Topic # 11

Design Concepts.
Selecting the Best Alternative Design Strategy

Design Phase of CIS Life Cycle
Design

Design - is a meaningful engineering representation of something that must be built (i.e. for engineers and/or developers).

In CIS context, CIS Design focuses on the following main areas:

1. **Architecture**
   - architectural design defines the relationships between major structural elements of the CIS

2. **Components**
   - component-level design transforms structural elements of the CIS architecture into a procedural description of CIS components (based on control specifications, process specifications and state transition diagram).

3. **Data**
   - data design transforms the information domain model into the particular data structures

4. **Interfaces**
   - interface design describes how the CIS communicates within itself, with other systems in the environment, and with users

---

**Design Model as a Bridge**

- **Analysis**
- **Design**
- **Development**
FROM Analysis Model TO Design Model

Analysis model (DOs, DFDs, ERDs, STDs, activity diagrams, use cases, etc.)

Maps into

Design model (architecture, components, data, interfaces)

Class objects and processing units

Software modules

CIS Design: Fundamental Concepts

1. Abstraction  data, procedure, control
2. Architecture  the overall structure of the software
3. Patterns     "conveys the essence" of a proven design solution (best cases, re-usable designs)
4. Modularity   compartmentalization of data and function
5. Information Hiding  controlled interfaces
6. Functional Independence  single-minded function and low coupling
7. Refinement  elaboration of details for all abstractions
8. Re-factoring a reorganization technique that simplifies the design
1.1. Data Abstraction
(abstractive data models not actual objects)

Abstraction - allows designers to simplify a problem and focus on solving a problem without being concerned about irrelevant lower level details

1.2. Procedural Abstraction (solution algorithms)

Abstraction - allows designers to simplify a problem and focus on solving a problem without being concerned about irrelevant lower level details

(en example: procedural abstraction: a subroutine – a named sequence of actions and events)
2. Architecture: Webster CIS - A Structural Design (system, subsystems, and component design)

- **System Level** (Webster System)
- **Level of Subsystems (Domains)** (Databases, GUI, Security, HELP, etc.)
- **Level of Elements or Components** (tables, forms, queries, reports, macros and modules, ...)
- **Level of Sub-elements, Details** (for ex., attributes) (ID, First Name, Last Name, DOB, YOA, status, ...)

3. Patterns
(Reusability of previous designs and design solutions)

**Design Pattern Template:**
- **Pattern name** describes the essence of the pattern in a short but expressive name
- **Intent** describes the pattern and its functions
- **Also-known-as** lists any synonyms for the pattern
- **Motivation** provides an example of the problem
- **Applicability** notes specific design situations in which the pattern is applicable
- **Structure** describes the classes that are required to implement the pattern
- **Participants** describes the responsibilities of the classes that are required to implement the pattern
- **Collaborations** describes how the participants collaborate to carry out their responsibilities
- **Consequences** describes the "design forces" that affect the pattern and the potential trade-offs that must be considered when the pattern is implemented
- **Related patterns** cross-references related design patterns

*) Patterns-based SE, patterns-based analysis, patterns-based software development, pattern languages, etc. – still a lot of research needed, a lot of Ph.D. dissertations
Patterns: Analogy in Construction Engineering
(reusability of previous designs and design solutions)

Example of Highly Reusable CIS
Selecting the Best Alternative Design:

Selecting the best alternative design involves at least two basic steps:

a) generating a comprehensive set of alternative design strategies, and

b) selecting the one that is most likely to result in the desired information system, given all of the organizational, economic, and technical constraints.

“One Design Strategy” Case: Disadvantages

1) NO GUARANTEE that the best, correct, or even an adequate system for the situation is being developed or purchased. This is not obvious because it is unclear if other alternatives were considered, and if they were, those present cannot see why the one choice won out.

2) ONE VENDOR – ONE PRICE. If the one strategy is chosen because only one vendor is used, there are no benefits from having multiple vendors compete for an RFP. For example, the vendor has no incentive to keep his price as low as possible.

3) NO PUBLIC DETAILED SPECIFICATIONS. Without the detailed, public systems specifications that are part of a competitive bid process, there is not likely to be much in the way of written documentation to refer back to if the vendor does not fulfill his promises.

If the analysts present only one design strategy during the oral presentation to the project steering committee or client, the recommendations are likely to be (at worst) rejected, or (at best) accepted with great skepticism.

It is also possible that those present at the meeting will start to generate alternatives, each representing that person’s position.

The meeting will quickly deteriorate since a fair assessment of ad hoc alternatives cannot be done within the limits of a meeting.

