NAME : DANYAL AHMAD

ID # 13838

MODULE: SEMESTER-VI

Course: Information system and data processing

Program: BS (SE)

Instructor: MUHAMMAD ABRAR KHAN

Sessional Assignment

Q1 : Define organization; also explain the structure of an organization by giving an example of a well known organization. (Note: every student should take the example of different organization from another).

ANSWER 1():

ORGANIZATION:

Organization refers to a collection of people, who are involved in pursuing defined objectives. It can be understood as a social system which comprises all formal human relationships. The organization encompasses division of work among employees and alignment of tasks towards the ultimate goal of the company.

It can also be referred as the second most important managerial function, that coordinates the work of employees. procures resources and combines the two, in pursuance of company’s goals.

STRUCTURE OF AN ORGANIZATION(GOOGLE):

Google has a ***cross-functional organizational structure***, which is technically a matrix corporate structure with a considerable degree of flatness. This flatness is a defining structural feature that supports the company’s growth and competitiveness. The main characteristics of Google’s corporate structure are as follows:

1.Function-based Structure
2.Product-based Structure
3.Flatness Structure

1.Function-based Structure:

A functional structure has people performing in specialized roles. Furthermore, Google’s organizational structure draws on the strengths of the functional structure. Functional organization is said to boost the productivity of the employees.

2.Product-based Structure:

In a divisional organization, each division has their own organizational function. These divisions are usually in correspondence to geography or product.
This way if a company has a division that deals with food and the other that deals with household items. Each division will have their own Finance, IT and marketing department. It allows operational flexibility for the organization and helps in increasing productivity and overall efficiency and effectiveness of the strategies.

3. Flatness Structure:

Flat structure in an organization allows greater flexibility and allows employees to explore newer dimensions. The middle management is nearly eliminated and therefore the employees and the managers have a direct contact with each other.

This allows for better flexibility and comfort for the employees and allows them to adapt to change.

Google has now reformed its organizational structure, by adopting the formation of Alphabet. All the subsidiaries and the products and services come under Alphabet. This has reduced the dependency of the different divisions on each other.

Thus, the lack of revenue production of one division is not going to affect the organization as the whole. As a result of which, there has been an increase in the transparency for the investors, who can now see if their investments are paying off as they were expecting it to.

The holdings of Alphabet has Google core products and other subsidiaries.

2. Explain System Development Life Cycle; also explain different types system development life cycle.

Answer(2):

SYSTEM DEVELOPMENT LIFE CYCLE:

A system is a set of interacting or interdependent components forming an integrated whole” it’s a term that can be used in different industries, therefore Software Development Life Cycle is a limited term that explains the phases of creating a software component that integrates with other software components to create the whole system.

The systems development life cycle (SDLC) is a conceptual model used in project management that describes the stages involved in an information system development project, from an initial feasibility study through maintenance of the completed application. SDLC can apply to technical and non-technical systems. In most use cases, a system is an IT technology such as hardware and software. Project and program managers typically take part in SDLC, along with system and software engineers, development teams and end-users.

Every hardware or software system will go through a development process which can be thought as an iterative process with multiple steps. SDLC is used to give a rigid structure and framework to define the phases and steps involved in the development of a system.

The most used, popular and important SDLC models are given below:

1.Waterfall model
2.Iterative model
3.Spiral model
4.V-shaped model
5.Agile model

1. WATERFALL MODEL:

Waterfall model is the very first model that is used in SDLC. It is also known as the linear sequential model.
In this model, the outcome of one phase is the input for the next phase. Development of the next phase starts only when the previous phase is complete.

2. ITERATIVE MODEL:

The Iterative model is repetition incarnate. Instead of starting with fully known requirements, you implement a set of software requirements, then test, evaluate and pinpoint further requirements. A new version of the software is produced with each phase, or iteration. Rinse and repeat until the complete system is ready.

One advantage over other SDLC methodologies: This model gives you a working version early in the process and makes it less expensive to implement changes. One disadvantage: Resources can quickly be eaten up by repeating the process again and again.

3. SPIRAL MODEL:

One of the most flexible SDLC methodologies, the Spiral model takes a cue from the Iterative model and its repetition; the project passes through four phases over and over in a “spiral” until completed, allowing for multiple rounds of refinement.

This model allows for the building of a highly customized product, and user feedback can be incorporated from early on in the project. But the risk you run is creating a never-ending spiral for a project that goes on and on.

