**ISDS 406 --- Final Exam Study Guide**

**Logistics**: Unless you have arranged a special agreement with the instructor, you must take the exam on the date and time assigned to your class section.

The exam will be administered via TITANium, but you must take the exam using one of the classroom computers. You may not use any notes or computer programs. You may not sit next to someone on your team.

**Point value:** The exam is worth 175 points.

**Breakdown**: 85% multiple choice + 15% Free Response

**Multiple Choice portion (37 questions):**

* 4 questions from Chapter 7 (Design Phase / Acquisition Strategies). You will need to know:
  + What are the steps of the design phase, and what is the final deliverable of the design phase?

1. Determine system acquisition strategy (make, buy, or outsource) – alternative matrix
2. Determine the technical architecture for the system – architecture design
3. Address security concerns and globalization issue –
4. Make hardware and software selections – hardware and software specification
5. Determine the way that users will interact with the system (interface, inputs, and outputs) – interface design
6. Design the programs for the underlying processes – program design specifications
7. Design the way data will be stored – data storage design
8. Create the final deliverable – system specification: all the above deliverables combined and presented to approval committee
   1. Conveys exactly what system the development team will implement during the implementation phase
   * What are the three acquisition strategies? Under what circumstances would you choose each one?

|  |  |  |  |
| --- | --- | --- | --- |
|  | When to use custom development | When to use a packaged system | When to use outsourcing |
| Business need | The business need is unique | The business need is common | The business need is not core to the business |
| In-house experience | In-house functional and technical experience exists | In-house functional experience exists | In-house functional or technical experience does not exist |
| Project skills | There is a desire to build in-house skills | The skills are not strategic | The decision to outsource is a strategic decision |
| Project management | The project has a highly skilled project manager and a proven methodology | The project has a project manager who can coordinate vendor’s efforts | The project has a highly skilled manager at the level of the organization that matches the scope of the outsourcing deal |
| Time frame | The time frame is flexible | The time frame is short | The time frame is short or flexible |

For custom development: What tools and technologies are needed for a custom development project?

- Pros: Get exactly what we want, new system built consistently with existing technology and standards, build and retain technical skills and function knowledge in-house, allows team flexibility and creativity

- Cons: Requires a significant time and effort, may add to existing backlogs, may require skills we do not have, often costs more, often takes more calendar time, risk of project failure

For pre-packaged software: What vendors make products that address the project needs?

- Pros: No need to “reinvent the wheel” for common business needs, tested and proven product, cost savings, time savings, utilize vendors’ expertise, some customization may be possible

- Cons: Rarely a perfect fit, organizational processes must adapt to software, reliance on vendor for maintenance and future enhancements, won’t develop in-house functional and technical skills, unique needs may go unmet, may require system integration

For outsourcing: What service provide would be able to build this application if outsourced

- Pros: Hire expertise we don’t have, may save time and money, lower risk, some consultants specialize in system integration

- Cons: No opportunity to build in-house expertise, reliance on vendor, future options limited, security (potential loss of confidential information), performance based on contract terms

* + What is an RFP? RFI? RFQ? What are their purposes?

Request for Proposals (RFP)

- Solicits proposals from vendor, developer, or service provider

- Explains the system to be built and criteria for selecting among applications

- Contents:

- Description of desired system

- Special technical needs or circumstances

- Evaluation criteria

- Instructions on how to respond

- Desired schedule

- Other information that will help the submitter to make a more complete or accurate proposal

Request for Information (RFI): A shorter and less detailed version

Request for Quote (RFQ): Use when you just need a price

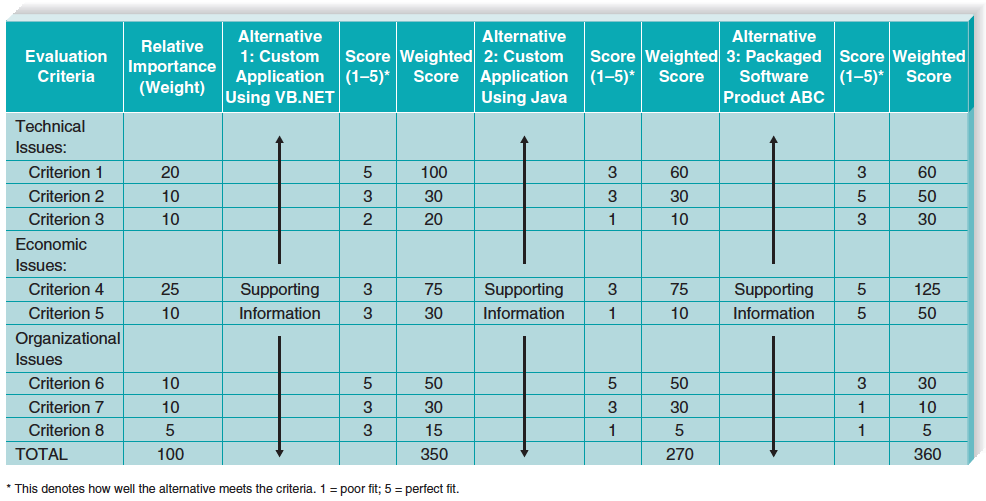
* + - Alternative matrix: what is listed across the top row and what is listed down the first column? How is the right alternative selected in a matrix? (hint: the one with the highest weighted score)

