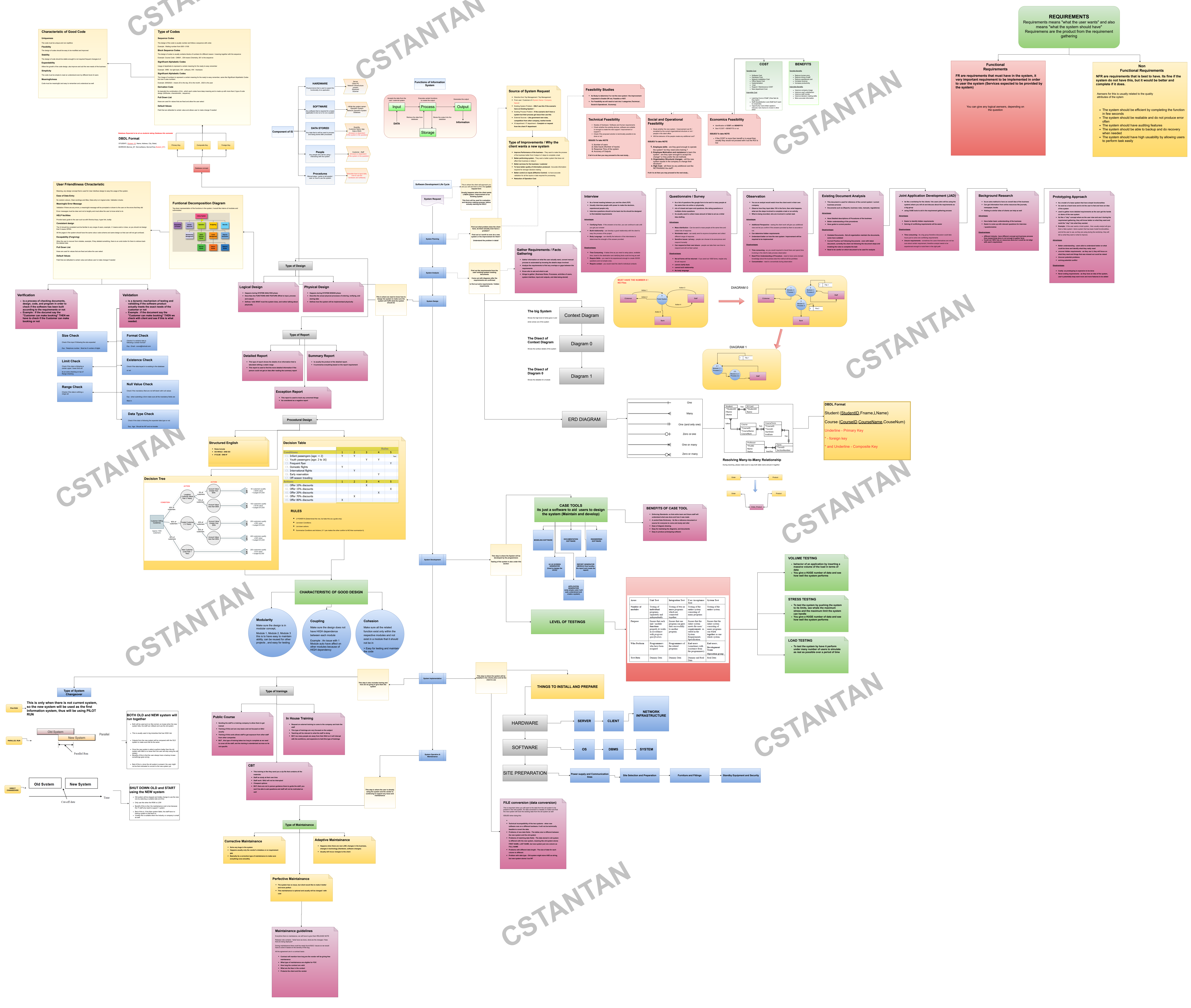
SAD NOTES

CSTANTAN



Characteristic of Good Code

Uniqueness

The code must be unique and non repititive

Flexibility

The design of codes should be easy to be modified and improved

Stability

The design of code should be stable enought to not required frequent changes to it

Expandability

Alllow the growth of the code design, aka improve and suit the new needs of the business

Simplicity

The code must be simple to read an understood even by different level of users

Meaningfulness

Code must be meaningful and easy to remember and understood as well

Type of Codes

Sequence Codes

The design of the code is usually number and follow a sequence with order.

