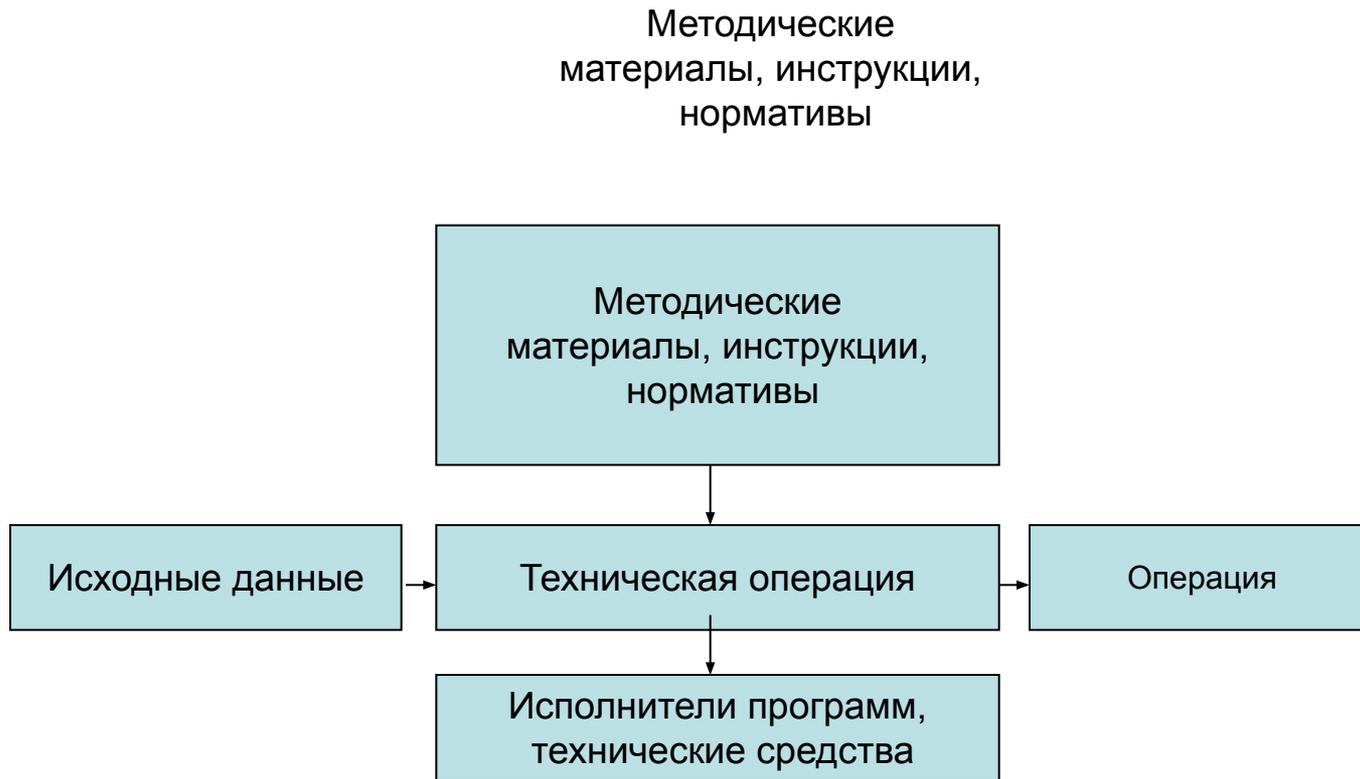


Создание программной системы – трудоёмкая задача, в которой объём программного обеспечения превышает сотни тысяч операторов. Специалист в области разработки программного обеспечения должен иметь представление о методах анализа проектирования, реализации и тестирования программных систем, а также ориентироваться в современных подходах и технологиях.

Технология программирования называют совокупность методов и средств используемых в процессе разработке программного обеспечения. Как и любая другая технология программирования представляет собой набор технологических инструкций, включающих:

1. указания последовательности выполнения технологических операций
2. перечисление условий, при которых выполняется та или иная операция
3. описание самих операций, где для каждой операции определяются исходные данные, результаты, нормативы и стандарты, критерии и методы оценки



Кроме набора операций и их последовательности технология определяет способ описания проектирования системы на каждом этапе разработки системы. Бывают технологии которые решают конкретную отдельную задачу, и технологии которые охватывают весь процесс разработки.₂

Основные этапы развития технологии программирования

1й этап: Стихийное программирование от начала 60х

Программы имели простейшую структуру. Программа состояла из глобальной программы и подпрограмм которые к ней обращались. При увеличении подпрограмм возрастала вероятность увеличения искажения глобальных данных. Для предотвращения эти ошибок появилось такое понятие как локальные данные. Такой подход создал кризис программирования. Он заключался в следующем:

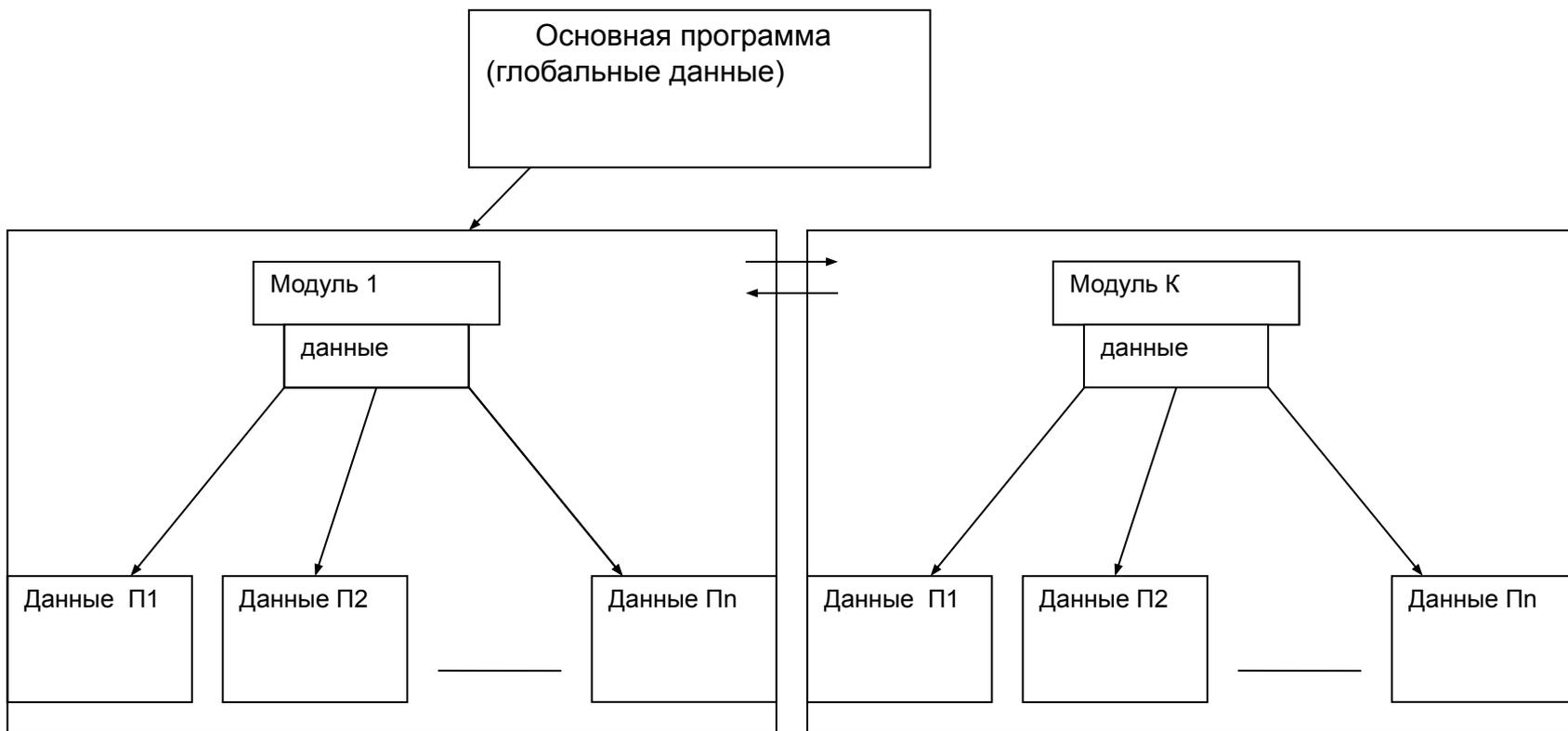
- 1) не соблюдались сроки создания
- 2) проект устаревал раньше, чем был готов
- 3) увеличивалась его стоимость
- 4) многие проекты так и не были запущены!

2й этап: Структурный подход к программированию 60-70гг.

Он представляет собой совокупность технологических приёмов, охватывающих выполнение всех этапов разработки ПО. В основе лежит декомпозиция сложных систем с целью последующей реализации в виде отдельных небольших подпрограмм, порядка 50-100 операторов. Проектирование строилось сверху вниз и обеспечивало проработку и реализацию общей идеи построения программы. В подход заложены языки программирования СИ и Паскаль. Но этого оказалось не достаточным для разработки сложных программ, и возникла потребность в новом структурировании данных, в связи с чем в языках программирования появляется возможность определять новые типы данных. В результате чего стала развиваться новая модульная технология программирования.

Модульное программирование предполагает выделение 2х программ, использующих одни и те же глобальные данные в отдельно компилируемые модули (библиотеки) графические модули, модуль. Связи между модулями в этой технологии осуществляется через специальный интерфейс. При этом доступ к реализации в модули запрещён. С++, VB, современный Паскаль. Использование модульного программирования существенно упростило разработку ПО несколькими программистами.

Основная программа (глобальные данные)



3й этап: Объектный подход к программированию 80-90г

Технология создания сложного ПО, основанная на представлении программы виде совокупности объектов, каждый из которых является экземпляром определённого типа (класса). Классы образуют иерархию с наследованием свойств. Взаимодействие объектов в таких системах осуществляется путём передачи сообщений. Основным свойством ООП является более естественная декомпозиция ПО. Это приводит к полной локализации данных, позволяющей вести независимую разработку отдельных частей ПО. ООП предлагает новый способ локализации программ, основанный на механизмах наследования, полиморфизма и композиции. Это позволяет создавать библиотеки классов для различных применений. Развитие технологий программирования основанных на ООП позволило создать визуальное программирование. Недостатки:

- Нет стандартов компоновки объектов в единое целое
- Изменение реализации одного из программных объектов связано с перекомпоновкой всего ПО.