- Combine several feasibility analyses into one matrix

- Include technical, economic, and organizational feasibilities

- Assign weights to indicate the relative importance of the criteria

- Assign scores to indicate how well the alternative meets the criteria



* 6 questions from Chapter 8 (Architecture Design). You will need to know:
  + Definition of architecture design

Plans for how the system will be distributed across computers and what hardware/software will be used for each computer

* + Two major architectural components (hint: see next bullet point)

Software and hardware

* + What are the four basic functions of software? What are the three primary hardware components?

Software:

- Data storage

- Data access logic: The processing required to access stored data

- Application logic: The logic undocumented in the DFD’s use cases, and functional requirements

- Presentation logic: The display of information to the user and the acceptance of the user’s commands

Hardware:

- Client computers: Input-Output devices employed by users (PC’s, laptops, handheld and mobile devices)

- Servers: Larger multi-user computers used to store software and data

- Network: Connects the computers

* + In a client-server architecture, which of the four software types usually resides on the client, which usually resides on a server, and which varies depending on the architecture chosen?

- Clients are responsible for the presentation logic

- Server’s responsible for the presentation logic

- Application logic location varies depending on the client server configuration chosen

* + In a two-tiered client-server architecture, what is the difference between a thick and thin client?

Thick client: Most of application logic on the client side, handles presentation and application logic

Thin client: Little application logic on the client side; most shifted to the server side, handles data access logic and data storage

* + In mobile architecture (where the client is a mobile device), what is a rich client and what is a thin Web-based client? What technologies are used to create cross-platform apps (hint: Web-based)

Rich Client: Involves processing on the mobile device using its resources. Presentation logic, application logic, and data access logic on the client side

- Equivalent to a thick client in a desktop or laptop app

Thin Web-Based Client: Application and data access logic on the server side; always connected to server

Rich Internet Application: Browser-based; Uses some technologies on the client device to provide a rich user interface

Cross-Platform apps: HTML5 and JavaScript with a n overlaying interface so that it looks different from the browser

* + What is cloud computing, and can you name an example of cloud computing?

Cloud Computing: Computing resources obtained as a service

- Everything from computing power to computer infrastructure, applications, business processes to personal collaboration can be delivered as a service wherever and whenever needed

- The “Cloud” can be defined as the set of hardware, networks, storage, devices, and interfaces that combined to deliver aspects of computing as a service

- QuickBooks

- Dropbox

- Office 365

- Advantages:

- Elasticity: The resources allocate can be increased or decreased quickly, based on demand

- Cloud customers can obtain cloud resources in a straightforward fashion

- Cloud services typically have standardized API’s

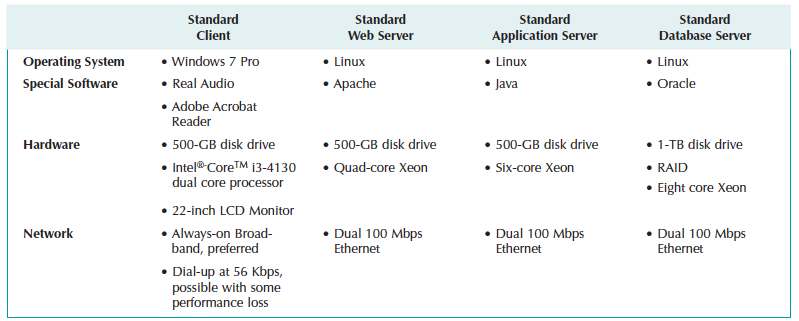
- Customers are billed for resources as they are used

* + HW-SW specification: purpose and format (what goes in the columns and what are the typical rows)

- Used if new hardware or software must be purchased

- Communicates project needs

- Actual acquisition of hardware and software may be done by purchasing department – especially in larger firms



* + You will not need to know: virtualization, sub-categories of nonfunctional requirements
* 7 questions from Chapter 9 (User Interface Design). You will need to know:
  + What are the three key mechanisms of the user interface?

User Interface: Portion of the system that directly interacts with users

- The navigation mechanism provides the way for users to tell the system what to do

- The input mechanism defines the way the system captures information

- The output mechanism defines the way the system provides information to users or other systems

* + What is the one key driving concept of a good user interface design?

