

[Главная](#) > [Словарь](#) >

Каскадная модель разработки ПО

11 января 2024 обновлено: 24 июля 2025

[➔ Поделиться](#)

Каскадная модель разработки (или Waterfall, водопад) – это один из самых первых и простых подходов к жизненному циклу разработки программного обеспечения. Его суть заключается в том, что процесс разработки разбивается на несколько этапов, каждый из которых следует строго один за другим, без возврата на предыдущие стадии.

Другими словами, разработка идет как бы «каскадом», или «водопадом» сверху вниз, от одного этапа к другому:

Основные принципы модели

[Этапы каскадной модели разработки ПО](#)[Преимущества и недостатки каскадной модели](#)[Когда использовать каскадную модель](#)[Отличия от других моделей разработки](#)

Основные принципы

МОДЕЛИ

1. Переход с одного этапа разработки на другой происходит только после полного и успешного завершения предыдущего этапа. Например, нельзя начинать написание кода, не имея на руках полностью утвержденного проекта. Или приступить к тестированию, если не все модули еще реализованы;
2. Строгая проектная документация. Каждый этап разработки подробно описан в проектной документации. Каскадной модели присуще строгое следование намеченному плану;
3. Жесткая последовательность этапов. Каскадная модель запрещает пропуск этапов, либо возвращение к предыдущим этапам;
4. Любые изменения должны быть согласованы с заказчиком и задокументированы;
5. Тестирование и выявление ошибок происходит только после окончания разработки;
6. После утверждения ТЗ, заказчик не участвует в процессе создания продукта.

Этапы каскадной модели разработки ПО

Классическая модель разработки ПО подразумевает следующие этапы:

1. **Планирование (Planning)** На этом этапе разрабатывается детальный план проекта. Определяются сроки, бюджет, ресурсы и задачи. Этап включает в себя определение команды разработки, распределение обязанностей и создание графика работ.
2. **Анализ требований (Requirements Analysis)** В рамках этого этапа осуществляется сбор и документирование требований к программному продукту. Он включает в себя общение с заказчиком и конечными пользователями для понимания их потребностей. Результатом этого этапа является создание документа с требованиями.
3. **Проектирование и дизайн (Design)** В рамках этого этапа

разрабатывается архитектура программного продукта. Проектируются структура данных, интерфейсы пользователя, компоненты системы и алгоритмы. Этот этап также включает в себя создание дизайна пользовательского интерфейса (UI) и определение технических спецификаций.

4. **Разработка (Development)** На этом моменте начинается активное создание кода. Разработчики пишут программу в соответствии с требованиями и дизайном, определенными на предыдущих этапах. Разработка может включать в себя создание различных модулей, компонентов и функциональных частей программы.
5. **Тестирование и интеграция (Testing & Deployment)** В рамках этого этапа происходит проверка качества программного продукта с помощью различных видов тестирования, включая модульное, интеграционное, функциональное и другие виды. После тестирования ПО выпускается в продакшн. Оно устанавливается на целевой сервер или распространяется конечным пользователям.
6. **Поддержка (Support)** После выпуска ПО продолжает поддерживаться и обновляться. Этап может включать в себя внедрение обновлений, исправление возникающих ошибок, предоставление технической поддержки пользователям и реагирование на запросы на изменение или добавление необходимых функций.

SimpleOne SDLC

Сделайте шаг на пути совершенствования сервисных бизнес-процессов

Узнать больше

Преимущества и недостатки каскадной модели

+

Простота управления проектом.

Каскадная модель предоставляет четкий и легко управляемый процесс разработки. Фазы выполняются последовательно, что облегчает контроль за жизненным циклом проекта;

+

Прозрачность. Весь процесс разработки представлен в виде четких этапов, что делает проект прозрачным и для разработчиков, и для заказчиков;

+

Фиксированные требования. Требования к проекту четко определены на начальном этапе, что предотвращает изменения в процессе разработки;

+

Проектная документация. Создается подробная документация на каждой фазе разработки, что удобно для последующего сопровождения и поддержки программного продукта.

