand and to represent the world" (Weber 1997, p. 59).

  – Premise: The basic function of all information systems is that they help individuals to understand the states of some real world systems that are relevant to them
    • e.g., states of their mind, states of their organization, states of their organization's environment
    • Ex: Inventory system versus counting boxes by hand


Slide courtesy of Prof. Andrew Burton-Jones
An Information System is a Model of the Real World

- Three types of information (and information systems):
  - information about reality, which tells us about the world
    - e.g., database systems, word processing systems
  - information for reality, which helps us act in the world
    - e.g., planning systems, decision support systems, BI systems
  - information as reality, which presents itself to us directly, as if it were reality
    - e.g., video conferencing
With that in mind…

A Closer Look at Analysis & Design

---

### SDLC Phase

<table>
<thead>
<tr>
<th>SDLC Phase</th>
<th>Activities</th>
</tr>
</thead>
</table>
| 1. Plan    | • Define the system to be developed  
             • Set the project scope  
             • Develop the project plan including tasks, resources, and time frames |
| 2. Analysis| • Gather the business requirements for the system |
| 3. Design  | • Design the technical architecture required to support the system  
             • Design system models |
| 4. Develop | • Build the technical architecture  
             • Build the database and programs |
| 5. Test    | • Write the test conditions  
             • Perform the testing of the system |
| 6. Implement| • Provide training for the system users  
             • Write detailed user documentation |
| 7. Maintain| • Build a help desk to support the system users  
             • Provide an environment to support system changes |

(Also known as “Analysis”, “Conceptual Modeling”)
The “Big Picture”: Conceptual and Logical Modeling

Conceptual Model

Data Modeling

Logical Model

Real World

Correct

Physical Model

The “System”

Complete

Implementation
Linking Conceptual & Logical Modeling with SAD

• Conceptual modeling is typically done at the “analysis” stage of SAD.
• Logical modeling is typically done at the “design” stage.
Conceptual v. Logical v. Physical Modeling

• **Conceptual Modeling**: Making the real world *explicit* (words, symbols, diagrams, etc)
  – People, “stuff”, and processes in the real world
    • Relationships, states, events, etc. associated with above
  – Firms struggle with this part of Systems Analysis
    • Cutting-edge research into this being conducted by Sauder Professors Wand, Woo to develop tools to do CM similar to tools for logical design
Conceptual v. Logical v. Physical Modeling

• Conceptual Modeling:
  – “Activity of formally describing some aspects of the physical and social world around us for the purposes of understanding and communication” (Mylopoulos 1991)
  – Conceptual models may or may not be created explicitly. Even if they are not, analysts/developers must have some rudimentary form of conceptual model in their minds in order to be able to implement anything
  – Clearly, without a mental model of the domain it is impossible to correctly implement requirements

Courtesy of Dr. Michael Wufka
Conceptual v. Logical v. Physical Modeling

- Conceptual models are visual artifacts used to help individuals understand a domain that is (or will be) supported by an information system.
- They are usually associated with systems analysis but can be used by many people, at any time in an IS project, and even in non-IS projects (e.g., business process modeling).
- They are used because understanding organizational domains is necessary for many tasks, but difficult!
Conceptual Model: Example 1

Quick Deposit Teller

customer

Jack

Carol

deposit money

Courtesy of Dr. Michael Wufka
Conceptual Model: Example 2

Demand Reported ➔ Customer

Write up Purchase order ➔ Order accepted ➔ Order checked

Customer Order ➔ Sales

Check order

Customer credit

Credit worthiness

inventory

Checked Order

Courtesy of Dr. Michael Wufka
Conceptual Model: Example 3

From “Ontology-Based Rules for Object-Oriented Enterprise Modeling”
Yair Wand and Carson Woo
Conceptual Model: Example 4

Mortgage Issuer
- A1. Approve mortgages
  - Perceived affordable mortgage products
- G1. Get mortgage transactions
  - Mortgages Price premium

Conduit
- A1. bundle mortgages into tranches
- G1. create tranches with good rating
  - Supplies of mortgage tranches Payment premium

Home buyer
- A1. receive an attractive offer
- A2. work to get income
- G1. Get a mortgage
- G2. Own a home

