

Практическое занятие № 2

ЖИЗНЕННЫЙ ЦИКЛ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ: МОДЕЛИ И СТАДИИ

Цель занятия

1. Ознакомиться со стандартами в области обеспечения жизненного цикла программных средств.
2. Определить модели и стадии жизненного цикла программного обеспечения.

Основные теоретические сведения

В основе деятельности по созданию и использованию программных средств лежит понятие жизненного цикла, являющегося моделью создания и использования программного обеспечения, отражающей его различные состояния, начиная с момента возникновения необходимости в ПС и заканчивая моментом его полного выхода из употребления у пользователей.

Основными целями применения стандартов и нормативных документов в жизненном цикле ПО являются:

- снижение трудоемкости, длительности, стоимости и улучшение других технико-экономических показателей проектов ПС;
- повышение качества разрабатываемых и/или применяемых компонентов и ПС в целом при их приобретении, разработке, эксплуатации и сопровождении;
- предоставление возможности расширять и масштабировать в зависимости от размерности решаемых задач набор прикладных функций;
- обеспечение переносимости прикладных программ и данных между разными аппаратно-программными платформами.

Применение стандартов позволяет ориентироваться на построение систем из крупных функциональных узлов, отвечающих их требованиям, внедрять отработанные и проверенные проектные решения. Они определяют унифицированные интерфейсы и протоколы взаимодействия компонентов таким образом, что разработчику системы, как правило, не требуется вдаваться в детали внутреннего устройства этих компонентов.

В России жизненный цикл разработки ПО установлен стандартом ГОСТ 19.102 «Стадии разработки программ и программной документации» и содержит стадии и соответствующие им этапы, приведенные в табл. 1.1.

Таблица 1.1

Стадии и этапы разработки ПС

Стадии разработки	Этапы работ
Техническое задание	Обоснование необходимости разработки программы
	Научно-исследовательские работы
	Разработка и утверждение технического задания
Эскизный проект	Разработка эскизного проекта
	Утверждение эскизного проекта
Технический проект	Разработка технического проекта
	Утверждение технического проекта
Рабочий проект	Разработка программы и программной документации
	Испытания программы
Внедрение	Подготовка и передача программы

Следует также отметить существование жизненного цикла автоматизированных систем (АС). Так, действие ГОСТ 34.601 «Информационная технология. Автоматизированные системы. Стадии создания» распространяется на АС, используемые в различных видах деятельности (исследование, проектирование, управление и т. п.), включая их сочетания, создаваемые в организациях, объединениях и на предприятиях. Стандарт устанавливает стадии и этапы, а также содержание работ на каждом из них (табл. 1.2), выполнение которых необходимо и достаточно для создания АС, соответствующей заданным требованиям.

Таблица 1.2

Стадии и этапы создания АС

Стадии	Этапы работ
1	2
1.Формирование требований к АС	Обследование объекта и обоснование необходимости создания АС. Формирование требований пользователя АС. Оформление отчета о выполненной работе и заявки на разработку АС (тактико-технического задания)
2.Разработка концепции АС	Изучение объекта. Проведение необходимых научно-исследовательских работ. Разработка вариантов концепции АС и выбор варианта концепции АС, удовлетворяющего требованиям пользователя. Оформление отчета о выполненной работе
3. Техническое задание	Разработка и утверждение технического задания на создание АС
4. Эскизный проект	Разработка предварительных проектных решений по системе в целом и ее частям, а также документации на АС и ее части
5. Технический проект	Разработка проектных решений по системе и ее частям, документации на АС и ее части, заданий на проектирование в смежных частях проекта объекта автоматизации. Создание и оформление документации на поставку изделий для комплектования АС и/или технических требований (технических заданий) на их разработку.
6. Рабочая документация	Разработка рабочей документации на систему и ее части. Создание или адаптация программ

В зависимости от специфики создаваемых АС и условий их создания возможно:

- выполнение отдельных этапов работ до завершения предшествующих стадий;
- параллельное выполнение этапов работ, а также включение новых.

ГОСТ Р ИСО/МЭК 12207 «Информационная технология. Процессы жизненного цикла программных средств» наиболее полно на уровне международных стандартов отражает жизненный цикл, технологию разработки и обеспечения качества сложных ПС.

Жизненный цикл ПС представлен набором этапов, частных работ и операций в последовательности их выполнения и взаимосвязи, регламентирующих ведение разработки на всех стадиях от подготовки технического задания до завершения испытаний ряда версий и окончания эксплуатации ПС. В жизненный цикл включаются описания исходной информации, способов выполнения операций и работ, устанавливаются требования к результатам и правилам их контроля, а также к содержанию технологических и эксплуатационных документов. Определяется организационная структура коллективов, распределение и планирование работ, а также контроль за реализацией жизненного цикла ПС.