Example : Waiting number from 0001- 0100

Block Sequence Codes

The design of codes is usually contains blocks of numbers for different reason / meaning together with the sequence

Example: Course Code : CM001 , CM means Chemistry, 001 is the sequence

Significant Alphabetic Codes

Usage of alpahbets to represent a certain meaning for the ready to easy remember

Example : BRB - be right back, SW - software, HW - Hardware

Significant Alphabetic Codes

The Usage of numbers to represent a certain meaning for the ready to easy remember, same like Significant Alpahbetic Codes but now it uses numbers

Example :20052023 - means 20 is the day, 05 is the month , 2023 is the year

Derivation Code

Its basically the combination of ALL, which each codes have deep meaning and is made up with more than 2 type of code design (Attributes, characterisitic , sequence)

Pull Down List

these are used for values that are fixed and allow the user select

Default Values

Field that are defaulted to certain value and allows user to make change if needed

Database Expected to be ok as students taking Database this semester

DBDL Format

STUDENT (Student_ID, Name, Address, City, State)

BORROW (Borrow_ID*, BorrowName, BorrowTime,Student_ID*)

Component of IS

HARDWARE

Physical device that is used to support the functionality of an application

SOFTWARE

Any software that is required for a basic application to be run or to run a system

DATA STORED

Any data that is will be used to process and being stored after processed

People

Any people who will be using / interacting with the system

Procedures

Manual steps / guide to tell people / user on how to use the system

[Describe how to input data, how to use the hardware and software]

Functions of Information System

Accepts the data from the user / external system
Executes certain functions to create the output
Input
Process
Output
DATA
Retrieve the data from database
Stores the output into the database
Storage

User Friendliness Chracteristic

Meaning, any design concept that is used for User interface design to easy the usage of the system.

Ease of Data Entry

No random colours, Clear wordings and titles, Data entry is in logical order, Validation checks

Meaningful Error Message

Validation if there are any errors, a meaningful message will be prompted or shown to the user on the errors that they did

Eroor messages must be clear and not to lengthy and must allow the user to know what to do

HELP facilities

Provide basic guide to the user such as with Shortcut keys, hyper link, tooltip

Consistent design

The UI should be consistent and be familiar to any range of users, example, 'x' means exist or close, so you should not design that X means OPEN.

All the pages in the system should have the same colour code scheme and same design so that user will not get confused.

Escapability (Forgiving)

Allow the user to recover from mistake, example, if they deleted something, there is an undo button for them to retrieve back the deleted data

Pull Down List

these are used for values that are fixed and allow the user select

Default Values

Field that are defaulted to certain value and allows user to make change if needed

Funtional Decomposition Diagram

Top-down representation of the functions in the system, it would like interns of modules and submodules

Verification

- is a process of checking documents, design, code, and program in order to check if the software has been built according to the requirements or not
- Example : If the documet say the "Customer can make booking" THEN we have to check if the Customer can make booking or not

Validation

- is a dynamic mechanism of testing and validating if the software product actually meets the exact needs of the customer or not
- Example : if the document say the "Customer can make booking" THEN we check with client and see if this is what needed.

Size Check

Check if the input if following the size expected

Exp : Telephone number : Must be X number of digits

Format Check

Checks if a certaind data is following a certain format?

Exp : Email - xxxxx@hotmail.com

Limit Check

Existence Check

Check if the data keyed in is existing in the database or not

Range Check

Null Value Check

Check if the mandatory filed are not left blank/ with null values

Exp : when submiting a form make sure all the mandatory fields are filled in

Data Type Check

Check if the data is following the expected data type or not

Exp : Age : Should be INT and not double

Type of Design

Logical Design

- Happens during SYSTEM ANALYSIS phase
- Describes the FUNCTIONS AND FEATURE (What to input, process and output)
- Defines / tells WHAT must the system does, and nother talking about physically

Physical Design

- Happens during SYSTEM DESIGN phase
- Describe the actual physical processes of entering, verifying, and storing date
- Defines how the system will be implementaed physically

This is where we need to know how to design the system to make sure the system will fulfill what the system should do

Type of Report

Detailed Report

- This type of report shows the details of an information that is tabulated withing a cetain range
- This report is used to find the more detailed informaiton if the person could not get an idea after reading the summary report

