**Theory-Practice Correspondence Document**

Of

NU-BigEye

Team:

M.Pradyumna

M.Himaja

Mahesh Kumar

Deepika Tripathi

Eklavya Mishra

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Key Concepts** | Relevance | **Real World**  **Contexts** | **Interdisciplinary**  **Connections** | **Critique** | **Technology,**  **Tools and**  **Test Cases** | **Project**  **Management** | **Project Sketch** |
| Introduction  software development | **Software**  **engineering** is an  **engineering**, for  designing and writing  programs for  electronic devices.If  the quality of the  software is good then  it will be easy for the  user and is called as  efficient software. | Suppose lets say a  hospital needs an  application or a  software which  connects to people,  then it will be very  useful for both in their  lives. | An Interdisciplinary  approach presents  software  engineering as a  strategic,  business-oriented,  interdisciplinary  endeavor, rather  than simply a  technical process. | The main concepts are V-shape madel,iterative model,spiral.  The objectives of testing  are changing , and  specific for each test  level. | The objectives of testing  are changing , and  specific for each test  level | We planned our  project to be  students help desk.  Nov 26th our code  will be ready and  testing will be done. | This website has many  contents like  placement  news ,student  achievements,previou  s question  papers,discussion  form,aptitude.. |
| Introduction to Software Development  Life-cycle | The life cycle defines  a methodology for  improving the quality  of software and the  overall development  process. | It consists of a detailed  plan describing how to  develop, maintain,  replace and alter or  enhance specific  software. | This can be used for  any project as all  the projects go  through a life cycle  that needs to be met  at the various  stages. | Planning for the quality  assurance requirements  and identification of the  risks associated with the  project is also done in the  planning stage. | GitHub,  • Google  Docs,  • Visual  Studio,  • Creately.  • compilers,  • interpreters,  • debuggers | Developers have to  follow the coding  guidelines defined  by their  organization . | Started following the  lifecycle and project will be  completed by 23trd  November, 2016 . |
| Various Software  Development  Methodologies | Waterfall Model:  We had to make some  Changes in our  design. Therefore, we did not choose  this model.  Prototype Model:  We did not  Require any  Prototypee for our customers, so prototype model was not  required.  Incremental  Model:  We took the  Reviews of our  customers and  started working  on this website  and did  Spiral model is  Used when we  need to make  changes in our  system even  after the completion of the website  so that it  could be improved as per the needs of our customers.  Further  demands of  our clients,  the number  of developers,  the total working hours  available are  the factors  which made  us choose  the spiral  model.  1. Requires a  Number of  Changes even  after the completion.  2. Takes longer  Than other  Models as  Modifications are required.  not take  frequent reviews until  completion so we did not  choose this  model.  Spiral  Model:  We had to make certain changes as per the need of our customers. Hence, we had  to choose this  model.  RAD  Model:  We did Not  Require sufficient human resource. | Spiral  model is  used when we need to  make changes in our system  even after the completion of the website  so that it  could be  improved as per the needs of our  customers. | Further demands of our clients, the number of  developers, the total  working hours available are the factors which made us choose the spiral model. | 1. Requires a  Number of changes even  after the completeon.  2. Takes longer than  Other models  As modifications are required |  |  |  |
| Software  Requirement  Specification | The software requirements specification document enlists enough and necessary requirements that are required for the project development. | An example organization of an SRS is as follows:  Purpose  Definitions  System overview  References | The Software Requirements Specification (SRS) is a communication tool between stakeholders and software designers. The specific goals of the SRS are:   * Facilitating reviews * Describing the scope of work * Providing a reference to software designers (i.e. navigation aids, document structure) * Providing a framework for testing primary and secondary use cases * Including [features](https://en.wikipedia.org/wiki/Software_feature) to customer requirements * Providing a platform for ongoing refinement . | The SRS may be one of a contract [deliverable](https://en.wikipedia.org/wiki/Deliverable) [Data Item Descriptions](https://en.wikipedia.org/wiki/Data_Item_Descriptions)or have other forms of organizationally-mandated content. | Self analysis. |  |  |