Стандарт может использоваться как непосредственный директивный, руководящий или рекомендательный документ, а также как организационная база при создании средств автоматизации соответствующих технологических этапов или процессов. Для реализации положений стандарта должны быть выбраны инструментальные средства, совместно образующие взаимосвязанный комплекс технологической поддержки и автоматизации жизненного цикла и не противоречащие предварительно скомпонованному

набору нормативных документов. Имеющиеся в стандарте пробелы следует заполнять спецификациями или нормативными документами, регламентирующими применение выбранных или созданных инструментальных средств автоматизации разработки и документирования ПС.

Под **моделью** жизненного цикла ПО понимается структура, определяющая последовательность выполнения и взаимосвязи процессов, действий, задач на протяжении жизненного цикла. Модель зависит от специфики, масштаба и сложности проекта и специфики условий, в которых система создается и функционирует.

Наибольшее распространение получили каскадная (рис. 1.1) и спиральная (рис. 1.2) модели жизненного цикла ПО. Согласно этим моделям проект реализуется в строго определенной последовательности: формирование требований к ПО, проектирование, реализация, тестирование, ввод в действие, эксплуатация и сопровождение, снятие с эксплуатации.

Принципиальной особенностью каскадного подхода является переход на следующую стадию, который осуществляется только после того, как будет полностью завершена работа на текущей стадии, и возвратов на пройденные стадии не предусматривается. Каждая стадия заканчивается получением некоторых результатов, которые служат в качестве исходных данных для следующей стадии.

