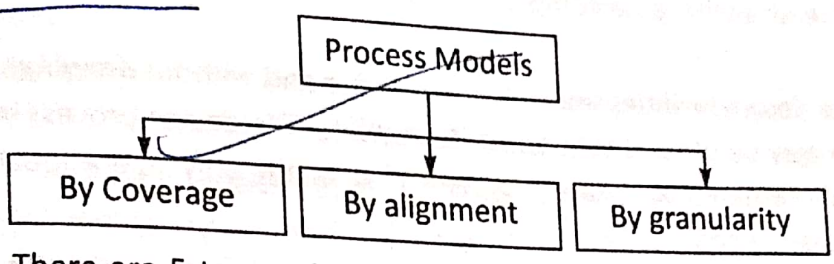
Need of Process Modeling

1. From a theoretical point of view, the meta-process modeling explains the key concepts needed to describe what happens in the development process, when it happens and why it happens.
2. From an operational point of view, the meta process modeling is aimed at providing guidance for method engineers and application developers.
3. The activity of modelnig a business process usually predicates a need to change processes or identify issues to be corrected. This transformation may or may not require IT involvement, although that is a common driver for the need to model a business process.
4. Process modeling addresses the process aspects of an Enterprise Business Architecture, leading to an all encompassing enterprise architecture.

Classification of Process Models



(a) By Coverage : There are 5 types of coverage where the term process model has been defined ifferently.

(i) **Activity Oriented :** Related to set of activities conducted for the specific purpose of product definition.

(ii) **Product Oriented :** It includes a series of activities that cause sensitive product transformations to reach the desired product.

(iii) **Decision Oriented :** It includes a set of related decisions conducted for the specific purpose of product definition.

(iv) **Context Oriented :** It includes sequence of contexts causing successive product transformations under the influence of a decision taken in a context.

(v) **Strategy Oriented :** It allows building models representing multi-approach processes and plan different possible ways to elaborate the product based on the notion of intentions and strategy.

(b) By Alignment : Process can be of different kinds. These definitions corresponds to the various ays in which a process can be modelled.

(i) Strategic Processes :

- Investigate alternative ways of doing a thing and eventually produce a plan for doing it.
- Are often creative and require human cooperation.

(ii) Tactical Processes :

- helps in the achievement of a plan.
- are more concerned with the tactics to be adopted for actual plan achievement than with the development of a plan of achievement.

(iii) Implementation Processes :

- are the lowest level processes
- are directly concerned with the details of 'what' and 'how' of plan implementation.

(c) By Granularity : Granularity refers to the level of detail of a process models and affects the kind of guidance, explanation and trace that can be provided. Coarse granularity restricts these to a rather limited level of detail where as fine granularity provides more detailed capability.

✔ User Interface Design

User interface design or user interface engineering is the design of websites, computers, appliances, machines, mobile communication devices and software applications with the focus on the user's experience and interaction. The goal of user interface design is to make the user's interaction as simple and efficient as possible, in terms of accomplishing user goals – which is often known as user centered design.

Good user interface design facilities finishing the task in hand without drawing unnecessary attention to itself. Graphic design may be utilized to support its usability. The design process must balance technical functionality and visual elements to create a system that is not only operational but also usable and adaptable to changing user needs.

User interface design is involved in wide range of projects from computer systems, to cars, to commercial planes. All of these projects involve much of the some basic human interactions yet also require some unique skills and knowledge. As a result, designers tend to specialize in certain type of projects and have skill centered around their expertise, whether that will be software design, user research, web design and industrial design.

✔ Fundamentals of User Interface Design

The various fundamentals of user interfaces design are as follows :

1. Know your user : Your user's goal are your goals, so learn them, restate them, repeat them. Then, learn about your user's skills and experience, and what they need. Find out what interfaces they like and sit down and watch how they use them. Do not carried away trying to keep up with the competition by mimicking trendy design styles or adding new features. By focusing on your user first, you will be able to create an interface that lets them to achieve their goals.

2. Pay attention to Patterns : Users spend a majority of their time on interafces other than your own. There is no need to reinvent the wheel. Those interfaces may solve some of the same problems that users perceive within the one you are creating.

3. Stay Consistent : Users need consistency. They need to know that once they learn to do something, they will be able to do it again. Language, layout and design are just a few interface elements that need consistency. A consistent interface enables your users to have a better understanding of how things will work, increasing their efficiency.

