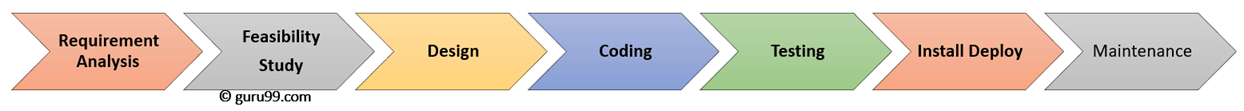
**SDLC Software Development LifeCycle**

технологическое

**WWW.ВАШВЕБСАЙТ.COM**



|  |  |
| --- | --- |
| 1.Планирование системы  Определение проблем ресурсов, альтернативных решений.  После анализа этих данных у вас будет три варианта: разработать новую систему, улучшить существующую или оставить систему как есть | Phase 1: Requirement collection and analysis The requirement is the first stage in the SDLC process. It is conducted by the senior team members with inputs from all the stakeholders and domain experts in the industry. Planning for the quality assurance requirements and recognization of the risks involved is also done at this stage.This stage **gives a clearer picture of the scope of the entire project and the anticipated issues, opportunities, and directives** which triggered the project. |
| 2.Анализ системы  определить и задокументировать требования конечного пользователя системы  на этом этапе **технико-экономическое обоснование +ясное видение конечного продукта и его функций** | Phase 2: Feasibility study  the next sdlc step is to define and document software needs. This process conducted with the help of 'Software Requirement Specification' document also known as **'SRS' documen**t. **It includes everything which should be designed and developed during the project life cycle**.  five types of **feasibilities checks:**  Economic: Can we complete the project within the budget or not?  Legal: Can we handle this project as cyber law and other regulatory framework/compliances.  Operation feasibility: Can we create operations which is expected by the client?  Technical: Need to check whether the current computer system can support the software  Schedule: Decide that the project can be completed within the given schedule or not. |
| 3.Дизайн системы  эта фаза определяет элементы системы, компоненты, уровень безопасности, модули, архитектуру, различные интерфейсы и типы данных, которыми оперирует система. Дизайн системы в общих чертах может быть сделан ручкой на листке бумаги,а потом перенесен в документ | Phase 3: Design  In this third phase, the system and software design documents are prepared as per the requirement specification document. This helps define overall system architecture.  There are two kinds of design documents developed in this phase:  High-Level Design (HLD)  Brief description and name of each module  An outline about the functionality of every module  Interface relationship and dependencies between modules  Database tables identified along with their key elements  Complete architecture diagrams along with technology details  Low-Level Design(LLD)  Functional logic of the modules  Database tables, which include type and size  Complete detail of the interface  Addresses all types of dependency issues  Listing of error messages  Complete input and outputs for every module |
| 4.Разработка, внедрение , развертывание  собственно процесс разработки системы;  **фаза внедрения** будет включать в себя конфигурацию и настройку «железа» под определенные требования и функции. На этой стадии **система готова к установке у заказчика, к запуску в боевом режиме**. Возможно, конечным пользователям потребуется тренинг, чтобы они освоились с системой и знали, как ее использовать. **Фаза внедрения может быть** очень долгой - это зависит от сложности системы. | Phase 4: Coding  It is the longest phase of the Software Development Life Cycle process.In this phase, Developer needs to follow certain predefined coding guidelines. They also need to use programming tools like compiler, interpreters, debugger to generate and implement the code. |
| 5. Опытная эксплуатация и интеграция  сборка , подаем системе различные входящие данные и анализируем выход, поведение и функционирование. Тестирование ( может выполняться настоящими пользователями или специальной командой сотрудников), также оно может быть систематическим и автоматизированным, с тем, чтобы удостовериться, что актуальные результаты работы системы совпадают с предусмотренными и желательными. | Phase 5: Testing  Once the software is complete, and it is deployed in the testing environment. The testing team starts testing the functionality of the entire system. This is done to verify that the entire application works according to the customer requirement.  During this phase, QA and testing team may find some bugs/defects which they communicate to developers. The development team fixes the bug and send back to QA for a re-test. This process continues until the software is bug-free, stable, and working according to the business needs of that system. |