-

Сложность внесения изменений. В силу своей линейной структуры, каскадная модель может оказаться неэффективной при необходимости быстрого реагирования на изменения в требованиях после начала разработки;

-

Длительный цикл разработки. Поскольку каждая фаза выполняется последовательно, процесс может быть долгим, и заказчик не увидит результатов до завершения всего проекта;

-

Ограниченное взаимодействие с заказчиком. Клиент может видеть результаты только после завершения проекта, что уменьшает возможность для обратной связи и корректировок в реальном времени.

Когда использовать каскадную модель

Каскадная модель разработки постепенно утрачивает свою популярность, уступая место более гибким итеративным моделям. Несмотря на это, классический «водопадный» подход по-прежнему остается востребованным в некоторых сценариях.

Каскадная модель находит свое применение в тех случаях, когда заказчик имеет четкое видение результата и хочет избежать существенных изменений в процессе разработки, а также не стремится активно взаимодействовать с командой разработчиков. Кроме того, каскадная модель подходит для проектов, где заранее определены четкие сроки и конечные результаты каждого этапа.

Отличия от других моделей разработки

Помимо каскадной существует несколько альтернативных моделей разработки программного обеспечения:

- **Agile-модель (гибкая методология разработки)** Agile основана на итеративном и инкрементном подходах с акцентом на постоянном тесном взаимодействии с заказчиком, готовности к изменениям и частой поставке работающего ПО. Модель предполагает гибкое планирование и возможность оперативно реагировать на меняющиеся требования.
- **Спиральная модель (Spiral Model)** Предусматривает поэтапную разработку с обязательным анализом рисков на каждой итерации и возможностью возврата к предыдущим этапам. Отличается поддержкой итерационного подхода и фокусом на оценке рисков.
- **V-образная модель (V-Model)** Строгая последовательность этапов как в waterfall, но с акцентом на их верификацию и тестирование на соответствие выходным данным предыдущего этапа. Более формализованный подход по сравнению с каскадной моделью.

- RAD (быстрая разработка приложений)Основа — сокращение сроков разработки за счёт использования прототипов и интенсивного вовлечения заказчиков. Модель сфокусирована на быстром создании рабочих версий ПО.

Отличия каскадной модели:

1. **Линейный подход**Каскадная модель применяет линейный и последовательный метод разработки, где каждый этап жестко зависит от предыдущего. Это структурированное последовательное выполнение этапов отличает ее от других моделей.
2. **Ограниченная гибкость**В отличие от других моделей, каскадная не предусматривает гибкое реагирование на изменение требований в процессе разработки.
3. **Четкая структура**Одним из преимуществ каскадной модели является ее четкая структура, что делает ее легкой в управлении проектом, особенно в случаях, когда требования заранее четко определены.
4. **Ориентированность на конечный результат**В отличие от итеративных моделей, которые ориентированы на поэтапное улучшение, каскадная модель сосредоточена на полном завершении каждого этапа перед переходом к следующему.

 Поделиться

Бизнес-Решения	Платформа	Решения	Ресурсы
ITSM	Low-code платформа	Искусственный интеллект	Блог
ITAM	ESM	Автоматизация Service Desk	Экспертные материалы
B2B CRM	GenAI-платформа	Канбан-доска	Документация
SDLC	SimpleOne	Автоматизация ОЦО	Telegram-канал
HRMS	О компании	Маркетплейс	RUTUBE
VCSM	Истории успеха	Управление метриками и KPI	SimpleOne Experts
Партнеры	Новости	Управление временем и трудозатратами WTM	
Выбрать партнера	Сообщество		
Маркетплейс	Мероприятия		
Партнерская программа	Обучение и		

Технологическое
партнерство

Партнерский портал

сертификация

Карьера

Контакты

Общие вопросы

+7 495 181-85-08

hello@simpleone.ru

Отдел продаж

+7 495 181-85-08

sales@simpleone.ru

Техническая поддержка

+7 495 181-85-08

service@simpleone.ru

© 2025 SimpleOne

Политика конфиденциальности

Политика

Мы используем cookie для улучшения работы, анализа трафика и персонализации контента. Продолжая использовать сайт, вы соглашаетесь на обработку данных в соответствии с нашей Политикой конфиденциальности и соглашаетесь на использование файлов cookie

Хорошо

Отклонить