**TESTING**

|  |  |
| --- | --- |
| **Key Concepts** | **Explore Concepts significance and relevance** |
| After the code is developed it is tested against the requirements to make sure that the product is actually solving the needs addressed and gathered during the requirements phase.  During this phase all types of [functional testing](http://istqbexamcertification.com/what-is-functionality-testing-in-software/) like [unit testing](http://istqbexamcertification.com/what-is-unit-testing/), [integration testing](http://istqbexamcertification.com/what-is-integration-testing/), [system testing](http://istqbexamcertification.com/what-is-system-testing/), [acceptance testing](http://istqbexamcertification.com/what-is-acceptance-testing/) are done as well as [non-functional testing](http://istqbexamcertification.com/what-is-non-functional-testing-testing-of-software-product-characteristics/) are also done.  In the field of Software testing we use different types of **Software Testing Methodologies**. In today’s Software Testing Class, we will see what all Software Testing Methodologies & software testing technique is used in the day today software life cycle..  In the Software Development Process different software development approaches are used. A software development process also known as a Software Development Life Cycle (SDLC)**.**   * Waterfall model * V model * Agile model * Spiral model * RUP * RAD | Once the application is migrated to a test environment, different types of testing will be performed including integration and system testing.  User acceptance testing is the last part of testing and is performed by the end users to ensure the system meets their expectations.   At this point, defects may be found and more work may be required in the analysis, design or coding.  Once sign-off is obtained by all relevant parties, implementation and deployment can begin.     1. Requirement gathering and analysis 2. Design 3. Implementation or coding 4. Testing 5. Deployment 6. Maintenance |

**Establish Relevance and make sense and meaning**

|  |  |
| --- | --- |
| **Find Real-life contexts** | **Find Interdisciplinary connections** |
| Software life cycle models describe phases of the software cycle and the order in which those phases are executed. Each phase produces deliverables required by the next phase in the life cycle. Requirements are translated into design. Code is produced according to the design which is called development phase. After coding and development the testing verifies the deliverable of the implementation phase against requirements.  Phase of the SDLC in which all possible system requirements are captured & analyzed .Software requirements specification includes the complete information about how actual end users are expecting from the system. This document covers all the necessary requirements for the development of project. Finally after completion of requirement gathering & analysis (validation of requirement against the user needs), a Requirement Specification document is created which give out as a input to the next phase of SDLC. In this model once we moved to the next phase then it won’t possible to add or update the requirements. | Interdisciplinary involves the combining of two or more into one activity Business requirements are gathered in this phase. This phase is the main focus of the project managers and stake holders. Meetings with managers, stake holders and users are held in order to determine the requirements like; Who is going to use the system? How will they use the system?  What data should be input into the system?  What data should be output by the system?  These are general questions that get answered during a requirements gathering phase. After requirement gathering these requirements are analysed for their validity and the possibility of incorporating the requirements in the system to be development is also studied. Finally, a Requirement Specification document is created which serves the purpose of guideline for the next phase of the model. In this phase the system and software design is prepared from the requirement specifications which were studied in the first phase. System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture. The system design specifications serve as input for the next phase of the model. Software life cycle models describe phases of the software cycle and the order in which those phases are executed |