Summary Report

- Is usually the product of the detailed report.
- It summarise everything based on the report requirement

Exception Report

- This report is used to track any unnormal things
- Its considered as a negative report

Procedural Design

Structured English

- Rules Include
- DO WHILE - END DO
- IF ELSE - END IF

Decision Table

Conditions	1	2	3	4	5
C1. Infant passengers (age: < 2)	Y	Y			Text
C2. Youth passengers (age: 2 to 16)			Y	Y	
C3. Frequent flyer					Y
C4. Domestic flights	Y				
C5. International flights		Y			
C6. Early reservation				Y	
C7. Off-season traveling					
Actions	**1**	**2**	**3**	**4**	**5**
A1. Offer 10% discounts			X		
A2. Offer 15% discounts					X
A3. Offer 20% discounts				X	
A4. Offer 70% discounts		X			
A5. Offer 80% discounts	X				

RULES

- 2 POWER N (Determined the row, but take this as a guide only)
- List down Conditions
- List down actions
- Summarize Conditons and Actions (if 1 yes makes the other confirm to NO then summarize it).

Decision Tree

ACTION
ACTION
CONDITION
Customer Gifting Guidelines
Approx 1000 customers
20% of customers
60% of customers
20% of customers
Longtime Customer (More than 3 Years)
Trusted Customer (1-3 Years)
New Customer (Less than 1 Year)
5% of customers
15% of customers
20% of customers
40% of customers
Account Value Greater than $50k
Account Value less than $50k
Account Value Greater than $50k
Account Value less than $50k
Gift 1
Gift 2
Gift 3
Gift 4
Gift 5
50 customers qualify x $300 value = budget $15,000
150 customers qualify x $100 value = budget $15,000
200 customers qualify x $50 value = budget $10,000
400 customers qualify x $25 value = budget $10,000
200 customers qualify x $10 value = budget $2,000

CHARACTERISTIC OF GOOD DESIGN

Modularity

Make sure the design is in modular concept,

Module 1, Module 2, Module 3 this is to have easy to maintain ability, can be reused for other projects , and easy for testing

Coupling

Make sure the design does not have HIGH dependence between each module

Example : An issue with 1 Module auto have affect on other modules because of HIGH dependency

Cohesion

Make sure all the related function exist only within the respective modules and not exist in a module that it should not be in

> Easy for testing and maintain the code

Software Development Life Cycle

- System Request
- System Planning
- System Analysis
- System Design
- System Development
- System Implementation
- System Operation & Maintainance

This is where the client will approach you and you will ask them to fill in the system request form

Usually happens when the client wants a NEW system / Improvement of an Existing system

This form will be used for evaluation and decision making purpose, before actually starting the SDLC

Find out what problem does the client have, do there actually even have a problem?

Find out is it worth to even do a new system or the improvement for them?

Understand the problem in detail

Find out the requirements from the user/ existing system/ existing documents

Come out with diagrams after the requirements are confirmed

to find out extra requirements / hidden requirments

Source of System Request

- Directive from Top Management / Top Management
- From user / Customer of <System Name / Company Name>
- Existing System Problem - ONLY use this if the scenario have an Existing System !
- Existing Process Problem - If the scenario don't have a system but their process got issue then use this
- External Sources - Like goverment new rules, competition from other company, market trends
- IS department / IT department - Complain or request from the client IT department

Type of Improvements / Why the client wants a new system

- Improve Performance of the business - They want to make the process of the business better from 5 steps to 3 steps to complete a task
- Better performing system - They want a better system that does not affect their business or delay it
- Better services for the business / customer
- To have better quality of information produced - Accurate information required for stronger decision making
- Better control on inputs (Effective Control) - to have accurate validation for all the inputs or data required for processing
- Reduction of Operation Cost

Feasibility Studies

- Its Study to determine if or not the new system / the improvement requested is Doable OR not, Feasible or NOT.
- For Feasibility we will need to look into 3 categories (Technical , Social & Operational , Economy)

Technical Feasibility

- Studies of Hardware / Software and Human requirements
- Check whether the existing devices, software are suitable or enough to create the new support / improvement or support it
- Check if the proposed solution is technically possible to be done or not

ISSUES To take NOTE

1. Number of users
2. Data Inputs (Number of inputs)
3. Response Time of the system
4. Accuracy of Outputs

If all 4 is ok then you may proceed to the next study...

