**Theory-Practice Correspondence Document**

Project:

Label Printing and Barcode Generator

Organisation:

Advanced Software

Team:

Priyanshu Bhatnagar

Rashika Bhargava

Simran Malhotra

Nirali Sampat

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Key Concepts** | **Relevance** | **Real World Contexts** | **Interdisciplinary Connections** | **Critique** | **Technology, Tools and Test Cases** | **Project Management** | **Project Sketch** |
| **Introduction to Software Development Life Cycle** | This phase is important in getting to know how we will be proceeding with all the stages of the project, the software that we would be making. | It is necessary for all developers to follow a lifecycle for their software. All professional software companies and developers are expected to follow this. | 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. This helps in keeping track of the progress of the project. | This helps the members in analysing the requirements and mapping out the path of the project development, but might fail as members might not take it seriously. | GitHub, Google Docs, Visual Studio, Creately | Planned to follow the life cycle by 14th August,2016 | Started following the lifecycle and project completed by 25th November, 2016 |
| **Various Software Development Methodologies** | * **Waterfall Model:** As the requirements changed, so waterfall is not the required model. * **Prototype Model:**As no such such prototype was build for the customer to evaluate, rather a clear guidance is taken from the customer and worked upon. * **Spiral Model:**As the constant interaction with the customer after every single progress is not performed, hence this model is not chosen. * **Incremental Model:** This is the most suitable as the software is made in increments .After completion of work on one part , the second system is handled and in this way the project progresses. * **RAD Mode**l: No sufficient human resource. | Incremental Model is used in shrink-wrap large applications and systems which built-in small phases or segments.  Shrink Wrap Applications like Food and Beverage Apps, Printing & Publication Apps etc. | The iterative procedure,the client involvement , the suggested requirements, the number of developers working and the time availability for the deployment are the various factors which made us choose Incremental Model for our project. | Few disadvantages of this methodology:   * Requires heavy documentation· * Follows a defined set of processes· * Defines increments based on function and feature dependencies· * Requires more customer involvement than the linear approaches· * Partitioning the functions and features might be problematic· Integration between iteration can be an issue if this is not considered during the development. | Self Analysis | * The project was finalized and * The members were decided. * The requirements were specified and some were not clear * And some requirements were thought of with the project’s progress * Hence the project model was decided in the August before the SRS document was made. | * The customer contacted and , * The project was finalized and * The specifications like customized design , easy interface etc. were specified and * As some aspects were undecided so the work was proceeded as the requirements were proposed by the customer. So Incremental model was decided to be the Software Development Methodology. |
| **Software Requirement Specification**  -Problem Analysis**:**  -Requirement Specification: | * Our project mainly focuses on generating Label and Barcode. Many companies in the base have their requirement specific apps for label generation and printing which generates enough expenditure and smaller companies have no access to such things.Hence, this open source app will be solution to it. * Simple UI,Portability,Custom design,Saving Design,Printing and Design Optimization | * Problem Analysis is very much important in real world as to complete a task we need to have all the objectives and aim clear in mind .Completing task will be much easy if the objectives will be clear. * The Requirement specification is done basically in every development based company and is very useful for knowing customer’s opinion. | * First we meet the customer and understand the problem, than the project team discuss and analyze it. * After the problem analysis, the requirements are decided by the agreement of the customer and tools,softwares are decided. | * In this it is analyzed that if the problem solving techniques are proper or not and examined why the alternatives were not selected. * In this it is checked if all the requirements are fulfilled and if any more requirements are there and if any changes are required. | * Self Analysis,   Word,Google Doc etc.   * Self Analysis,   Word,Google Doc etc. | * The team decided its aims and requirements and hence developed the whole SRS on 26 August 2016. | * The members were decided and * the customer was contacted. * The requirements and the purpose , * And the scope of the project was written * Hence SRS was documented. |
