8

Architecture design:

Plans for how the system will be distributed across multiple computers and what hardware, operating system software and application software will be used for each computer.

Key factors: Nonfunctional requirements developed early in the analysis phase.

Object: determine how the software components of the information system will be assigned to the hardware devices of the system.

Architectural components: software and hardware

Software: Data Storage: type and amount of data stored.

Data access logic: the processing required to access stored data

Application logic: logic documented in DFDs, use case and functional requirements.

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

Hardware: Client computers(pcs, laptops)

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

Network: connects the computers.

Client-Server Architectures: balance the processing between client devices and one or more server devices.

Client: presentation logic. Server: data access logic and data storage.

Limitation: complexity

Two-tiered: client: presentation logic, application and data access logic. Sever: data storage

Three-tiered: Client: presentation logic. Application Server: business logic. Database server: database.

N tiered.

Server-based: terminal and server computer

Virtualization:

Server virtualization: partioning a physical server into smaller virtual servers.

Storage virtualization: combing multiple network storage devices into a single unit.

Cloud computing:

Advantage:

resources allocated can be changed based on demand.

The current infrastructure restricts the choice of architecture.

Create an architecture design:

Begins with the non-functional requirements

Non-functional requirements into more detailed system requirements for the selection of architecture.

Non-functional is Used to develop the hardware and software specification

Operational:

Technical environment: business requirements decide special hardware and software

System integration: extent to which the system will operate with other systems.

Portability: extent to which system will need to operate in other environment.

Maintainability: should adopt new business changes.(update system).

Performance:

Speed

Capacity

Availability and reliability:extent to which system will be available to users.

Security:

Culture and political:

Hard and software specification:

First define software

Then create a list of the hardware needed

Describe minimum requirements for hardware

9

Interface design:defining how the system will interact with external entities.

User interfaces: how the system will interact with the users.

User interfaces: navigation mechanism: the way in which the user tells the system what to do.

Input mechanism: system captures information

Output mechanism: system provides information to the user or to other systems.

Graphical user interfaces: use windows, menus, icons,etc are the most common type of user interfaces.

Principles for user interface design:

Layout: organizing areas of the screen or report.

3 parts: top for navigation through the system.

Middle: and the largest area for display of user’s work

Bottom: status information

Should minimize user’s movement from one area to the next.

Content awareness: ability of an interface to make the user aware of the information it contains. Should show where the user are and where the user came from.

Applies to the fields and field labels within each area.

Aesthetics: designing interfaces that pleasing to the eyes. Design of text: Fonts and font sizes. Colors and patterns.

User Experience: designing the user interface with the users level of computer experience in mind. ease of learning and ease of use.

Consistency: All parts of the same system work in the same way. Consistent with other computer systems in the organization. Navigation controls, terminology report and form design

Minimize user effort: using the fewest possible mouse clicks to move.

No more than 3 clicks from start to information or action.

Use Scenario development:

Outline of steps that users perform to accomplish some part of work

One path through the use case.

Simple narrative description that is tied to the DFD

Interface Structure design

Defines the basic components of the interface and how they work together to provide functionality to users.

Interface structure diagram(ISD)

Interface metaphor: concept from real world used in computer (checkbook metaphor)

Interface template: general appearance of all screens and the paper-based forms and reports.

Interface objects: fundamental building blocks of the system(entities and data storage)

Interface actions: common commands.

Interface Design prototyping:

Simulation of a computer screen, form or report.(story board, HTML prototypes, language prototypes)

HTML prototype: built with the use of web pages created in HTML

Input design: facilitate the entry of data into the computer system. Goal: capture accurate information for the system simply and easily.

Online and batch processing:

Online processing: each input item is entered into the system immediately.

Batch processing: inputs collected are gathered together and entered into the system at one time.

Output design:

Goal: present information to users so that they can accurately understand it with least effort.

Understand report usage: understand how they are used.

Manage information load: goal is to provide all needed information without information overload.

Minimize bias: No

Value of technical design

Looking forward: what we gonna do, coordinate development activities, consistent programming patterns

Looking backward: Document what we did, assist future developers.

Prototype(light bulb) vs Design Docs(garage)

Uncertainty, cost of prototype, complexity, coordination

When prototype:

Drive out uncertainty, front load risk( most difficult, uncertain), know when to stop

When design:

Focus on value

DRY

Create, report, update, delete

Focus on difference

Program design: determine what programs will be written

Program specification: detailed instructions for programmer

Physical data flow diagram: DFD shows implementation decisions

Difference between physical DFD and logical DFD is that a physical DFD contains additional details that describe how the system will be built.

Step:

1. Add implementation references
2. Draw a human-machine boundary
3. Add system-related data stores, data flows, and processes
4. Update the data elements in the data flows
5. Update the metadata in the CASE repository

CASE: computer-aided software engineering

Designing programs:

4 components for program specification:

Program information events inputs and outputs pseudocode.

Data storage function: how data is stored and handled by programs.

Design:

1. Select the data storage format
2. Convert the logical data model created during analysis into physical data model.
3. Design the data storage format

2types of data storage formats:

Files: lists of data that have been optimized to perform a particular transaction.

Database: a collection of groupings of information that are related to each other.

Files types:

Master files: information important to the application.

Transaction Files: used to update a master file.

Audit files: record “before” and “after” images of data

History Files: store past historic data.

Database:

Legacy database: currently in production

Relational Database: collection of tables, which has primary key. SQL used to access the data in tables.

Object Database: All things should be treated as objects, change to one have no effect on other. Used to support multimedia application

Multi-dimensional Database: a type of relational database that is used in data warehousing. Support data aggregations on multiple dimensions

Select a storage format:

1. data types
2. Type of application system
3. Existing storage formats
4. Future needs

Physical Entity Relationship Diagram(ERD)

Contains references to how data will be stored and much more metadata.

Transition from logical to physical data model

1. change entities to tables or files
2. Change attributes to fields
3. Add keys
4. Add system related components.

Optimizing data storage:

1. storage efficiency: have no redundant data. Normalization
2. Access speed: Denormalization, indexing, estimating

t123 4 5 6 7890