Social and Operational Feasibility

- Study whether the new system / improvement can fit / accepted by the current organizational structure or not / people in the client office
- Will the behavior of the people create any additional cost?

ISSUES To take NOTE

1. Employee skills - are they good enough to operate the system? do they need extra training ?
2. Employee Motivation and resistance to learn the system - are they open enought to accept the change? or they prefer the old methods
3. Organization Structural changes - will the new system results in the company to change their structure?
4. High Cost- will there be any additional cost like RETRAINING the staff?

If all 4 is ok then you may proceed to the next study...

Economics Feasibility

- Identification of COST and BENEFITS
- See if COST > BENEFITS or not

ISSUES To take NOTE

- If the COST is more than benefit or is equal then usually they would not proceed with it as the ROI is low

COST

Tangible Cost

- Software Cost
- Hardware Cost,
- Network installation Cost
- Office Space Cost
- Implementation Cost
- Support / Maintenance COST
- New department Cost

Intangible Cost

- Learning Curve of Staff (How fast do they learn)
- Staff dissatisfaction cost (Staff don't want to cooperate)
- Opportunity Cost (Initial capital is reduced, lost chance to invest in other area)

BENEFITS

Tangible Benefits

- Reduce human error
- Reduce hiring of staff
- Reduce operational cost
- Increase revenue
- Increase productivity

Intangible Benefits

- Improve company image
- Improve user's satisfaction
- Improve staff morale
- Improve decision making skills
- More accurate information

REQUIREMENTS

Requirements means "what the user wants" and also means "what the system should have"
Requiremens are the product from the requirement gathering

Functional Requirements

FR are requirements that must have in the system, it very important requirement to be implemented in order to user the system (Services expected to be provided by the system)

You can give any logical awnsers, depending on the question

Non Functional Requirements

NFR are requirements that is best to have. Its fine if the system do not have this, but it would be better and complete if it does.

Awnsers for this is usually related to the quality attributes of the sytem

- The system should be efficient by completing the function in few seconds
- The system should be realiable and do not produce error often
- The system should have auditing features
- The system should be able to backup and do recovery when needed
- The system should have high usuability by allowing users to perform task easily

Gather Requirements / Facts Gathering

- Gather information on what the user actually want, convert manual process to automated by knowing the details steps involved
- Analyze the requirements to find any overlaps or gaps between the requirements
- Know who to ask and what to ask
- things to gather (Business Rules, Processes, activities of users, system interface, input and outputs, and data being stored)

Interview

- Its a formal meeting between you and the client (F2F)
- Usually interview people with power to make the decision, experienced people only
- Interview questions should not be basic but its should be designed to find detailed requirements

Advantages

- Clarifying Facts - if the answers is too blur, you can continue to ask until you get your answer
- Build relationship - can develop a good relationship with the client to easy the process of the project
- Body Language - can identify the behavior of the interviewee to determined the strength of the answer provided.

Disadvantages

- Time Consuming - It takes time as you need to setup place, available time, travel to the destination and clarifying facts could be long as well
- Require Skills - you need to be experienced enough to create GOOD questions and not simple ones.
- Require contact - you would need the client's individual contacts

Questionnaire / Survey

- Its a list of questions like google form to be sent to many people at the same time via online or physically.
- mix of closed and open end questions, like rating questions or multiple choice questions
- Its usually used to collect mass amount of data to act as a initial idea drafting

Advantages

- Mass distribution - Can be send to mass people at the same time and collect lots of response
- Worldwide reach - can easily send to anyone at anywhere and collect different range of response
- Sensitive issues / privacy - people can choose to be anonymous and respond honestly
- Can respond at their own leisure - people can take their own time to respond and will not feel rushed.

Disadvantages

- Not all forms will be returned - if you send out 1000 forms, maybe only 50 will respond
- cannot clarify facts
- cannot built relationship
- No body language

Observation

- You as an analyst would watch how the client work in their own company
- Observe how they input data / fill in the form, then what happens, what are the steps involved to complete a task or an activity
- What is being recorded, who are involved in certain task

Advantages

- Additional Perspective - seeing the client work will give you additional view and let you confirm if the answers provided by them is accurate or not.
- Can determine hidden requirements
- Understand better on how the new system / improvement will be required to be implemented

Disadvantages

- Time consuming - as you would required to travel there and spend time involving in the client's daily process observation
- Need Prior Understanding of Procedure - need to have some domain knowledge about the business else this method will be pointless
- Concentration - need to concentrate during observation