In any event, this is not a good way to begin the development of an information system (or build a career).
Design Strategy

- The primary deliverables from generating alternative design strategies and selecting best one are:
  1. at least three (better – 5) substantially different system design strategies,
  2. a design strategy judged most likely to lead to the most desirable information system, and
  3. A Baseline Product Plan for turning the most likely design strategy into a working information system.

  Ex: 1) You want to buy a house. What are your alternative approaches?
  2) You want to replace... a computer, a furnace, a water-heater, carpeting, roofing, etc.

Generating Alternatives

Issues to Consider in Generating Alternatives

- System analysts should consider at least two cases:
  1. which design strategies would minimally satisfy objectives and not violate constraints, and
  2. which design strategies would meet or exceed objectives with minimal violation of constraints.

  There are many possible design strategies between these two extreme positions.

- Analysts should generate three alternatives because three alternatives can represent both ends and the middle of a continuum of potential solutions.

  Ex: Expensive    Middle    Cheap
      High Quality  Average    Low Quality
**Selection Criteria**

**Criteria to be used:**

The following selection criteria can be used in order to select best alternative:

1. cost,
2. functionality,
3. vendor support,
4. viability of vendor,
5. flexibility,
6. documentation,
7. response time, and
8. ease of installation.

Selecting the Most Likely Alternative

**Selecting Most Likely Alternative**

- The method for evaluating alternatives is called Weighting and Scoring.

  1. Create a table with (a) requirements, (b) constraints, and (c) alternative solutions.
  2. Weights for requirements and constraints
  3. Ratings for alternatives (rating of 5 indicates that the alternative meets or exceeds the requirement or clearly abides by the constraint).

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Weight</th>
<th>Alt A</th>
<th>Alt B</th>
<th>Alt C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements</td>
<td></td>
<td>Rating</td>
<td>Score</td>
<td>Rating</td>
</tr>
<tr>
<td>Real-time data entry</td>
<td>10</td>
<td>5</td>
<td>50</td>
<td>5</td>
</tr>
<tr>
<td>Auto re-order</td>
<td>10</td>
<td>3</td>
<td>30</td>
<td>5</td>
</tr>
<tr>
<td>Real-time data query</td>
<td>10</td>
<td>1</td>
<td>10</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>90</td>
<td>130</td>
<td>150</td>
</tr>
<tr>
<td>Constraints</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Development costs</td>
<td>25</td>
<td>5</td>
<td>125</td>
<td>4</td>
</tr>
<tr>
<td>Hardware costs</td>
<td>25</td>
<td>5</td>
<td>125</td>
<td>4</td>
</tr>
<tr>
<td>Time to operation</td>
<td>15</td>
<td>5</td>
<td>75</td>
<td>4</td>
</tr>
<tr>
<td>Ease of training</td>
<td>5</td>
<td>5</td>
<td>25</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>350</td>
<td>275</td>
<td>245</td>
</tr>
<tr>
<td>Total</td>
<td>100</td>
<td>440</td>
<td>405</td>
<td>395</td>
</tr>
</tbody>
</table>
## Weighted Scoring Model Steps

WSM main steps:
1. First identify criteria important to the project selection process
2. Then assign weights (percentages) to each criterion so they add up to 100%
3. Then assign BY YOURSELF scores to each criterion for each project (based on calculated numeric values of NPV, ROI, IRR, PTP, etc.)
4. Multiply the scores by the weights and get the total weighted scores

### New PC Purchase

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Weight</th>
<th>Computer A</th>
<th>Computer B</th>
<th>Computer C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements</td>
<td></td>
<td>Rating</td>
<td>Score</td>
<td>Rating</td>
</tr>
<tr>
<td>Speed</td>
<td>15</td>
<td>3</td>
<td>45</td>
<td>4</td>
</tr>
<tr>
<td>Storage</td>
<td>15</td>
<td>3</td>
<td>45</td>
<td>4</td>
</tr>
<tr>
<td>Ease of Use</td>
<td>15</td>
<td>4</td>
<td>60</td>
<td>5</td>
</tr>
<tr>
<td>Reliability</td>
<td>15</td>
<td>4</td>
<td>60</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>60</td>
<td>210</td>
<td>270</td>
</tr>
<tr>
<td>Constraints</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Costs</td>
<td>25</td>
<td>5</td>
<td>125</td>
<td>4</td>
</tr>
<tr>
<td>Time to Operation</td>
<td>15</td>
<td>5</td>
<td>75</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>40</td>
<td>200</td>
<td>175</td>
</tr>
<tr>
<td>Total</td>
<td>100</td>
<td>410</td>
<td>445</td>
<td>425</td>
</tr>
</tbody>
</table>
WSM Method and Selecting the Best Alternative Design Strategy: In-Classroom Exercise