| **System Design**  **-**Abstraction  -Modularity  -Coupling  -Cohesion  -Top-Down Design | -Abstraction is an important design tool as it extracts only the relevant information and ignores the rest.In our project different screens will show only required information and rest will be hidden.  -Modularizing a design helps to plan the development in more effective manner. Different features are added as a separate function in our project.  -Coupling is helpful to measure the level of inter-dependability among modules. Some of the functions in our project are dependent on each other.  -Cohesion is helpful to measure the level of intra-dependability within elements of module.  -In our project it was easy to implement as compared to bottom-up design. | -It is necessary for developers for hiding irrelevant details so that  one can focus on important things at a time.  -It is necessary as it helps in testing and debugging effectively.  -It is necessary as it tells at what level the modules interact with each other.  -It is necessary as it decides how well modules fits together.  -It is necessary as we know our requirements beforehand. | -It allows to control the complexity of design process by proceeding from abstract design model to concrete design model.  -It allows the easy maintenance without affecting the functionality of the software.  -If we need only one field of record then there is no need to pass all the records.  -Thorough knowledge of the functionality of every component is required.  -Requirements should be clear to the developer then only one can proceed. | -Can use it without knowing how it is implemented. Some important components can be left out .  -Greater the number of modules, greater will be the effort to integrate them.  -Lower coupling will lead to better program.  -Higher cohesion will lead to better program.  -It is usually more effective for smaller programs. | Argouml, Staruml,  Visual Studio, Photoshop.  -Self Analysis  -Self Analysis  -Self Analysis  -Self Analysis | -Brainstorming was done to decide what to show and what to hide.  -Project was divided into different modules.  -Figured out the dependency of modules.  -Figured out the dependency within modules.  -After observing the model design was decided. | -Meetings were done and all these concepts were applied.  -Meetings were done and all these concepts were applied.  -Meetings were done and all these concepts were applied.  -Meetings were done and all these concepts were applied.  -Meetings were done and all these concepts were applied. |
| **Coding**  **-**Top-Down Programming  -Structured Programming  -Information Hiding | -Modules at the top level performs general tasks and proceed to other modules to perform particular task.  -We chose this as it is helpful when we need some repetitive tasks in our program. It helped in some of our features like adding text boxes etc.  -It is used to minimize the complexities among different modules of software. | -Program is broken into smaller modules so it is easy to trace a particular segment of code in software program.  -It helps in reducing statements, multiple exit and entry points from the program.  -It focuses on hiding non-essential details of function in a program so that they are inaccessible to other components of the software. | -It makes the functions and procedures globally visible.  -It makes the software code easy to modify when required.  -After using information hiding, modules are connected with a specific section of program and not the whole program. | -There is a risk of implementing data structures as modules are dependent on each other.  -It is restricted to top-down approach of coding.  -Modules created without using information hiding  Affect other modules. | Visual Studio, Github  -Self Analysis  -Self Analysis | -We knew the requirements beforehand so we chose top-down programming.  -Decided the possibility of repetitive functions.  -Relevant data was decided according to different screens. | -Started coding after all the design documentation part was done.  . |
| **Testing** | Testing is important because it is useful in finding hidden bugs and errors.  In our project, we did Black box testing for all the modules. | The Testing is required in majorly every project as we need to identify if the software or the module is working properly or not,If it is efficient enough or not and giving out precise outputs or not. | Testing requires a process which needs to be followed->   1. First we do the unit testing 2. Than further integration testing. 3. And at last the System Testing | Its few disadvantages are that:   * it might be draining enough and * The results are often overestimated * Not all properties can be tested . * Reason of failure are not found. * White Box testing cannot be done because UWP does not support White Box Testing. | * Unit Test class * Visual Studio Debugger | * First we identify the Test Cases * We find Equivalence Classes * We decided boundary values * And Run the Test * The procedure started on 19/11/16 | * After the completion of design * The coding part was done . * Then when the satisfying outputs were seen then the project was finally wrapped up by performing testings and writing test cases. * The procedure started on 19/11/16 |
| **Software Project Management**  -Cost Estimation  -Project Scheduling  -Staffing  -Software configuration management  -risk management  -quality assurance | -Cost estimation is an important aspect as any project cannot be made without it being financially feasible.  - It is necessary to come up with a feasible schedule for all the members.  -Hiring efficient and right people for the project is necessary for staffing. | -All professional companies chart out the financial feasibility of a project before proceeding with it, hence, this is a very important step.  -A schedule has to be made to complete the project efficiently with optimum use of resources. | -Understanding of current business scenario, economics and finances is a must. It helps in also estimating costs for other real life projects.  -It helps in designing schedule for other projects as well. | -HR members and project team members should be well versed in finances to use resources optimally. | -Self Analysis | -Since we are using all our own resources and open source we have not spent money on the project, so the only thing that needed management was time. | -Zero cost project. |