**THEORY PRACTICAL CORRESPONDENCE REPORT**

**FOR**

**REVIEW MANAGEMENT SYSTEM**

**Prepared by RMS members:**

**RISHABH GANGWAR (U101114FCS116)**

**SURAJ PRAKASH SHARMA (U101114FCS144)**

**SANDEEP KUMAR (U101114FCS213)**

**RAJAT SINHA (U101114FCS114)**

**SAJAL KUMAR MISHRA (U101114FCS171)**

**Guided By: Prof. Amit Kumar (CSE Dept. 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** |
| Defining 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 helps the module to run on cross platforms. | Software Engineering is different from other respective studies as it helps to differentiate the designing of software. | Involves learning about basic designing and structures of software. | Helps the project from start to end. Basic designing of the project. | We integrated all the above mentioned requirements into the project so as 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. | Helps in analyzing the order of activities in a project stages. | It also captures the order in which software activities are to be undertaken. | Several models interface different kinds of procedures. | Life cycle model represent all the activities required to make software product transit through its life cycle phases. | The models are suitable 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 corresponds 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** |
| Waterfall model, prototyping, interactive enhancement, spiral model. Role of Management in software development. Role of metrics and measurement. | It is a descriptive and diagrammatic representation of the software life cycle. | It defines entry and exit criteria for every phase. A phase can start only if its phase-entry criteria have 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 suitable 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 corresponding our methodology 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 basicallyto describe the principal requirements engineering activities and to introduce techniques for requirements elicitation and analysis. | Without the SRS, the definition of the document is incomplete. | Plans 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 even 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’s a formal way of sharing the problems with the management team and hence gets the best response. | 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. | 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 | Testing is very important phase which is done when the software is developed.  If you do not do the testing then there is always a high uncertainty in the answer of the question whether the software has bugs or not. | 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 the driving test cases automatically from a formal specification of the functional requirements | The number of test cases increases exponentially with the number of input/output variables because of which it might take long time to check each test case.  It might be draining our resources. | J-Unit 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** |
| The ratings will be recorded and an average will be found on the basis of which proper actions will be taken. The comments and suggestions will be seen and managed accordingly. | 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. | This will improve the connection between the management team and the customers. | 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 in which Continuous meetings and analysis is done. | Every two weeks to a month anyone can see real working software and decide to release it as is or continue to enhance for iteration. | We followed this module and did rigorous brainstorming every week to develop and enhance the project development process. |