

Business requires essay



**ASSIGN
BUSTER**

It outlines the various costs involved in fetching the right opportunity and the associated risks in place. The costs involved in procuring raw materials, manufacturing processes, various tax rules and marketing costs would outweigh the cost of the final products and may pose a risk to business continuity. An idea would be to set up a website for this company and allow other organizations to advertise through their products. This would pull customers to their website and fetch extra revenue for making the right number of clicks. 1.

Process model and design choices Using a unique system design and model is utmost essential for getting the requirements on the particular software and would make sure that optimum utilization of time, effort and money is done at the same time. The following sections evaluate the various system development paradigms which would be required to map the requirements of the system. Waterfall model: It follows a planned approach. It takes into account all the stages required to develop the project. The advantage of waterfall development is that it allows for departmentalization and managerial control.

A schedule can be set with deadlines for each stage of development and a product can proceed through the development process in periodic manner, and theoretically, be delivered on time. Development moves from concept, through design, implementation, testing, installation, troubleshooting, and ends up at operation and maintenance. Each phase of development proceeds in strict order, without any overlapping. The challenges of waterfall development are that it does not allow for much reflection or revision. Once

an application is in the testing stage, it is very difficult to go back and change something that was not well-thought out in the concept stage.

It cannot satisfactorily handle the different types of risks that a real life software project is subjected to. To achieve better efficiency and higher productivity, most real life projects cannot follow the rigid phase sequence imposed by the waterfall model. Spiral model: The spiral model is an evolutionary software process model where the developer and the customer better understand and react to risks at each evolutionary level. The spiral model uses prototyping model as a risk reduction mechanism and enables the developer to apply the prototyping approach at any stage in the evolution of the product.

This precisely helps to get a view of the system and get it approved by the customer to match his specifications. It uses the concept of waterfall model to maintain a systematic stepwise approach but in a little different way. The waterfall model here helps in processing the individual steps before stepping into another one. It incorporates this stepwise approach into an iterative framework that more realistically reflects the real world.

The waterfall model is accommodated where there is low specification risk and no need for prototyping for risk resolution. Thus the activities of the second quadrant of the spiral model can be skipped. In another pass of the loop the same stages are revisited and both the waterfall and prototyping model works once again. The following analysis of the spiral model would contemplate the coverage of whether it can be taken up as a viable option in designing a system.

The business environments are seldom volatile in nature where the requirements are ever changing and are adding every moment to expand their scope and stay competitive in the market. The spiral model would make sure that the proposed system is prone to changes. It provides enough mechanism to take into account the changes in the customers' environment and act accordingly. Unlike other SDLC methods where the business requirements are completely taken as standstill when taking into account the building of the system. Unlike the waterfall model where the design stages are non-reversible in nature, once the stage is surpassed it cannot be revisited again. The weaknesses of the spiral model are that it is quite expensive in nature as every deployment is revisited with the spiral cycle once it receives a customer communication and feedback.

It demands quite risk assessment expertise. The time taken to build the complete system is considerable high as for every new requirement is cycled all through from planning to implementation. The consumption of resources is substantially large as all the developmental stages are followed for creating a version of the system in every cycle. The opportunities of using the spiral model are relatively high in comparison to the other models in use. The business external environment is fast changing to the requirements and enveloping newer challenges to analyze their market and take risks to prosper. In this situation having a flexible system development strategy would make it easy to design and build a system susceptible to frequent changes with changing business requirements.

Taking into account the other methods available in the market, the spiral model would stand out in the crowd. The other prototyping methods like RAD

and Incremental methods of SDLC pose a threat to spiral method. The incremental approach is quite complex in operation but its flexibility is quite high in solving discrepancies. The RAD methodology is quite popular in SDLC but is not prone to taking the frequent changes into consideration which is a drawback.

The proponents of the spiral model are built up with risk assessment and management strategies which is not brought forward with any other SDLC methodology. It is risk driven approach rather than document or code driven (Boehm, 1998). It follows a strategy to minimize risks with the repeated use of prototypes in every cycle it loops. The risk assessment is the biggest feature during the development of a software productivity tool. The risks of using this execution method are quite helpful as the changing business environmental events are taken into the picture. The choice of the design methodology is the understanding the business requirements and deciding the crucial methods to be used for implementation process.

For large projects where the requirements are easily documented and are not likely to change over time usually employs waterfall model and other development methods (Sommerville, 2005). Specific situations where business external and internal factors are changing quite regularly, spiral model is taken as an execution process for analyzing the requirements correctly and framing the needs with risk analysis model and design techniques. Other SDLC methods also find a place where development time is less and time is not a luxury. Information engineering and RAD Information engineering is a technique which focuses on an organization's information

requirements and generic processes which are quite flexible for future changes.

In contrast to waterfall model which involved method that did not take care much of the organization's requirements but focused too much on the product development which makes it closed to changes, the former lays a good amount of emphasis to organization's data. Rapid application development (RAD) techniques reduce the time complexity of the waterfall methods by prototyping methods and other techniques. Unlike waterfall and information engineering methods it creates joint application development sessions to understand the user requirements better and decide on functionality for design and usability. In place, RAD addresses the above problems of not involving the users and makes sure that the stakeholders of the system are able to meet and discuss on all grounds including design and look and feel. RAD also encompasses methods which take care of the organizations current processes.

The phased approach of RAD makes it unique to address the present and emergency requirements of the organization and segregating the entire system into sections for implementation. In other methods the information system is not broken into parts and developed as a whole. In comparison, RAD identifies the crucial components which the business requires urgently and develops them first and then takes care of other components in phased manner to complete the entire system.