



ЭК.04 ВВЕДЕНИЕ В СПЕЦИАЛЬНОСТЬ квалификация - программист

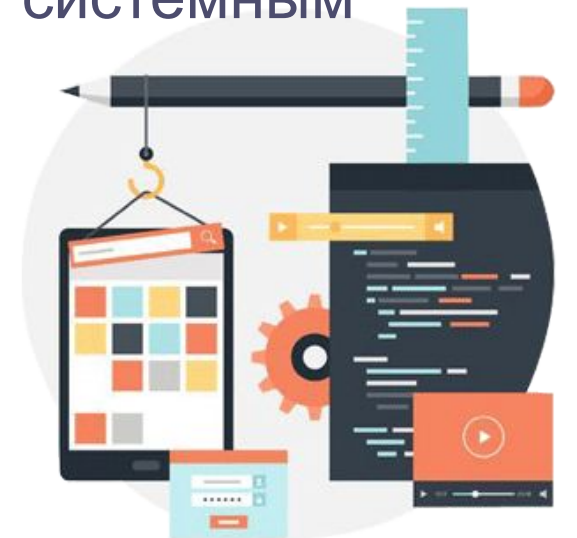
Тема занятия: Этапы разработки программного обеспечения. Жизненный цикл программы.

Цель занятия: Рассмотреть этапы разработки программы. Проанализировать виды жизненных циклов программы.

ЭТАПЫ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Программа – это комбинация компьютерных инструкций и данных, позволяющая аппаратному обеспечению вычислительной системы выполнять вычисления или функции управления.

Программный продукт – понимается программа, которую независимо от ее разработчиков можно использовать в предусмотренных целях на разных компьютерах, если только они удовлетворяют ее системным требованиям.



ЭТАПЫ РАЗРАБОТКИ
ПРОГРАММНОГО
ОБЕСПЕЧЕНИЯ

Программное
обеспечение

Системное

Прикладное

Инструментарий
программировани
я

СИСТЕМНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

ЭТАПЫ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Категория программного обеспечения, являющаяся необходимой принадлежностью компьютера, так как обеспечивает взаимодействие человека, всех устройств и программ компьютера.

Данный комплекс программ определяет на компьютере системную среду и правила работы в ней.

Операционная система обеспечивает:

1. Выполнение прикладных программ;
2. Управление ресурсами компьютера – памятью, процессором и всеми внешними устройствами;
3. Контакт человека с компьютером.

ПРИКЛАДНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

ЭТАПЫ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Все имеющиеся на компьютере прикладные программы составляют прикладное программное обеспечение.

Прикладные программы могут работать на компьютере только при условии, что на компьютере уже установлена операционная система.

Комплекс прикладных программ в среде операционной системы Windows называют приложением. Нередко его называют также пакетом прикладных программ.

ПРИКЛАДНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

ЭТАПЫ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Группы прикладного программного обеспечения:

1. Текстовые процессоры – для создания текстовых документов;
2. Табличные процессоры (электронные таблицы) – для вычислений и анализа информации, представленной в табличной форме;
3. Базы данных – для организации и управления данными;
4. Графические пакеты – для представления информации в виде рисунков и графиков.
5. Коммуникационные программы – для обмена информацией между компьютерами;
6. Интегрированные пакеты, включающие несколько прикладных программ разного назначения;
7. Обучающие программы, электронные учебники, словари, энциклопедии;
8. Игры.

ИНСТРУМЕНТАРИЙ ПРОГРАММИРОВАНИЯ

ЭТАПЫ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Эта категория предназначена для создания системного и прикладного программного обеспечения.

Методы работы с инструментарием программирования определяются той средой, в которой осуществляется преобразование алгоритма в программу для компьютера.