|  |  |
| --- | --- |
| **Engage in critical thinking** | **Technology, Tools, Techniques** |
| Comparison and contrast:  **Waterfall model:**  Advantage-   1. Easy to explain to the user. 2. Stages and activities are well defined. 3. Helps to plan and schedule the project. 4. Each phase has specific deliverables.   Dis- advantage -   1. Assumes that the requirements of a system can be frozen· 2. Very difficult to go back to any stage after it finished. 3. Little flexibility and adjusting scope is difficult and expensive. 4. Costly and required more time, in addition to detailed plan.   **lterative model:**  Advantage-   1. Produces business value early in the development life cycle· 2. Better use of scarce resources through proper increment definition· 3. Can accommodate some change requests between increments·   Dis-advantage:   1. Requires heavy documentation· 2. Follows a defined set of processes· 3. Requires more customer involvement than the linear approaches·   **Spiral model:**  Advantage-   1. Estimates (i.e. budget, schedule, etc.) become more realistic as work progresses. 2. Early involvement of developers· 3. Manages risks and develops system into phases   Dis-advantage:   1. High cost and time to reach the final product. 2. Needs special skills to evaluate the risks and assumptions· 3. Highly customized limiting re-usability   **Prototyping Model:**  Advantage-   1. Reduced time and costs, 2. Improved and increased user involvement   Dis-advantage:   1. User confusion of prototype and finished system· 2. Excessive development time of the prototype· 3. Expense of implementing prototyping | Business requirements are gathered in this phase. This phase is the main focus of the project managers and stake holders. Meetings with managers, stake holders and users are held in order to determine the requirements like; Who is going to use the system? How will they use the system?  What data should be input into the system?  What data should be output by the system?  These are general questions that get answered during a requirements gathering phase. After requirement gathering these requirements are analyzed for their validity and the possibility of incorporating the requirements in the system to be development is also studied.  In this phase the testers comes up with the [Test strategy](http://istqbexamcertification.com/what-are-the-test-approaches-or-strategies-in-software-testing/), where they mention what to test, how to test. On receiving system design documents, the work is divided in modules/units and actual coding is started. Since, in this phase the code is produced so it is the main focus for the developer. This is the longest phase of the software development life cycle. As soon as the product is given to the customers they will first do the [beta testing](http://istqbexamcertification.com/what-is-beta-testing/). If any changes are required or if any bugs are caught, then they will report it to the engineering team. Once those changes are made or the [bugs](http://istqbexamcertification.com/what-is-defect-or-bugs-or-faults-in-software-testing/) are fixed then the final deployment will happen. Once when the customers starts using the developed system then the actual problems comes up and needs to be solved from time to time. This process where the care is taken for the developed product is known as maintenance.  There are following six phases in every Software development life cycle model:   1. Requirement gathering and analysis 2. Design 3. Implementation or coding 4. Testing 5. Deployment 6. Maintenance |

|  |  |
| --- | --- |
| **Plan project management** | **Project specification and project brief** |
| The Project Matrix is a model of the software development work of a project which can be used for project management. The matrix provides a single framework for organizing, relating, and viewing several diverse aspects of the project. It has been used successfully on several projects, serving a variety of purposes including:  ● providing a framework for project planning.  ● Identification of intermediate and final deliverables.  ● providing a systematic method for deriving a work breakdown structure  ● providing a framework for tracking progress in terms of completed/not completed status of all activities  ● supporting the tracing of requirements through all stages of software development | Our project specification is to ensure everyone to know the details of the unknown persons by clicking their picture. This application works similar to true caller and the deference is where you need to have is picture instead of phone number. This makes things easier. All security and privacy features are taken care of. In real world you make have your friends’ picture but you don’t know anything about him now including phone numbers, address, working etc..,. To make this thing happen and to connect each other we you this application to get the details of that desired ones. The role of student is just to advertise this application and make all of them to part in this by creating account in face polygon and use it wisely |

**Software project management**

|  |  |
| --- | --- |
| **Key Concepts** | **Explore Concepts’ significance and relevance** |
| Software Development Life Cycle, SDLC for short, is a well-defined, structured sequence of stages in software engineering to develop the intended software product.  **Factors which define this are:-**  Communication  Requirement Gathering  Feasibility Study  System Analysis  Software Design  Coding  Testing  Integration  Implementation  Operations & Maintenance  Disposition  **Important sustaining concepts:-**  Waterfall model  Iterative model  Spiral model  V – model  Big bang model | The purpose of the SDLC phases is to break the system down into developmental phases to better track productivity and progress.  You will be able to more effectively guide the development team (coders, testers and technical writers) through all of the SDLC. Your development projects will be easier to plan, control and delver. Team productivity will go up while development risk goes down.  **To avoid unnecessary development risks such as:**  ·Incomplete or incorrect business requirements  ·Unintended impediments  ·Uncontrolled requirement changes  ·Incomplete testing  ·Not having a Software Requirements Specification  ·Many other self-inflicted wounds |