| 6. Поддержка системы  техническая поддержка системы | Phase 6: Installation/Deployment:  Once the software testing phase is over and no bugs or errors left in the system then the final deployment process starts. Based on the feedback given by the project manager, the final software is released and checked for deployment issues if any. |
|  |  |
|  | Phase 7: Maintenance  **Bug fixing** - bugs are reported because of some scenarios which are not tested at all  **Upgrade** - Upgrading the application to the newer versions of the Software  **Enhancemen**t - Adding some new features into the existing software  The main focus of this SDLC phase is **to ensure that needs continue to be met and that the system continues to perform as per the specification mentioned in the first phase.** |
| МОДЕЛИ |  |
| 1.водопадная  2.Vобразная |  |
| 3. Инкрементальная модель в SDLC  серия водопадных циклов. В начале проекта требования разделены на группы. Для каждой группы используется модель SDLC для разработки программного обеспечения. Процесс SDLC повторяется, с каждым выпуском добавляются дополнительные функции, пока не будут выполнены все требования. В этом методе каждый цикл действует как этап обслуживания предыдущей версии программного обеспечения. Модификация инкрементальной модели позволяет перекрывать циклы разработки. После этого следующий цикл может начаться до завершения предыдущего | Incremental Model in SDLC  The incremental model is not a separate model. It is essentially a series of waterfall cycles. The requirements are divided into groups at the start of the project. For each group, the SDLC model is followed to develop software. The SDLC process is repeated, with each release adding more functionality until all requirements are met. In this method, every cycle act as the maintenance phase for the previous software release. Modification to the incremental model allows development cycles to overlap. After that subsequent cycle may begin before the previous cycle is complete. |
| **Методология Agile** - это практика, которая способствует непрерывному взаимодействию разработки и тестирования в процессе SDLC любого проекта. В методе Agile весь проект делится на небольшие инкрементальные сборки. Все эти сборки предоставляются в виде итераций, и каждая итерация длится от одной до трех недель. | **Agile Model in SDLC**  Agile methodology is a practice which promotes continue interaction of development and testing during the SDLC process of any project. In the Agile method, the entire project is divided into small incremental builds. All of these builds are provided in iterations, and each iteration lasts from one to three weeks. |
| **Спиральная модель** - это модель процесса, ориентированного на риски. Эта модель SDLC помогает команде внедрять элементы одной или нескольких моделей процессов, таких как водопад, инкрементальный, водопад и т.д.  Эта модель сочетает в себе лучшие черты прототипной модели и водопадной модели. Спиральная методология представляет собой сочетание быстрого прототипирования и параллелизма при проектировании и разработке. | **Spiral Model**  The spiral model is a risk-driven process model. This SDLC model helps the team to adopt elements of one or more process models like a waterfall, incremental, waterfall, etc.  This model adopts the best features of the prototyping model and the waterfall model. The spiral methodology is a combination of rapid prototyping and concurrency in design and development activities. |
| Модель большого взрыва фокусируется на всех типах ресурсов в разработке и кодировании программного обеспечения без какого-либо планирования или очень небольшого планирования. Требования понимаются и выполняются, когда они поступают.  Эта модель лучше всего подходит для небольших проектов с небольшой командой разработчиков, которые работают вместе. Это также полезно для академических проектов по разработке программного обеспечения. Это идеальная модель, где требования либо неизвестны, либо не указана окончательная дата выпуска. | **Big bang model**  Big bang model is focusing on all types of resources in software development and coding, with no or very little planning. The requirements are understood and implemented when they come.  This model works best for small projects with smaller size development team which are working together. It is also useful for academic software development projects. It is an ideal model where requirements is either unknown or final release date is not given |