**THEORY-PRACTICE CORRESPONDANCE DOCUMENT.**

**FOR**

**AGILEPDS**

**Prepared By:**

Saurav Singh

Ronak Jain

Raaghav Mehta

Siddhant Misra

Shiv Bajpai

**NIIT UNIVERSITY**

* **Introduction**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Key Concepts** | **Significance and Relevance** | **Real-world Contents** | **Inter-disciplinary connections** | **Critical Thinking** | **Technology, Tools and Techniques** | **Plan Project management** | **Project Sketch** |
| To define Software Engineering | Software Engineering defines how the internal modulation of how a software works and how it interacts with hardware. | Software Engineering helps the students to understand the making and designing of software. | It links the software with hardware modules and allows the module to run across cross platforms. | Software Engineering is different from other respective studies as it helps in differentiating the design of software. | Involves learning about basic designing and structures of software. | Helps the project from the beginning to the end. Basic design of the project. | We have integrated all the earlier mentioned requirements into the project to deliver better performance on both ends. |

* **Introduction to Software Development Life-cycle**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Key Concepts** | **Significance and Relevance** | **Real-world Contents** | **Inter-disciplinary connections** | **Critical Thinking** | **Technology, Tools and Techniques** | **Plan Project management** | **Project Sketch** |
| Requirements analysis, software design, coding, testing, maintenance, etc  We chose SDLC Model. | A software development life cycle model is a descriptive and diagrammatic representation of the software life cycle. | Helpful in analyzing the order of tasks in project stages. | It also captures the order in which software activities are to be performed. | Several models interface different types of procedures. | Life cycle model represents all the activities required to make a software product transit through its life cycle stages. | The models are suitable for development of technically challenging and difficult software products that are prone to various kinds of risks. | We incorporated the Spiral Model into the project after researching and correspond to our methodology to the module. |

* **Various Software Development Methodologies**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Key Concepts** | **Significance and Relevance** | **Real-world Contents** | **Inter-disciplinary connections** | **Critical Thinking** | **Technology, Tools and Techniques** | **Plan Project management** | **Project Sketch** |
| Prototyping, interactive, waterfall model, enhancement, spiral model. Role of Management in software development. Role of metrics and measurement. | A descriptive and diagrammatic representation of the software life cycle. | The entry and exit criteria for every phase. A phase can begin only if its phase-entry criteria has been satisfied. | Without software life cycle model the entry and exit criteria for a phase cannot be recognized. | Develop and validate the next level of the product after resolving the identified risks. | Progressively more complete version of the software gets built with each iteration around the spiral. | The spiral model is perfect for development of technically challenging software products that are prone to several kinds of risks. | We incorporated the Spiral Model into the project after researching and thinking out our methodology through to the module. |

* **Software Requirement Specification**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Key Concepts** | **Significance and Relevance** | **Real-world Contexts** | **Inter-disciplinary connections** | **Critical Thinking** | **Technology, Tools and Techniques** | **Plan Project management** | **Project Sketch** |
| Problem analysis, requirement specification, validation, metrics, monitoring and control. | The main objective of the SRS document is to describe the main requirements and engineering activities and to introduce techniques for requirements elicitation and analysis. | Without the SRS, the definition of the document is not complete. | To describe requirements validation and to discuss the role of requirements management in support of other requirements engineering processes. | Helps the project developers to draw parallels between the requirements so that they can be connected. | Requirement Traceability Matrix and Development Matrices help defining relationships. | Helps to understand the project even closely and in a better form and the SRS helped the developers to design the app more efficiently. | System Features that presented the complete module of the app/project and have been explained. |

* **System Design**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Key Concepts** | **Significance and Relevance** | **Real-world Contents** | **Inter-disciplinary connections** | **Critical Thinking** | **Technology, Tools and Techniques** | **Plan Project management** | **Project Sketch** |
| Problem partitioning, abstraction, top-down and bottom-up design, Structured approach. Functional versus object-oriented approach, design specification and verification metrics, monitoring and control. | It is a formal way of representing how a business system interacts with its environment and illustrates the activities that are performed by the users of the system. | The design phase documents define the way the software is designed. | The design document works a cross between the requirement phase and the actual codes. | Design phase lays the foundation of how software actually gets designed. | The design phase requires the class diagrams, sequence diagrams and state diagrams. | After the design phase gets created, work starts upon the actual coding. | The design document works as an empirical part of the software development. |

* **Coding**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Key Concepts** | **Significance and Relevance** | **Real-world Contents** | **Inter-disciplinary connections** | **Critical Thinking** | **Technology, Tools and Techniques** | **Plan Project management** | **Project Sketch** |
| Top-down and bottom-up, structured programming, information hiding, programming style, and internal documentation. Verification, Metrics, monitoring and control. | The coding is the most intricate part of the software as it basically makes the software work. | Coding makes the software or app actually workable. | The coding makes the software actually go cross platform.  Like using  Bootstrap  For a  Dynamic  Design. | Working with the coding gives the complete idea of how each and every function works out. | The IDE that works on the code, scripts and frame works. | The codes get planned after the documentation on design and requirements and get completed before the testing. | The codes sketch out the whole project all in all. |

* **Testing**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Key Concepts | Significance and Relevance | Real-world Contents | Inter-disciplinary connections | Critical Thinking | Technology, Tools and Techniques | Plan Project management | Project Sketch |
| Levels of testing functional testing, structural testing, test plane, test cases specification, reliability assessment | Goal of testing is finding faults in the software and demonstrating that there are no faults in the software (for the test cases that has been used during testing) | It is not possible to *prove* that there are no faults in the software using testing  Testing should help locate errors, not just detect their presence | Testing the driving test cases automatically from a formal specification of the functional requirements | The number of test cases increase exponentially with the number of input/output variables | J-Unit,Appdynamics,WAP testing tools. We know that if we find an error during unit testing it is in the module we are testing | Testing clearly removes out all the errors from the software plan. | Takes out bugs from the project codes. |

* **Software Project Management**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Key Concepts** | **Significance and Relevance** | **Real-world Contents** | **Inter-disciplinary connections** | **Critical Thinking** | **Technology, Tools and Techniques** | **Plan Project management** | **Project Sketch** |
| Cost estimation; Project scheduling, Staffing, Software configuration management, Quality assurance, Project Monitoring, Risk management, etc. | It is an agile process that allows us to focus on delivering the highest business value in the shortest time. | The business sets the priorities. | It allows us to rapidly and repeatedly inspect actual working software (every two weeks to one month). | The project manage-ment is the most crucial part of the project that defines it. | Agile Scrum method, Continuous meetings and analysis. | Every two weeks to a month anyone can see real working software and decide to release it as is or continue to enhance for another iteration. | We followed this module and did rigorous research every week to develop and enhance the project development process. |