| System Design  **-** Abstraction  -Modularity  -Coupling  -Cohesion  -Top Down | the process of taking away or removing characteristics from something in order to reduce it to a set of essential characteristics.  Modularity is an important Software Engineering principle which works by dividing a complex system into simpler and more manageable modules.  Coupling is a measure that defines the level of inter-dependability among modules of a program. It tells at what level the modules interfere and interact with each other.  Cohesion is a measure that defines the degree of intra-dependability within elements of a module.  In top down strategy we start by testing the top of the hierarchy and we incrementally add modules that it calls and then test the new combined system. | programmer hides all but the relevant data about an [object](http://searchsoa.techtarget.com/definition/object) in order to reduce complexity and increase efficiency.  Each module has its own defined functionality but when they are put together synergistically, the complete functionalities of a are realized.  When a module can directly access or modify or refer to the content of another module, it is called content level coupling.  The greater the cohesion, the better is the program design.Cohesion of any software determines the strength of its source code.   A module cannot be tested in isolation because they invoke some other modules. | the programmer tries to ensure that the [entity](http://whatis.techtarget.com/definition/entity) is named in a manner that will make sense and that it will have all the relevant aspects included and none of the extraneous ones.  Modularization can take place in two ways: composition or decomposition.  When multiple modules have read and write access to some global data, it is called common or global coupling.  It can further be comprehended through exploring its associative principles, types, and calculating methods.  An extensive planning and research phase that leads into the development of a product. | Abstraction is related to both [encapsulation](http://searchnetworking.techtarget.com/definition/encapsulation) and [data hiding](http://searchsqlserver.techtarget.com/definition/data-hiding).  A module is a self-contained component of a larger software system. This doesn't mean that it is an atomic component.  Two modules are called control-coupled if one of them decides the function of the other module or changes its flow of execution.  The concept of cohesion is introduced to capture the 'quality', 'conciseness' and 'effectiveness' features of a program into software engineering domain.  The assembly of a product is done on a system by system basis. | The use of abstractions during design gives the designer freedom to ignore certain details,for the time being, and to determine or design the “bi picture” aspects of his design.  Each task forms a separate, distinct program module. At implementation time each module and its inputs and outputs are well-defined.  This concept makes it easy to determine how closely-linked each module of software program is, which indicates how rapidly it can perform different tasks assigned to it.  This approach allows structured control of a project. | The omitted details can be attributes, relationships among sub-entities or sub-entities. |  |
| Coding | General tasks are handled by high level functions and for specific tasks other functions are used.  -helpful in keeping track of repetitive tasks like loops or function calls.  -minimizing the complexity between different modules of the software | .-breaking down of large modules into smaller modules helps in keeping the code clean, organised and also makes bug-handling easier.  -reduces statements, multiple entry and exit points  -details of a function that are not required to be known to other parts of the software are hidden from being accessed. | -the code gets easier to read  -future changes are easily implementable  -due to information hiding modules are connected to a particular section of the code and not to the entire software hence making changes easier. | -as the working of a module may require other modules to function, this dependency can risk implementation of data structures  -restricted to top-down approach  -Modules without information hiding affect other modules |  |  |  |
| Software  Project  Management | Project  Scheduling ,Staffing ,  Software configuration  management risk management ,quality  assurance are the key concepts of the SPM. | Software project  management  comprises of a  number of activities,  which contains  planning of project,  deciding scope of  software product,  estimation of cost in  various terms,  scheduling of tasks  and events, and  resource  management. Project  management  activities may include:  • Project Planning  • Scope Management | We will come to know,  •Managing People  • Act as project leader  • Liaison with  stakeholders  • Managing human  resources  • Setting up reporting  hierarchy etc. | All the activities,  process need to be  done in order to  make a deliverable  software product.  Software size may  be estimated either  in terms of KLOC  (Kilo Line of Code)  or by calculating  number of function  points in the  software. | Decomposition technique:  line of code  Function points.  • Empirical estimation  technique:  Putnam model  cocoa model. | Techniques that are  use in this are:  Software size  estimation,  effort estimation,  Time estimation,  cost estimation. | For scheduling a  project, it is necessary  to -  • Break down the  project tasks into  smaller,  manageable form  • Find out various  tasks and correlate  them |