**В своем блоге я буду рассматривать темы, которые важны для моего развития и профессионального пути. Одной из базовых и первостепенных тем в компбютерном изучении является Жизненный цикл разработки систем** ( SDLC **). В Википедии он определяется как** процесс планирования, создания, тестирования и развертывания [информационной системы](https://en.wikipedia.org/wiki/Information_system). Согласно Тейлору (2004), «жизненный цикл проекта охватывает все действия [проекта](https://en.wikipedia.org/wiki/Project) , в то время как жизненный цикл разработки систем фокусируется на реализации [требований](https://en.wikipedia.org/wiki/Requirement) к продукту ».  
  
Мое собственное понимание этого термина заключается в такой формулировке: ряд событий, происходящих с системой в процессе ее создания и дальнейшего использования, начиная с зародившейся идеи и до смерти вселенной. Почему так долго, что аж бесконечно? Потому что в реальной жизни окончания разработки проекта не существует – всегда поступают либо новые пожелания от заказчика о расширении функциональных возможностей, либо требуется поддержка пользователям продукта.

Но хоть жизненный цикл разработки ПО и исчисляется в бесконечностях, его можно разделить на шесть четких этапов. В этом отношении первые три этапа стараются дать ответы на проверочные вопросы, а последние три оптимизированы для достижения фактических результатов.

* **Анализ требований** отвечает на вопрос «Какие проблемы требуют решений?» На этом этапе SDLC вам необходимо получить обратную связь и поддержку от соответствующих внутренних и внешних заинтересованных сторон.
* **Планирование** отвечает на вопрос «Что мы хотим сделать?» Это обеспечивает понимание юнит-экономики проекта, факторов снижения рисков и ожидаемых стоимостей.
* **Проектирование и дизайн** отвечает на вопрос «Как мы добьемся наших целей?» На этом этапе важно учитывать операционное превосходство, безопасность, надежность, эффективность производительности и оптимизацию затрат.
* **Разработка ПО** регулирует процесс создания продукта. Для соответствия результата ожиданиям критично при начале разработки следовать первым трем этапам.
* **Тестирование** регулирует обеспечение качественной работы продукта. Выполняются разные виды тестирования: модульное, интеграционное, тестирование интерфейса и другие виды в зависимости от цели. Выводом станет заключение о том, что в продукте нет ошибок и он готов к релизу.
* **Развертывание** и поддержка регулирует использование финального продукта. Продукт можно внедрять и интегрировать со сторонним программным обеспечением. Но разработка на этом не заканчивается, проекту требуется постоянный надзор и поддержка.

Были созданы различные методологии SDLC, такие как [водопад](https://en.wikipedia.org/wiki/Waterfall_model) , [спираль](https://en.wikipedia.org/wiki/Spiral_model) , [agile](https://en.wikipedia.org/wiki/Agile_software_development) , [быстрое прототипирование](https://en.wikipedia.org/wiki/Software_prototyping#Throwaway_prototyping) , [инкрементное](https://en.wikipedia.org/wiki/Incremental_development) , а также синхронизация и стабилизация, но все основаны на идее того, что для обеспечения корректных действий в корректное время и по корректным причинам важно следовать каждому необходимому шагу. SDLC существует для помощи в сокращении времени вывода продукта на рынок, обеспечении более качественной производительности, экономии бюджета и повышения потенциальной пользы вашего продукта для заинтересованных сторон.

Итак, жизненный цикл — это период времени, который начинается с момента принятия решения о необходимости создания чего либо и заканчивается в момент полного изъятия этого «чего либо» из эксплуатации.

Here in my blog, I will cover topics that are important for my personal growth and professional path. One of the basic and paramount topics in computer science is the Systems Development Life Cycle (SDLC). Wikipedia defines it as the process of planning, building, testing, and deploying an information system. According to Taylor (2004), "the project life cycle covers all project activities, while the systems development life cycle focuses on the implementation of product requirements."

My own understanding of this term matches the following wording: a series of events that occur with the system in the process of its creation and further use, from the birth of an idea to the death of the universe. Why is it so long that it's endless? Because in real life there is no end to the development of the project - there are always either new wishes from the customer to expand the functionality, or support are required for users of the product.

Although the unit of measurement of the software development life cycle is infinity, it can be divided into six distinct phases. The first three steps attempt to provide answers to screening questions, while the last three are optimized to achieve actual results.

• **Requirements analysis** answers the question "What problems need solutions?" At this stage of the SDLC, you need to get feedback and support from relevant internal and external stakeholders.

• **Planning** answers the question "What do we want to do?" This provides an understanding of the project's unit economics, risk mitigation factors, and expected costs.

• **Architectural design** answers the question "How will we achieve our goals?" At this stage, it is important to consider operational excellence, safety, reliability, performance efficiency and cost optimization.

• **Software development** governs the process of creating a product. To meet the expectations, it is critical to follow the first three steps when starting development.

• **Testing** regulates the quality assurance of the product. Different types of testing are performed: unit, integration, interface testing and other ones depending on the purpose. The conclusion will be that the product has no errors and is ready for release.

• **Deployment and support** governs the use of the final product. The product can be implemented and integrated with third-party software. But the development does not end there, the project requires constant supervision and support.

Various SDLC methodologies have been created, such as waterfall, spiral, agile, rapid prototyping, incremental, and synchronization and stabilization, but all are based on the idea that in order to ensure the right actions at the right time and for the right reasons, it is important to follow every necessary step. The SDLC exists to help people reduce time to market, deliver better performance, save money, and increase the potential value of the product.

So, the life cycle is a period of time that begins from the idea step and ends when the project is completely taken out of service.