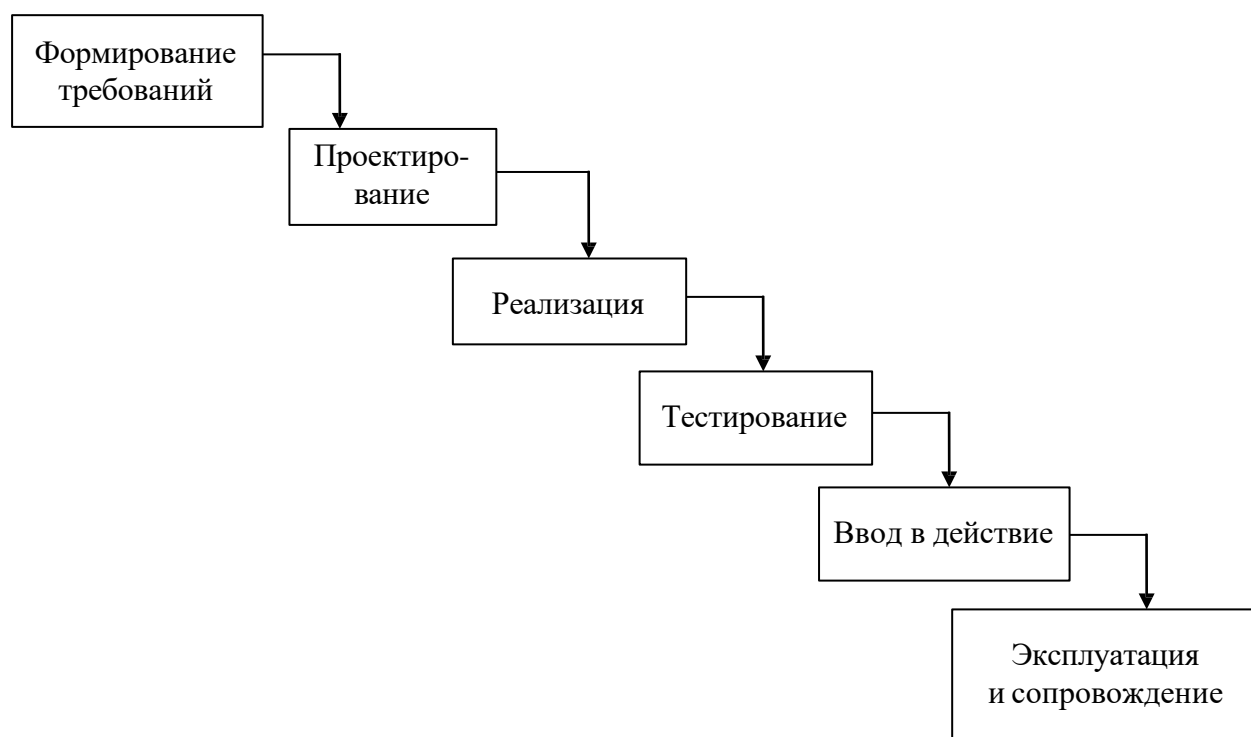


Рис. 1.1. Каскадная модель

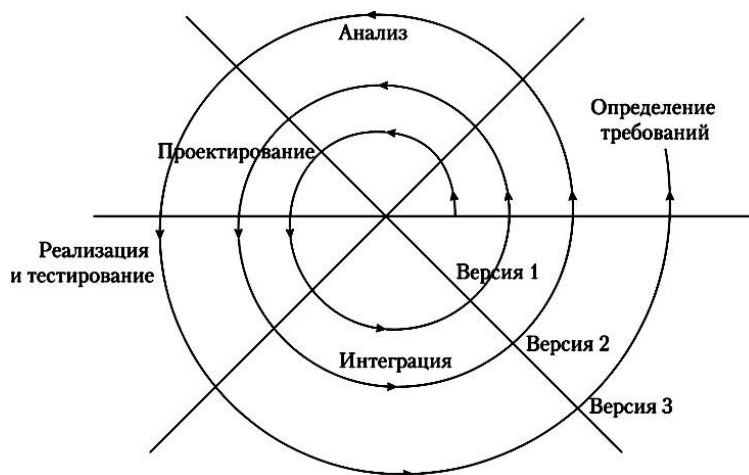


Рис. 1.2. Спиральная модель

Преимущества применения каскадного способа:

- на каждой стадии формируется законченный набор проектной документации, отвечающий требованиям полноты и согласованности;
- выполняемые в логической последовательности стадии работ позволяют планировать сроки их завершения и соответствующие затраты.

Каскадный подход хорошо зарекомендовал себя при построении систем, для которых в самом начале разработки можно достаточно точно и полно сформулировать все требования. В эту категорию попадают сложные расчетные системы, системы реального времени и другие подобные задачи.

В то же время этот подход обладает рядом недостатков, вызванных, прежде всего тем, что реальный процесс создания программного обеспечения никогда полностью не укладывается в такую жесткую схему. Процесс создания ПО носит, как правило, итерационный характер: результаты очередной стадии часто вызывают изменения в проектных решениях, выработанных на предыдущих стадиях. Таким образом, постоянно возникает потребность в возврате к предыдущим стадиям и уточнении или пересмотре ранее принятых решений.

Для преодоления перечисленных проблем была предложена спиральная модель, делающая упор на начальные этапы жизненного цикла: анализ и проектирование.

Каждый виток спирали соответствует созданию фрагмента или версии ПО, на нем уточняются цели проекта, определяется его качество и планируются работы следующего витка спирали. Углубляются и последовательно конкретизируются детали проекта, и в результате выбирается обоснованный вариант, который доводится до реализации.

Неполное завершение работ на каждом этапе позволяет переходить на следующий этап, не дожидаясь полного завершения работы на текущем. Основная проблема спирального цикла – определение момента перехода на следующий этап. Для ее решения необходимо ввести временные ограничения на каждый из этапов жизненного цикла.

Задание для самостоятельного выполнения

1. Выберите тему проекта (прил. А) и модель жизненного цикла разрабатываемой информационной системы.
2. Определите процессы для первой стадии «Формирование требований к ПО».
3. Оформите отчет.

Требования к оформлению отчета

Отчет оформляется обучающимся индивидуально и должен содержать:

- номер и тему практической работы;
- обоснование выбора модели жизненного цикла разрабатываемого ПО;
- описание процессов, характерных для первой стадии;
- ответы на контрольные вопросы.

Контрольные вопросы

1. Опишите особенности каскадной модели жизненного цикла ПС.
2. Опишите особенности спиральной модели жизненного цикла ПС.
3. Какая модель жизненного цикла ПС характерна для периода 1970–1985 гг.?
4. В чем состоит отличие спиральной модели жизненного цикла ПС от каскадной?
5. Перечислите этапы работ согласно ГОСТ 19.102 «Стадии разработки программ и программной документации».

1. Разработка информационной системы аэропорта.
2. Разработка информационной системы диспетчера такси.
3. Разработка информационной системы учета и анализа деятельности сети автомоек.
4. Разработка информационной системы больницы.
5. Разработка информационной системы детского сада.
6. Разработка информационной системы для учета посещаемости школы.
7. Разработка информационной системы для учета успеваемости школы.
8. Разработка информационной системы университета – учет заработной платы сотрудников.
9. Разработка информационной системы университета – учет посещаемости студентов.
10. Разработка информационной системы банка (различных направлений: ипотека, кредиты, вклады и др.).
11. Разработка информационной системы статистического центра по обработке звонков.
12. Разработка информационной системы налогового управления
13. Разработка мобильного приложения «...» для операционной системы «...».
14. Разработка электронного учебника по дисциплине «...».
15. Разработка обучающего приложения по дисциплине «...».
16. Разработка интерактивной системы для поддержки курса «...».
17. Разработка программного обеспечения для автоматизации работы отдела кадров.
18. Разработка корпоративного мессенджера организации.
19. Разработка программного обеспечения для сервиса по подбору специалистов.
20. Разработка программного обеспечения для автоматизации рабочего места отдела кадров предприятия.
21. Разработка единой базы знаний для внутреннего использования в организации.
22. Разработка корпоративной социальной сети организации.
23. Мобильное приложение для автоматизации какой-либо деятельности (бронирование заявок; составление отчетов, документов; доставки и др.).
24. Разработка веб-сервиса для размещения электронных объявлений.