|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Key Concepts** | **Explore concepts' significance and relevance** | **Establish relevance, make sense and meaning -Find real-life contexts** | **Establish relevance, make sense and meaning -Find interdisciplinary connections** | **Engage in critical thinking** | **Technology, tools and techniques** | **Plan project management** | **Project specification and sketch** |
| In [software engineering](https://en.wikipedia.org/wiki/Software_engineering), a software development methodology (also known as a system development methodology, software development life cycle, software development process, software process) is a splitting of [software development](https://en.wikipedia.org/wiki/Software_development) work into distinct phases (or stages) containing activities with the intent of better planning and management. It is often considered a subset of the [systems development life cycle](https://en.wikipedia.org/wiki/Systems_development_life_cycle).  There are the following methodologies:-  **1.** Agile Software Development.  **2.** Crystal Methods.  **3.** Dynamic Systems Development Model (DSDM).  **4.** Extreme Programming (XP).  **5.** Feature Driven Development (FDD).  **6.** Joint Application Development (JAD).  **7.** Lean Development (LD).  **8.** Rapid Application Development (RAD).  **9.** Rational Unified Process (RUP).  **10.** Scrum.  **11.** Spiral.  **12.** Systems Development Life Cycle (SDLC).  **13.** Waterfall (a.k.a. Traditional). | Software Development Methodology is important since it determines all ingredients that lead to fail/success, in a software project.  **1.** All the roles within the project and their collaboration is defined by the methodology of your SDLC, Briefly your project management is dependent on it.  **2.** Your project management technique is dependent on the methodology of your SDLC.  **3.** Your requirement management and change management is defined by the methodology of your SDLC. For instance if you are doing waterfall, you have to freeze them in order to mitigate the risk of project fail. But if the changes are too frequent and cannot be frozen then you can go with agile.  **4.** The quality of your software is dependent on the methodology of your SDLC. If your methodology does not cover the required quality tests like unit, functional, performance, robustness, etc. then your software will be suffering from poor quality. But if it covers them especially with a test driven approach then you saved the project. | The software development methodologies have uses in almost every project, except, the ones that require upfront planning and cannot deliver incremental value, such as open-heart surgery.  One software development methodology framework is not necessarily suitable for use by all projects. Each of the available methodology frameworks are best suited to specific kinds of projects.  Sometimes, contractors may require methodologies employed, an example is the U.S. [defences industry](https://en.wikipedia.org/wiki/Arms_industry), which requires a rating based on [process models](https://en.wikipedia.org/wiki/Process_model) to obtain contracts.  A decades-long goal has been to find repeatable, predictable processes that improve productivity and quality. Some try to systematize or formalize the seemingly unruly task of designing software. Others apply [project management](https://en.wikipedia.org/wiki/Project_management) techniques to designing software. | **1.** Configuration management. **2.** Infrastructure as Code.  **3.** Documentation.  **4.** Software Quality assurance (SQA).  **5.** Project management. **6.** User experience.  Above are the supporting interdisciplinary connections. | Before taking part in any verbal confrontation or examination, guarantee that the suggestion being considered is in any event possibly falsifiable. A typical element of non-falsifiable recommendations is unclearness.  Such recommendations can be refined by:-  **1.** Characterizing any expansive or novel wording in the suggestion.  **2.** Making understood quantifiers unequivocal.  **3.** Putting forth certainly relative expressions unequivocally relative.  Here a few examples of non-falsifiable propositions that many would consider credible:  **1.** Open source software is more reliable than commercial software.  **2.** Agile techniques are the future of software development.  **3.** OO programming is better than structured programming.  These three propositions are simply worthless. The subject areas they deal with may well be fruitful areas of investigation, but you will only be able to make progress in your investigations if you refine these propositions into more specific and thereby falsifiable statements. | There are commercial products that include requirements gathering tools, prototyping tools, software development environments such as those for the Java platform, groupware for communication among development members, and testing tools.  Techniques are skill areas such as developing use cases; and policiesdictate organizational “musts”.  Other computerized development tools include:-  **1.** [Compiler](https://en.wikipedia.org/wiki/Compiler).  **2.** [Debugger](https://en.wikipedia.org/wiki/Debugger).  **3.** [Profiler](https://en.wikipedia.org/wiki/Profiling_(computer_programming)).  **4.** [GUI designer](https://en.wikipedia.org/wiki/Graphical_user_interface_builder).  **5.** [Modelling](https://en.wikipedia.org/wiki/UML_tools).  **6.** [IDE](https://en.wikipedia.org/wiki/Integrated_development_environment).  **7.** [Build. automation](https://en.wikipedia.org/wiki/Build_automation)  **8.** [Release. automation](https://en.wikipedia.org/wiki/Application_release_automation)  **9.** [Testing](https://en.wikipedia.org/wiki/Category:Software_testing_tools).  **10.** [Computer Aided Software Engineering](https://en.wikipedia.org/wiki/Computer_Aided_Software_Engineering) (CASE) tools.  **11.** [Database Management Systems](https://en.wikipedia.org/wiki/Database_Management_System) (DBMS).  **12.** [Fourth-generation programming languages](https://en.wikipedia.org/wiki/Fourth-generation_programming_language).  **13.** Code generators.  **14.** Object-oriented techniques. | The software methodology has been decided that is the Agile development.  Immediate deliverables are nothing yet, but in the future, SRS document has to be delivered.  PISE-PBL Sub Project 2 will be form September 21-24. | As mentioned above.  ---------do-------- |