Making payment
- Threat of foreclosure and loss of equity in the house

Investment Bank
- A1. get mortgage tranches
- A2. borrow fund to securitize the mortgages
- A3. work with other financial institutions & investors
  - Secured loan contract
- G1. make/sell/trade MBSs

Making payment
- Threat of foreclosure and loss of equity in the house

Other Banks
- A1. ensure the ability to receive payments on the loan
- G1. earn back on their investment

Other financial institutions & investors
- A1. determine the asset's potential for a good ROI by looking for a good rating
  - Buying MBSs Products with good ROI
- G1. earn a good ROI and/or reduce cost of regulatory capital

Woo, Limonad, Monu & Lemieux
The Dual Role of Conceptual Modeling

- Conceptual Models can be used to:
  - Design, and maintain or evolve information systems
  - BUT, also to support organizational workers in their decision-making
- Ergo, conceptual models can provide value to users (not just analysts and developers)
Conceptual v. Logical v. Physical Modeling

• **Logical Modeling:**
  – Bridging the conceptual model with the physical “software code”

• Whereas Conceptual Modeling focuses on the “why” and semantics of the world, **Logical Modeling** focuses on the “what” and syntactics
  – Data tables, data fields, relationship among defined entities

• Is the distinction between CM and LM fuzzy?
  – You bet it is
Example of a portion of a Logical Model
Conceptual vs. Logical vs. Physical Modeling

• The Logical Model allows for immediate representation in the physical model (e.g. the software code)

• The Logical Model is independent of the physical model
  – The preceding “doctor/patient” Logical Model could be coded in any available database (Access, MySQL, etc.)
Conceptual Modeling Importance

• Conceptual modeling is often **not** performed in business
  – Why? It’s difficult, requires specialized techniques, and professionals with the requisite expertise (**such as those who specialize in ITM!**)

• **BUT**: Conceptual modeling is a vital step in information system design and implementation
  – This is where **process improvements** are identified
  – **Otis Elevator** will demonstrate the importance of first modeling processes and avoiding “automating a mess”
SAD: Bottom Line

- Regardless of what formal/informal SAD approach is used, the focus of SAD should be on modeling the real world within a working and stable IS.
- The role of the IT manager should be on reducing the “gulf” between the real world and the IS.
Norman’s Gulfs of Execution and Evaluation

• Translating a user’s goals into a system action and the user perceiving and evaluating that system action.

Donald A. Norman “The Design of Everyday Things” 1988
Norman’s 7 stages of action

- The User...
- Translating goals...
- To actions...
- Based on the user’s model
Norman’s 7 stages of action

- System presents the results back to the user
- According to the design model
The Two Gulfs
Where thought is required

- **Gulf of execution** -- thinking required to figure out how to get something done -- transforming high-level intention into specific physical actions

- **Gulf of evaluation** -- thinking required to understand what is being perceived -- transforming raw sensory data into an understanding of objects, properties and events
The Two Gulfs simplified

• Gulf of execution
  – How do I do it?
• Gulf of evaluation
  – What did it do?
Gulf of Execution

Type "<bold>" then "</bold>"
Gulf of Execution Issues
Zero Gulf of Execution?

http://www.ted.com/talks/tan_le_a_headset_that_reads_your_bra...
Gulf of Evaluation

Two showers:
Which one do you have more confidence of the water temperature?
Gulf of Evaluation

Two Crosswalks:

http://jzyo.blogspot.com/
Bridging the Gulfs

• The goal of SAD is to produce good IT design

• The purpose of good IT design is to reduce the “gulfs” of execution and evaluation, thus leading to IT being a more faithful model of the Real World
Regardless of the tools, techniques, methods... The true end goal is:

- Conceptual Model
- Logical Model
- Real World
- Physical Model (The “System”)

Data Modeling → Implementation

Correct ≈ Complete
Conclusion: Systems Analysis and Design

• Why should I care?

  – SAD provides for a formalized and detailed means of going from the “real world” to an IT System that...
    • Best represents that real world and is a model of that world
    • Meets the needs of the business and the users
      – Reduce the “gulfs” between business needs and IT
    • Is robust and of high quality
    • Succeeds, not fails as many IT systems do

  – SAD attempts to bridge the real world (which is vague) with an IT (which is highly precise)