4. V-shaped Model:

Also known as the Verification and Validation model, the V-shaped model grew out of Waterfall and is characterized by a corresponding testing phase for each development stage. Like Waterfall, each stage begins only after the previous one has ended.

This model is useful when there are no unknown requirements, as it’s still difficult to go back and make changes.

5. Agile Model:

Agile Model is a combination of the Iterative and incremental model. This model focuses more on flexibility while developing a product rather than on the requirement.

In Agile, a product is broken into small incremental builds. It is not developed as a complete product in one go. Each build increments in terms of features. The next build is built on previous functionality.

In agile iterations are termed as sprints. Each sprint lasts for 2-4 weeks. At the end of each sprint, the product owner verifies the product and after his approval, it is delivered to the customer.

3. Explain Incremental model and Spiral; also explain main deference between spiral and incremental model.

ANSWER(2):

Incremental Model:

Incremental Model is a process of software development where requirements divided into multiple standalone modules of the software development cycle. In this model, each module goes through the requirements, design, implementation and testing phases. Every subsequent release of the module adds function to the previous release. The process continues until the complete system achieved.

The various phases of incremental model are as follows:

1. Requirement analysis: In the first phase of the incremental model, the product analysis expertise identifies the requirements. And the system functional requirements are understood by the requirement analysis team. To develop the software under the incremental model, this phase performs a crucial role.

2. Design & Development: In this phase of the Incremental model of SDLC, the design of the system functionality and the development method are finished with success. When software develops new practicality, the incremental model uses style and development phase.

3. Testing: In the incremental model, the testing phase checks the performance of each existing function as well as additional functionality. In the testing phase, the various methods are used to test the behavior of each task.

4. Implementation: Implementation phase enables the coding phase of the development system. It involves the final coding that design in the designing and development phase and tests the functionality in the testing phase. After completion of this phase, the number of the product working is enhanced and upgraded up to the final system product.

Advantage of Incremental Model:

>Errors are easy to be recognized.

>Easier to test and debug

>More flexible.

>Simple to manage risk because it handled during its iteration.

>The Client gets important functionality early.

Disadvantage of Incremental Model:

>Need for good planning

>Total Cost is high.

>Well defined module interfaces are needed.

SPIRAL MODEL :

The spiral model, initially proposed by Boehm, is an evolutionary software process model that couples the iterative feature of prototyping with the controlled and systematic aspects of the linear sequential model. It implements the potential for rapid development of new versions of the software. Using the spiral model, the software is developed in a series of incremental releases. During the early iterations, the additional release may be a paper model or prototype. During later iterations, more and more complete versions of the engineered system are produced.

Each cycle in the spiral is divided into four parts:

Objective setting: Each cycle in the spiral starts with the identification of purpose for that cycle, the various alternatives that are possible for achieving the targets, and the constraints that exists.

Risk Assessment and reduction: The next phase in the cycle is to calculate these various alternatives based on the goals and constraints. The focus of evaluation in this stage is located on the risk perception for the project.

Development and validation: The next phase is to develop strategies that resolve uncertainties and risks. This process may include activities such as benchmarking, simulation, and prototyping.

Planning: Finally, the next step is planned. The project is reviewed, and a choice made whether to continue with a further period of the spiral. If it is determined to keep, plans are drawn up for the next step of the project.

Advantages:

High amount of risk analysis

Useful for large and mission-critical projects.

Disadvantages:

Can be a costly model to use.

Risk analysis needed highly particular expertise

Doesn't work well for smaller projects.

Deference between spiral and incremental model:

The Spiral model puts focus on thorough risk assessment. Thus, to reap the benefits of the model to the fullest, you’ll need to engage people with a strong background in risk evaluation. A typical Spiral iteration lasts around 6 months and starts with 4 important activities – thorough planning, risk analysis, prototypes creation, and evaluation of the previously delivered part. Repeated spiral cycles seriously extend project time frames.

The development process based on the Incremental model is split into several iterations . New software modules are added in each iteration with no or little change in earlier added modules. The development process can go either sequentially or in parallel. Parallel development adds to the speed of delivery, while many repeated cycles of sequential development can make the project long and costly. With Iterative development software changes on each iteration, evolves and grows. As each iteration builds on the previous one, software design remains consistent.

THE END