

ЖИЗНЕННЫЙ ЦИКЛ ИТ-ПРОДУКТА: ЭТАПЫ, РИСКИ И КОНТРОЛЬНЫЕ ТОЧКИ

У. В. Киселёва¹⁾, Е. С. Горбылёва²⁾

¹⁾ студент, Белорусский государственный университет, г. Минск, Беларусь,
@ulanakiseleva6gmail.com

²⁾ студент, Белорусский государственный университет, г. Минск, Беларусь,
v1688611@gmail.com

Научный руководитель Д. И. Муравицкий

преподаватель, Белорусский государственный университет, г. Минск, Беларусь, muravitskiDI@bsu.by

В статье рассматриваются ключевые этапы жизненного цикла разработки ПО (SDLC), анализируются риски и контрольные точки, повышающие эффективность разработки. Рассмотрены управление требованиями, проектирование, программирование, тестирование, внедрение и сопровождение ИТ-продуктов. Описаны модели SDLC: водопадная, Agile, спиральная и DevOps. Эффективное управление снижает затраты, оптимизирует сроки и повышает качество ПО в условиях быстро меняющейся цифровой среды.

Ключевые слова: жизненный цикл разработки; SDLC; управление рисками; контрольные точки; Agile; DevOps; тестирование ПО; внедрение программного обеспечения.

IT PRODUCT DEVELOPMENT LIFE CYCLE: PHASES, RISKS AND CONTROL MILESTONES

U. V. Kiseleva¹⁾, H. S. Harbyliova²⁾

¹⁾ student, Belarusian State University, Minsk, Belarus, *@ulanakiseleva6gmail.com*

²⁾ student, Belarusian State University, Minsk, Belarus, *v1688611@gmail.com*

Supervisor: D. I. Muravitsky

teacher, Belarusian State Medical University, Minsk, Belarus, muravitskiDI@bsu.by

The article examines key stages of the software development lifecycle (SDLC), analyzing risks and control points that enhance development efficiency. It covers requirements management, design, programming, testing, deployment, and maintenance of IT products. SDLC models such as Waterfall, Agile, Spiral, and DevOps are described. Effective management reduces costs, optimizes timelines, and improves software quality in a rapidly changing digital environment.

Keywords: software development life cycle; SDLC; risk management; control points; Agile; DevOps; software testing; software deployment.

Современная цифровая среда предъявляет высокие требования к качеству и надежности программного обеспечения, что обуславливает необходимость применения системного под-

хода к его разработке. Конкуренция на рынке ИТ-продуктов, постоянное изменение потребностей пользователей и стремительное развитие технологий делают процесс создания программных решений более сложным и многоэтапным. В этих условиях особое значение приобретает жизненный цикл разработки программного обеспечения (Software Development Life Cycle, SDLC), представляющий собой структурированный процесс, включающий последовательные стадии от зарождения идеи до эксплуатации и сопровождения продукта.

Целью настоящей статьи является рассмотрение ключевых этапов жизненного цикла ИТ-продукта, анализ возможных рисков, возникающих на каждом из них, а также определение контрольных точек, которые позволяют повысить предсказуемость и эффективность разработки. Актуальность исследования обусловлена тем, что грамотное управление жизненным циклом обеспечивает снижение затрат, оптимизацию временных ресурсов и повышение качества программного продукта, что особенно важно в условиях быстро меняющейся рыночной среды.

Жизненный цикл разработки традиционно начинается с инициации и анализа требований. На этом этапе формируется концепция будущего продукта, выявляются потребности заказчика и пользователей, а также разрабатывается техническое задание, которое служит основой для всей дальнейшей работы. Одним из главных рисков здесь является неполнота или противоречивость требований, что может привести к искажению конечных результатов. Контрольной точкой данного этапа выступает согласованный со всеми заинтересованными сторонами документ с требованиями [5].

Следующая стадия – проектирование системы, в ходе которой определяется архитектура продукта, создаются прототипы интерфейсов, выбираются технологии и формируется логика взаимодействия компонентов. Ошибки при выборе инструментов или чрезмерная сложность проектных решений могут привести к затруднениям на этапе реализации. В качестве контрольной точки выступает утверждённая архитектурная схема и набор прототипов, прошедших экспертную оценку [1].