Graphical User Interface (GUI): User interface that utilizes colors and graphics (as opposed to text only)

- Most common type of interface in use today

**Usability**

- What makes a system usable?

- The system is easy to use and easy to learn

- Why focus on usability as the key concept?

- Tasks are completed more efficiently and with more accuracy

- Mistakes with system are reduced

- User satisfaction with new system is increased

- Adoption of system is more likely

* + Definitions of the 6 principles of user interface design

Layout of the screen, form, report: Layout refers to organizing areas of the screen or document for different purposes and using those areas consistently throughout the user interface

Content Awareness: Refers to the ability of an interface to make the user aware of the information it contains with the least amount of effort by the user

Aesthetics: Aesthetics refers to designing interfaces that are pleasing to the eye

Usage Level: Refers to designing interfaces to accommodate both users who use the system heavily and routinely, and users who use the system only occasionally

Consistency: Consistency usually refers to the interface within one computer system, so that all parts of the same system work in the same way

Minimize user effort – Should be designed to minimize the amount of effort needed to accomplish task. Using the fewest possible mouse clicks or keystrokes to move from one part of the system to another

* + Three layout areas (see “Layout” slide from lecture)

Navigation area on top

Status area in the bottom

Work area in middle

* + For which type of user should you focus on ease of use? For which type of user should you focus on ease of learning (hint: see “Usage level” slide from lecture)
    - What is the 3 clicks rule and which of the 6 user interface design principles does it correspond to?
    - What are the 5 steps in UI design process?
    - What is an interface structure diagram (ISD)?
    - What is an interface metaphor? Be able to recognize one
    - What are the types of prototypes?
    - What is source data automation? Know examples
    - You don’t need to know: touch screen design, personas, interface evaluation methods, menu tips, message tips and types of messages, input validation types, lookups/dropdowns/default values, output design principles
* 4 questions from Chapter 10 (Program Design). You will need to know:
  + Definitions and purposes of structure charts, program specifications, and pseudocode
  + Difference between logical and physical process models
  + Steps to create a physical DFD
  + You don’t need to know: in-depth details of structure charts and program specifications; pages from Ch 10 that were not assigned reading; Ch 11 (physical *data* models)
* 6 questions from Chapter 12 (Moving into Implementation). You will need to know:
  + Definition of “Implementation” – what happens during the Implementation Stage?
  + Best practices for managing programmers
  + What are the three work areas for programming?
  + What are the 4 categories of testing? Define each
  + Black box vs white box testing
  + In which of the 4 categories do you test non-functional requirements?
  + Alpha vs beta testing
  + What are the two types of documentation?
  + What are the five types of documentation navigation (hint: one of them is table of contents)
  + You don’t need to know: slide about “people involved in implementation”, change control and program logs, classic mistakes, five common functions to test in unit testing, types of integration testing, specific types of “system” testing, three types of user documentation, three ways to organize documentation, writing guidelines for crafting documentation topics
* 6 questions from Chapter 13 (Transition to New System). You will need to know:
  + Conversion strategies:
    - * Direct vs. parallel
      * Pilot vs. phased vs. simultaneous
      * Whole system vs. module-by-module
      * 3 key factors that determine which combination of conversion strategy you select
      * Which strategy is best for high risk / low risk / high cost / low cost / limited time / plenty of time
  + What is a business contingency plan? o What are the 4 steps of change management?
  + Categories of potential adopters
  + What is the correct focus of training on a new system? (hint: it is NOT trying to explain every system feature)
  + What are the 3 important POST-implementation activities?
  + What are Level 1 and Level 2 support?
  + You don’t need to know: “elements of a migration plan”, management tools for supporting adoption, sources of maintenance change requests, details about project assessment  4 questions from
* Chapter 14 (Object-Oriented Approach). You will need to know:
  + Definitions of “object-oriented approach”, class, object, instance, UML
  + What is the methodology created specifically for the OO approach?
  + What are the two major groupings of UML diagrams?
  + What are the four fundamental UML diagrams, and which of the four serves as the basis or starting point for the rest?
  + You will not need to know: complex use case diagrams (e.g., “includes” and “extends”), how to read any UML diagrams beyond use case diagrams

**Free Response portion (4 questions):** For each of these questions, I will show one example document or diagram. You will be required to (a) name what type of document or diagram it is, and (b) state at least 3 things that you see wrong with the document (e.g., “button sizes in the prototype are not consistent”, etc.). Therefore, you must understand the key elements and guidelines for the following types of documents and diagrams that analysts and designers create (or are involved with):

* Acquisition Strategy Recommendation
* Hardware-Software Specification
* Interface Structure Diagram
* User Interface Prototype (use the 6 design guidelines to critique)
* Physical Data Flow Diagram
* Use Case Diagram