Existing Document Analysis

- This document is used for reference of the current system / current business process
- Documents such as (Reports, business rules, manuals, regulations)

Advantages

- Have Detailed descriptions of Procedures of the business
- Better understanding of the procedures
- Have guide to current practice

Disadvantages

- Outdated Documents - Not all organization maintain the documents, some have outdated
- Current Practice not Following Documents - even with latest document, currently the client not following the document steps and breaking the rules to complete the task
- Need to be careful on which document to be used for analysis

Joint Application Development (JAD)

- Its like a workshop for the clients / the users (who will be using the system) where you will sit and discuss about the requirements as a big group
- using CASE tools to aid in the requirement gathering process

Advantages

- Easier to identify hidden requirements
- Solving of conflicting requirements will be easier

Disadvantages

- Time consuming- its a big group therefore discussion could take hours just to solve one conflicting requirements
- Unsure requirements - sometimes the users themselves are not too sure about certain requirement, therefore analyst needs to be experienced enough to route them in the right path

Background Research

- Its an extra method to have an overall idea of the business
- Can get information from online resources like journals, newspaper, books
- Visiting to similar sites of clients can help as well

Advantages

- Have better basic understanding of the business
- Easier to come up with relevant questions for interview / questionnaires

Disadvantages

- different industry have different concept and business process thus researching alone is not enough to gather facts
- Could not 100% rely on resources alone as it could be not align with user's requirement

Prototyping Approach

- Its a model of a base system that have simple functionalities
- Its used as a back bone and to let the user to feel and have an idea of the system
- used to gather more detailed requirements as the user get the hands on demo of the new system
- Its like a "clay" concept where as the user view and use it during the requirement gathering, they will know better on what they want and mold the "clay" into what they wanted.
- Example : if the user wants a hotel system , then usually analyst will give them a fake system / demo system that has basic hostel functionalities, and let the user to use, as they are using during the workshop, they will tell us what they want more or what to improve.

Advantages

- Better understanding : users able to understand better on what could be done and identify what they really need
- uncover hidden requirements : as they use it, they will focus on what they need and things that was missed out could be raised
- Uncover potential problems
- solving potential conflict

Disadvantages

- Costly: as prototyping is expensive to be done
- Never ending requirements : as they have an idea of the system, user's potentially keep want more and more features to be added

Context Diagram

The big System

Shows the high level of what goes in and what comes out of the system

MUST HAVE THE NUMBER 0 ! NO Files

Diagram 0

The Disect of Context Diagram

Shows the surface details of the system

DIAGRAM 0

Diagram 1

The Disect of Diagram 0

Shows the detailed of a module

DIAGRAM 1

ERD DIAGRAM

Resolving Many-to-Many Relationship

During resolving, please make sure to copy both table name and join it together

DBDL Format

Student (StudentID,Fname,LName)

Course (CourseID,CourseName,CouseNum)

Underline - Primary Key

* - foreign key

* and Underline - Composite Key

CASE TOOLS

its just a software to aid users to design the system (Maintain and develop)

This step is where the System will be developed by the programmers

Testing of the system is also under this section

MODELING SOFTWARE
DOCUMENTATION SOFTWARE
ENGINEERING SOFTWARE
UI/ UX SCREEN GENERATOR (Used to design the UI/UX)
REPORT GENERATOR (MODULE that handles the report and create the report)
APPLICATON GENERATIOR (only simple code it will auto understand and create a system)

BENEFITS OF CASE TOOL

- Enforcing Standards- so that entire team and future staff will understand what was done and how it was made
- A central Data Dictionary - Its like a reference document or source for everyone to come and study and refer
- Ease of diagram drawing
- Easy for maintaing the diagrams, and documents
- Easy to produce prototyping software

LEVEL OF TESTINGS

Areas	Unit Test	Integration Test	User Acceptance Test	System Test
Number of modules	Testing of **individual** programs, separately and independently.	Testing of **two or more** programs which are connected together.	Testing of the **entire** system consisting of many programs	Testing of the **entire** system.
Purpose	Ensure that each unit / module **functions** properly & works in accordance with *program specification.*	Ensure that one program can **pass** data successfully to another program.	Ensure that the entire system meets the users' **requirements** as stated in the System Requirements Specification.	Ensure that the entire system consisting of many programs can **work together** as one whole system.
Who Perform	**Programmers** who have been assigned	**Programmers** of the related programs	**End-users** (sometimes with assistance from the programmers).	**End-users**, **Development Team** **Operation group**
Test Data	Dummy Data	Dummy Data	Dummy and Real Data	Real Data