Разработка представляет собой ключевой этап жизненного цикла, на котором осуществляется программирование, интеграция внешних сервисов и взаимодействие с базами данных. Здесь основными рисками являются ошибки в коде, нарушение сроков и снижение производительности. Контрольной точкой можно считать создание прототипа или рабочей версии программного обеспечения, успешно прошедшей юнит-тестирование.

После завершения программирования продукт проходит тестирование, которое включает функциональные, нагрузочные и регрессионные проверки. Данный этап направлен на выявление ошибок и подтверждение соответствия продукта ранее сформулированным требованиям. Недостаточный охват тестированием или выявление критических дефектов слишком поздно могут привести к значительным задержкам. Контрольная точка заключается в подготовке отчёта о тестировании и исправлении выявленных ошибок [2].

Завершение разработки сопровождается внедрением, в ходе которого программное обеспечение разворачивается в рабочей среде и становится доступным пользователям. Здесь могут возникнуть риски, связанные со сбоями при установке или неполной совместимостью с инфраструктурой заказчика. Успешный запуск продукта в продуктивной среде является подтверждением завершения данного этапа.

Финальной стадией является поддержка и сопровождение, включающие исправление ошибок, выпуск обновлений, внедрение новых функций и обеспечение безопасности. Отсутствие системной поддержки или несвоевременное устранение уязвимостей способны негативно сказаться на работе продукта и его восприятии пользователями. Контрольной точкой служат регулярные обновления и отчёты о стабильности системы, подтверждающие её долгосрочную работоспособность [4].

Жизненный цикл разработки ИТ-продукта

Этап SDLC	Основной результат	Контрольная точка	Основные риски
Инициация и анализ требований	Техническое задание (SRS)	Утверждённый документ с требованиями	Неполные или противоречивые требования
Проектирование системы	Архитектурные схемы, прототипы, структура данных	Экспертная оценка и согласование	Ошибки в выборе инструментов, чрезмерная сложность
Разработка (программирование)	Рабочая версия или прототип ИТ-продукта	Успешное прохождение юнит-тестирования	Ошибки кода, нарушение сроков, снижение производительности
Тестирование	Отчёты о функциональном, нагрузочном и регрессионном тестировании	Исправленные ошибки, подтверждение качества	Недостаточный охват тестами, критические дефекты
Внедрение (деплоймент)	Развёрнутое ПО в продуктивной среде	Успешный запуск и интеграция	Сбои при установке, несовместимость с инфраструктурой
Поддержка и сопровождение	Обновления, исправления, улучшения	Регулярные релизы и отчёты о стабильности	Отсутствие поддержки, безопасности

Важно отметить, что процесс разработки может строиться на основе различных моделей жизненного цикла. Линейная или водопадная модель предполагает последовательное выполнение этапов и подходит для проектов с фиксированными требованиями. Agile-модель ориентирована на итеративную разработку и эффективна при высокой изменчивости задач. Спиральная модель используется в крупных и рискованных проектах, где необходима поэтапная проверка решений. DevOps-подход обеспечивает непрерывное тестирование, интеграцию и развертывание, сокращая время выхода продукта на рынок [3].

Таким образом, жизненный цикл ИТ-продукта представляет собой не только последовательность этапов разработки, но и систему управления рисками, где контрольные точки играют роль ориентиров для оценки качества и готовности продукта. Последовательное прохождение стадий, выявление и минимизация рисков, а также выбор подходящей модели жизненного цикла позволяют значительно повысить качество программного обеспечения, обеспечить его соответствие требованиям пользователей и увеличить конкурентоспособность компании. В условиях цифровой трансформации и ускоренного технологического прогресса эффективное управление SDLC становится важнейшим условием успешного функционирования ИТ-организаций и экономики в целом.

Библиографические ссылки

1. Шаститко Д. В. Организация проектного обучения на курсе управления ИТ-проектами на основе стандарта РМВОК 6: проблемы и ограничения. 2020. ЭБ БГУ. Тема: Информатика; Народное образование. Педагогика.
2. Лауферман О. В., Лыгина Н. И. Разработка программного продукта: профессиональные стандарты, жизненный цикл, командная работа : учебное пособие. М. : Юрайт, 2022.
3. Product Life Cycle – Opportunities for Digital and Sustainable Transformation. URL: <https://www.mdpi.com/> (дата обращения: 19.09.2025).
4. Громов В. Ф. Информационные технологии : учебник. Тамбов : ТГТУ, 2021.
5. Чернышева А. М., Якубова Т. Н. Управление продуктом : учебник и практикум для вузов. М. : Юрайт, 2023.