ЭТАПЫ РАЗРАБОТКИ
ПРОГРАММНОГО
ОБЕСПЕЧЕНИЯ

ЭТАПЫ РАЗРАБОТКИ

АНАЛИЗ ТРЕБОВАНИЙ

ПРОЕКТИРОВАНИЕ

КОДИРОВАНИЕ

ВНЕДРЕНИЕ

АНАЛИЗ ТРЕБОВАНИЙ

В рамках этой стадии происходит максимально эффективное взаимодействие нуждающегося в программном решении клиента и сотрудников компании-разработчика, в ходе обсуждения деталей проекта помогающих более четко сформулировать предъявляемые к ПО требования.

Результатом проведенного анализа становится формирование основного регламента, на который будет опираться исполнитель в своей работе – технического задания на разработку программного обеспечения.

Техническое задание должно полностью описывать поставленные перед разработчиком задачи и охарактеризовать конечную цель проекта в понимании заказчика.

ПРОЕКТИРОВАНИЕ

ЭТАПЫ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Ключевой этап – стадия моделирования теоретической основы будущего продукта.

Современные средства программирования позволяют частично объединить этапы проектирования и кодирования, то есть технической реализации продукта, будучи основанными на объектно-ориентированном подходе.

Одной из составных частей этапа проектирования является выбор инструментальных средств операционной системы.

ПРОЕКТИРОВАНИЕ

В рамках данного этапа необходимо осуществить:

1. Оценку результатов проведенного первоначального анализа и выявленных ограничений;
2. Поиск критических участков проекта;
3. Формирование окончательной архитектуры создаваемой системы;
4. Анализ необходимости использования программных модулей или готовых решений сторонних разработчиков;
5. Проектирование основных элементов продукта – модели базы данных, процессов и кода;
6. Выбор среды программирования и инструментов разработки, утверждение интерфейса программы, включая элементы графического отображения данных.
7. Определение основных требований к безопасности разрабатываемого ПО

КОДИРОВАНИЕ

Опираясь на выбранный в процессе подготовки язык программирования следующим этапом будет кодирование.

Кодирование может происходить параллельно со следующим этапом разработки – тестированием ПО.

Уровень и эффективность взаимодействия всех элементов, задействованных для выполнения сформулированных задач компанией разработчиком, на текущем этапе является самым важным – от слаженности действий программистов, тестировщиков и проектировщиков зависит качество реализации проекта.

ТЕСТИРОВАНИЕ И ОТЛАДКА

После достижения задуманного программистами в написанном коде следуют не менее важные этапы разработки программного обеспечения, зачастую объединяемые в одну фазу — тестирование продукта и последующая отладка, позволяющая ликвидировать огрехи программирования и добиться конечной цели — полнофункциональной работы разработанной программы.

Процесс тестирования позволяет смоделировать ситуации, при которых программный продукт перестает функционировать. Отдел отладки затем локализует и исправляет обнаруженные ошибки кода.

Эти два этапа занимают не меньше 30% затрачиваемого на весь проект времени, так как от их качественного исполнения зависит судьба созданного силами программистов программного обеспечения

ВНЕДРЕНИЕ

Процедура внедрения программного обеспечения в эксплуатацию является завершающей стадией разработки.

Как правило, ввод в эксплуатацию ПО осуществляется в три этапа:

1. Первоначальная загрузка данных;
2. Постепенное накопление информации;
3. Вывод созданного ПО на проектную мощность.

Ключевой целью поэтапного внедрения разработанной программы становится постепенное выявление не обнаруженных ранее ошибок и недочетов кода.