4. Use Visual Hierarchy : Design your interface in a way that allows the user to focus on what is most important. The size, color and placement of each element work together, creating a clear path to understanding your interface. A clear hierarchy will go great lengths in reducing the appearance of complexity.

5. Provide Feedback : Your interface should at all times speak to your user, when his/her actions are both right and wrong or misunderstood. Always inform your users of actions, change in state and errors, or exceptions that occur.

6. Be Forgiving : No matter how clear your design is, people will make mistakes. Your user interface should allow for and tolerate user error. Design ways for users to undo actions, and be forgiving with varied inputs. Also, if the user does cause an error, use your messaging as a teachable situation by showing what action was wrong and ensure that he/she knows how to prevent the error from occuring again.

7. Empower Your User : Once a user has become experienced with your interface, reward him/her and take off the training wheels. The breakdown of complex tasks into simple steps will become cumbersome and distracting. Providing more abstract ways to accomplish tasks, that will allow your design to get out of the way.

8. Speak their Language : All interfaces require some level of copy writing. Keep things conversational, not sensational. Provide clear and concise labels for actions keep your messaging simple.

9. Keep It Simple : The best interface design are invisible. They do not contain unnecessary elements. Instead, the necessary elements are succinct and make sense. Whenever you are thinking about adding a new feature or element to your interface, ask the question, "Does the user really need this?" Are you adding things because you like or want them? Never let your user interfaces ego steal the show.

10. Keep Moving Forward : When creating a user interface, you will make mistakes. Just keep moving forward, and remember to keep your user interface out of the way.

✓ Process of User Interface Design

The process of user interface design includes following steps.

Step 1. Functionality Requirements Gathering

It assembles a list of the functionality required by the system to accomplish the goals of the project and potential needs of the users.

Step 2. User Analysis

Analysis of the potential users of the system can be determined either through discussion with the people who work with the user and/or the potential users themselves. Typical questions involves.

(a) What would be user want the system to do?

(b) How would the system fit in with the user's normal workflow or daily activity?

(c) How technically sawy is the user and what similar systems does the user already use?

(d) What interface look and feel styles appeal to the user?

Step 3. Information Architecture

It includes development of the process and/or flow of the system (*i.e.*, for phone tree systems, this would be an option tree flow chart and for websites, this would be site flow that shows the hierarchy of the pages).

Step 4. Prototyping

It includes development of wire frames, either in the form of paper prototypes or simple interactive screens. These prototypes are stripped of all look and feel elements and most content in order to concentrate on the interface.

Step 5. Usability Inspection

It includes letting an evaluator to inspect a user interface. This is generally considered to be cheaper to implement than usability testing and can be used early on in the development process since it can be used to evaluate prototypes or specifications for the system, which usually cannot be tested on users. Some common usability inspection methods include cognitive walk through, which focuses on the simplicity to accomplish tasks with the system for new users, heuristic evaluation, in which a set of heuristics are used to identify usability problems in the user interface design.

Step 6. Usability Testing

It is the testing of the prototypes on an actual user-often using a technique called think aloud protocol, where you ask the user to talk about their thoughts during the experience.

Step 7. Graphic Interface Design

It may be based on the findings developed during the usability testing if usability is unpredictable, or based on communication objetives and styles that would appear to the user.

✔ Prototyping

Prototyping is the technique of constructing a partial implementation of a system so that customers, users, or developers can learn more about a problem or a situation to that problem. It is a partial implementation because if it is full implementation, it would be the system, not a prototype of it. Prototype allows the users to explore and criticize proposed system before undergoing the cost of a full scale development. The field of prototyping software systems has emerged around two prototyping technologies *i.e.*, throw-away and evolutionary.

In throw-away approach, the proto-type software is constructed in order to learn about the problem or its solution and is usually discarded after the desired knowledge is gained.

In the evolutionary approach, the prototype is constructed in order to learn about the problem or its solution in successive steps. Once the prototype has been used and the required knowledge is gained, the prototype is then adapted to satisfy the requirements, now better understood as needs.

The various benefits of developing a prototype early in the software process are :

(a) Misunderstanding betwen software developers and customers may be identified as the system functions are demonstrated.

(b) Missing user requirements may be detected.

(c) Difficult to use or confusing user requirements may be identified and refined.

(d) A working system is available quickly to demonstrate the feasibility and usefulness of the application to the management.

(e) The prototype serves as a basis for writing the specification of the system.