VOLUME TESTING

- behavior of an application by inserting a massive volume of the load in terms of data
- You give a HUGE number of data and see how well the system performs

STRESS TESTING

- To test the system by pushing the system to its limits, see whats the maximum stress and the maximum limit the system can handle
- You give a HUGE number of data and see how well the system performs

LOAD TESTING

- To test the system by have it perform under many number of users to simulate as real as possible over a period of time

THINGS TO INSTALL AND PREPARE

This step is where the system will be installed in the clients place and let the client to use

FILE conversion (data conversion)

This is important when you still want to the data from the old system to be present in the new system. So data conversion is needed, to make sure that the new system will have the existing data from the old system as well

ISSUES when doing this :

- Technical incompatibility of the two systems - when new software runs on a different hardware. It will not be technically feasible to covert the data.
- Problems of new data fields - The tables size is different between the new system and the old system
- Problems of matching data fields - The data stored in old system is different with the new system, meaning like old system stores FIRST NAME, LAST NAME, but new system just one column as FULL NAME
- Problems with different data lenght - The size of data for each column is different
- Problem with data type - Old system might store AGE as string, but new system stores it as INT

This step is also includes traning and how are we going to give them the system

Type of System Changeover

Pilot RUN

This is only when there is not current system, so the new system will be used as the first Information system, thus will be using PILOT RUN

PARALLEL RUN

BOTH OLD and NEW system will run together

- Both will be used and run like normal, so incase when the new system fails, the staff can rollback and use the old system
- This is usually used in big industries that has HIGH risk
- Outputs from the new system will be compared with the OLD system to make sure that its the same
- Once the new system is able to perform better than the old system with little to no issue then the user will stop using the old system
- Benefits of this is that the user always have a backup incase somethings goes wrong
- Bad of this is, since the old system is present, the user might not be that motivated to convert to the new system yet

DIRECT CHANGEOVER

SHUT DOWN OLD and START using the NEW system

- Old system will be stopped and totally change to use the new one by selecting a suitable date and time
- Only use this when the RISK is LOW
- Benefit of this is that, the maintainance cost is low because the IT staff only need to support 1 system
- Bad of this is, if the New system failed, the staff have no backup system to fall back on
- Usually this is suitable when the Industry or company is small as well

Type of tranings

Public Course

- Sending the staff to a training company to allow them to get trained
- Training of this sort are very basic and not focused on BAU usually
- Training of this sorts allows staff to get exposure from other staff from other companies
- BUT , this type of training takes too long to complete as we need to cover all the staff, and the training is standarized accross so its not specific

In House Training

- Request an external training to come to the company and train the staff
- This type of trainings are very focused on the subject
- Teaching will be relevant to what the staff is doing
- BUT, too many people are away from their BAU so it will interupt with the workforce, and expensive to held this type of trainings

CBT

- This training is like they send you a zip file that contains all the materials
- Staff an study at their own time
- Staff work / BAU will not be interupted
- Cheapest options
- BUT, there are not in person guidance there to guide the staff, you won't be able to ask questions and staff will not be motivated as well

This step is where the user is already using the system and the vendor is continuing to support any issue and maintainance

Type of Maintainance

Corrective Maintainance

- Solve any bugs in the system
- Happens usually only the vendor's mistakes or on requirement gap
- Basically its a correction type of maintainance to make sure everything runs smoothly

Adaptive Maintainance

- Happens when there are new LAW, changes in the business, changes in technology (Hardware, software changes)
- Usually will inccur charges to the client

Perfective Maintainance

- The system has no issue, but client would like to make it better and more perfect
- This maintainance is optional and usually will be charged / with cost

Maintainance guidelines

Everytime there is maintainance, we will have to give them RELEASE NOTE

Release note contains "what have we done, what are the changes / fixes that are being deployed"

During maintainance there could be newly found BUG / issues so we would have to solve it based on the severity of the bug

All the agreement are in a contract basis :

- Contract will mention how long are the vendor will be giving free maintainance
- What type of maintainance are eligible for FOC
- How long the contract are valid
- What are the fees in the contact
- Protects the client and the vendor