ЖИЗНЕННЫЙ ЦИКЛ ПО

КАСКАДНАЯ МОДЕЛЬ

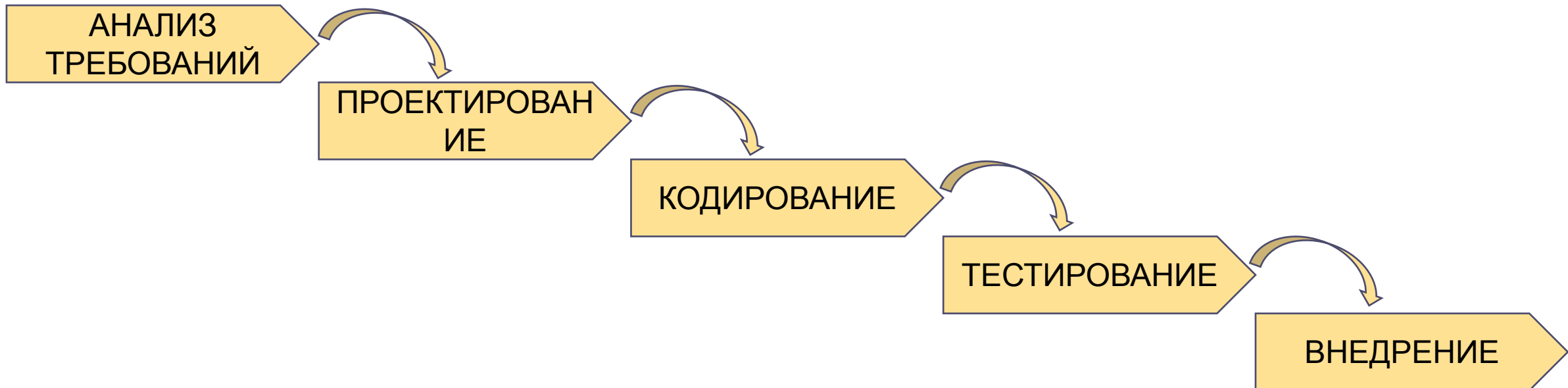
ИТЕРАЦИОННАЯ МОДЕЛЬ

СПИРАЛЬНАЯ МОДЕЛЬ

ИНКРЕМЕНТНАЯ МОДЕЛЬ

КАСКАДНАЯ МОДЕЛЬ (WATERFALL)

Каскадная модель (однократный проход, водопадная или классическая модель) подразумевает линейную последовательность выполнения стадий создания информационной системы. Другими словами, переход с одной стадии на следующую происходит только после того, как будет полностью завершена работа на текущей.



КАСКАДНАЯ МОДЕЛЬ (WATERFALL)

Данная модель применяется при разработке информационных систем, для которых в самом начале разработки можно достаточно точно и полно сформулировать все требования.

ДОСТОИНСТВА

1. На каждой стадии формируется законченный набор документации
2. Выполняемые в четкой последовательности стадии позволяют уверенно планировать сроки выполнения работ