**Establish Relevance and make sense and meaning**

|  |  |
| --- | --- |
| **Find Real-life contexts** | **Find Interdisciplinary connections** |
| In addition, the SDLC also provides a detailed guide to help Program Managers with ALL aspects of IT system development, regardless of the system size and scope. The SDLC contains a comprehensive checklist of the rules and regulations governing IT systems, and is one way to ensure system developers comply with all applicable Government regulations, because the consequences of not doing so are high and wide ranging. This is especially true in the post 9/11 environment where larger amounts of information are considered sensitive in nature, and are shared among commercial, international, Federal, state, and local partners. | A majority of programming courses offered by computer science departments focus on application design and implementation. While this type of programming is useful for the computer science (CS) student, it is not as beneficial to the information technology (IT) student. Students planning to work as IT professionals need a different set of skills than students planning to work in a software developer capacity. An IT discipline-specific perspective should emphasize a practical hands-on approach to install, secure, maintain, and upgrade an organization's portfolio of application programs. In contrast, the traditional CS teaching approaches are geared more at the analysis, design, and implementation of application development. Given this dichotomy, how can instructors' best teach the software development life-cycle (SDLC) process to a class comprised of both IT and CS students? This paper discusses the types of skills needed in the software community and describes a successful teaching approach to teach those skills to a capstone project interdisciplinary class comprised of both IT and CS students. |

|  |  |
| --- | --- |
| **Engage in critical thinking** | **Technology, Tools, Techniques** |
| Comparison and Contrast:  Waterfall model:  Adv:-  1. Simple to understand and use.  2. Each phase is independent of other phases and is processed and completed separately.  3. Suitable for smaller projects, and for projects where the requirements are clearly outlined.  Dis adv:-  1. No output or working software is produced until late in the life cycle.  2. High degree of uncertainty and risks.  3. Not a good choice for big or ongoing projects.  Iterative model:  Adv:-  1. Produces working software early during the lifecycle.  2. More flexible as scope and requirement changes can be implemented at low cost.  3. Testing and debugging is easier, as the iterations are small.  4. Low risks factors as the risks can be identified and resolved during each iteration.  Dis adv:-  1. This model has phases that are very rigid do not overlap.  2. Not all the requirements are gathered before starting the development; this could lead to problems related to system architecture at later iterations.  Spiral model:  Adv:-  1. Good for large and critical projects.  2. Working software is produced early during the life cycle.  3. Large amount of risk analysis.  Dis adv:-   1. 1. Involves higher cost. 2. 2. Not suitable for smaller projects. 3. 3. Project success depends on the risk analysis phase - hence, it requires highly specific expertise in risk analysis. 4. Prototype model: 5. Adv:- 6. 1. Benefits from user input. 7. 2. As working model of the system is provided, user get a better understanding of the system that is being developed. 8. 3. Errors and risks can be detected at a much earlier stage, as the system is developed using prototypes. 9. Dis adv:- 10. 1. Increases complexity of the overall system. 11. 2. Involves exploratory methodology and therefore involves higher risk. 12. 3. Involves implementing and then repairing the way a system is build, so errors are an inherent part of the development. | Preliminary system study is the first stage of system development life cycle. This is a brief investigation of the system under consideration and gives a clear picture of what actually the physical system is? In practice, the initial system study involves the preparation of a System proposal which lists the Problem Definition, Objectives of the Study, Terms of reference for Study, Constraints, Expected benefits of the new system, etc. in the light of the user requirements. The system proposal is prepared by the System Analyst (who studies the system) and places it before the user management. The management may accept the proposal and the cycle proceeds to the next stage. The management may also reject the proposal or request some modifications in the proposal. In summary, we would say that system study phase passes through the following steps:   * problem identification and project initiation * background analysis * inference or findings |

|  |  |
| --- | --- |
| **Plan project management** | **Project specification and project brief** |
| The intermediate deliverables are just picture details of only the people who registered in the start. In this procedure we need to know any discrepancies and server errors. In the final project delivers the exact and accuracy of our application by giving the details of the person of which you have taken a picture. The progress is monitored by our technical support staff to ensure any privacy errors or to breach into our software. | Our project specification is to ensure everyone to know the details of the unknown persons by clicking their picture. This application works similar to true caller and the deference is where you need to have is picture instead of phone number. This makes things easier. All security and privacy features are taken care of. In real world you make have your friends’ picture but you don’t know anything about him now including phone number, address, working, etc..,. To make this thing happen and to connect each other we you this application to get the details of that desired ones. The role of student is just to advertise this application and make all of them to take part in this by creating account in face polygon and use it wisely. |