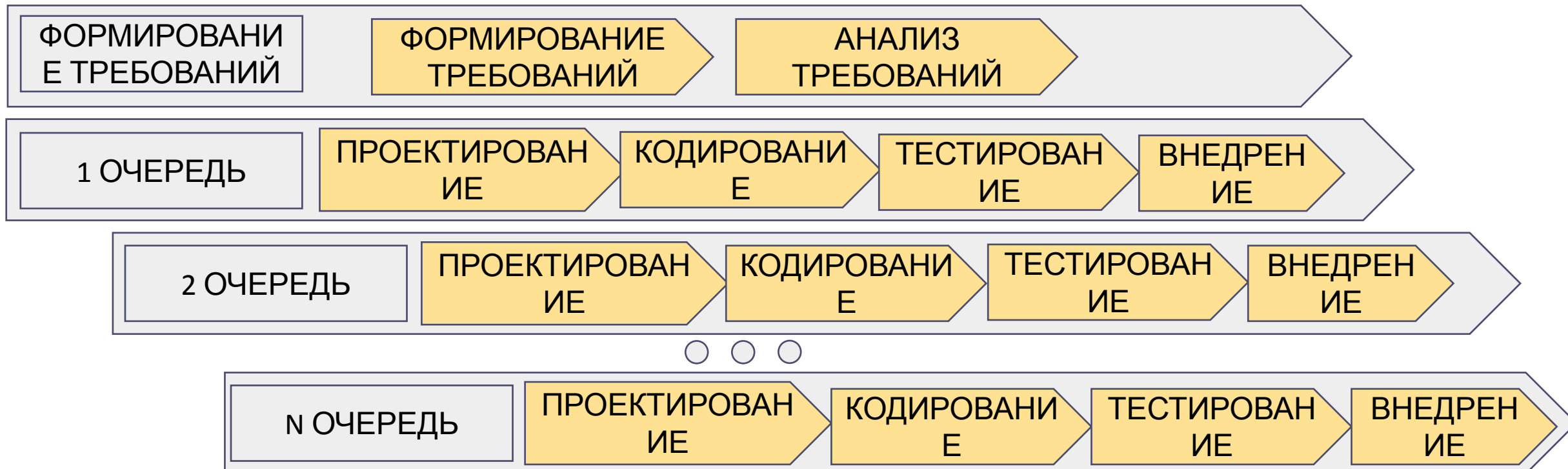
НЕДОСТАТКИ

1. Реальный процесс разработки ИС редко укладывается в такую жесткую схему.
2. Основана на точной формулировке исходных требований к ИС.
3. Результаты разработки доступны заказчику только в конце проекта

ЭТАПЫ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

ИНКРЕМЕНТНАЯ МОДЕЛЬ

Инкрементная стратегия (англ. increment – увеличение, приращение) подразумевает разработку информационной системы с линейной последовательностью стадий, но в несколько инкрементов (версий), т. е. с запланированным улучшением продукта.



ИНКРЕМЕНТНАЯ МОДЕЛЬ

ЭТАПЫ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

В начале работы над проектом определяются все основные требования к системе, после чего выполняется ее разработка в виде последовательности версий.

При этом каждая версия является законченным и работоспособным продуктом.

Первая версия реализует часть запланированных возможностей, следующая версия реализует дополнительные возможности и т. д., пока не будет получена полная система.

Достоинства и недостатки этой стратегии такие же, но в отличие от классической стратегии заказчик может раньше увидеть результаты. Уже по результатам разработки и внедрения первой версии он может

СПИРАЛЬНАЯ МОДЕЛЬ

Спиральная стратегия (эволюционная или итерационная модель, автор Барри Боэм, 1986-88 гг.) подразумевает разработку в виде последовательности версий, но в начале проекта определены не все требования. Требования уточняются в результате разработки версий.



СПИРАЛЬНАЯ МОДЕЛЬ

ЭТАПЫ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Достоинства модели:

1. Позволяет быстрее показать пользователям системы работоспособный продукт;
2. Допускает изменение требований при разработке информационной системы;
3. Обеспечивает большую гибкость в управлении проектом;
4. Позволяет получить более надежную и устойчивую систему.
5. Позволяет совершенствовать процесс разработки
6. Анализ, проводимый в каждой итерации, позволяет проводить оценку того, что должно быть изменено в организации разработки;
7. Уменьшаются риски заказчика.

Недостатки модели:

1. Увеличивается неопределенность у разработчика в перспективах развития проекта

Характеристика проекта	Модель (стратегия)		
	Каскадная	Инкрементная	Спиральная
Новизна разработки и обеспеченность ресурсами	Типовой. Хорошо проработаны технология и методы решения задачи		Нетиповой (новаторский). Нетрадиционный для разработчика
	Ресурсов заказчика и разработчика хватает для реализации проекта в сжатые сроки	Ресурсов заказчика или разработчика не хватает для реализации проекта в сжатые сроки	
Масштаб проекта	Малые и средние проекты	Средние и крупные проекты	Любые проекты
Сроки выполнения проекта	До года	До нескольких лет. Разработка одной версии может занимать срок от нескольких недель до года	
Заключение отдельных договоров на отдельные версии	Заключается один договор. Версия и есть итоговый результат проекта	На отдельную версию или несколько последовательных версий обычно заключается отдельный договор	
Определение основных требований в начале проекта	Да	Да	Нет
Изменение требований по мере развития проекта	Нет	Незначительное	Да
Разработка итерациями (версиями)	Нет	Да	Да
Распространение промежуточного ПО	Нет	Может быть	Да