С середины 90х годов появился новый этап:

Компонентный подход и CASE – технологии.

Подход предполагает построение ПО из отдельных компонентов, физически отдельно существующих частей ПО, которые взаимодействуют через интерфейсы. В отличие от обычных объектов объекты компоненты можно собрать в динамически вызываемые библиотеки, или в исполняемый файл. И дальше использовать их в любой программе, поддерживающей данную технологию.

КП лежит в основе технологий разработанный на базе COM (component object model) и технологии распределённых систем COBRA. Технология COM определяет общие правила взаимодействия программ любых типов: библиотек, приложений, ОС. и позволяет одной части ПО использовать функции другой в пределах одного процесса. COBRA можно использовать для создания определённого ПО в разнородной вычислительной среде. Отличительной особенностью современного этапа развития технологий является создание и внедрение автоматизированных технологий, которые названы CASE технологиями, поддерживающими объектный и структурный подходы. Сейчас получил развитие унифицированный язык моделирования UML.

Проблемы разработки сложных программных систем

Сложные ПО:

- Сложность формального определения требования к программным системам
- Коллективная разработка
- Необходимость увеличения степени повторяемости кодов
- отсутствие удовлетворительных средств формального описания поведения дискретных систем.

Проблемы:

- При определении требований необходимо учесть большое количество различных факторов.
- Разработчики не являются специалистами предметной области и не могут верно сформулировать проблему.

- В процессе создания программных систем используют языки сравнительно низкого уровня, это приводит к крайней детализации операций и увеличивает объём описаний разрабатываемых программ. Средств позволяющих детально описывать поведение сложных систем на более высоком уровне, чем язык программирования не существует
- Коллективная разработка
- Необходимость увеличения и степень кодов. Для повышения производительности труда компании стремятся к созданию библиотек компонентов которые можно использовать в дальнейшей разработки. При этом компоненты стараются делать более универсальные, что в итоге приводит к увеличению сложности разработки.

Жизненные циклы разработки ПО

Жизненным циклом ПО называют период с момента появления идеи ПО до момента завершения его поддержки фирмой разработчиком. Состав процессов жизненного цикла регламентируется международным стандартом ISO/IEC 12207.

ISO – международная организации по стандартизации

IEC – международная комиссия по электротехнике. Этот стандарт описывает структуру жизненного цикла ПО и его процессов. Процесс жизненного цикла определяется как совокупность взаимосвязанных действий преобразующие некоторые входные данные в выходные. Каждый процесс характеризуется отдельными задачами и методами их решения, а так же исходными данными.

Основные процессы: приобретение, поставка, разработка, эксплуатация, сопровождение.

Организационные процессы: Управление, усовершенствование, создание инфраструктуры, обучение.

Вспомогательные процессы: документирование, управление конфигурацией, обеспечение качества, аттестация, аудит, разрешение проблем. ¹¹

Процесс разработки предусматривает работу, выполняемую разработчиком ПО в соответствии с заданными требованиями, включая оформление всей документации, а также материалов проверки работоспособности и качества ПО. По стандартам проверка включает следующие действия:

1. подготовительная работа – план работы, определяются стандарты и методы разработки
2. анализ требований к ПО – пользовательские требования, требования по безопасности
3. проектирование архитектуры ПО
4. детальное проектирование ПО – определяются интерфейсы, разработка и тестирование
5. кодирование и тестирование ПО
6. интеграция ПО – все компоненты собираются в целое
7. тестирование ПО
8. квалификационное тестирование ПО – оформление полноты документации
9. Установка ПО на оборудование заказчика
10. Приём ПО (подписание договоров)

- **Постановка задачи:** в процессе постановки задачи чётко формулируется назначение ПО и определяются основные требования к нему. Каждое требование представляет собой описание необходимого или желаемого свойства ПО. Функциональные требования определяют функции, которые должны выполнять разрабатываемое ПО. Эксплуатационные требования определяют особенности его функционирования. Этап постановки задачи заканчивается постановкой ТЗ, фиксирующим все потенциальные требования к ПО и принимаются основные проблемы мышления.
- **Анализ требований и определение спецификаций.** Спецификации называют точное формализованное описание функций и ограничений разрабатываемого ПО.
Различают: функциональные и эксплуатационные.
Для получения спецификаций выполняют анализ технического задания, выбирают математический аппарат формализации, строят модель предметной области, определяют подзадачи и разрабатывают методы их решения.

- **Проектирование** – стадия технического проекта. Проектирование включает в себя:
 - проектирование общей структуры
 - декомпозицию компонентов
 - проектирование этих компонентов

Результатом проектирования является детальная модель разрабатываемого программного обеспечения вместе со спецификациями его компонентов всех уровней.

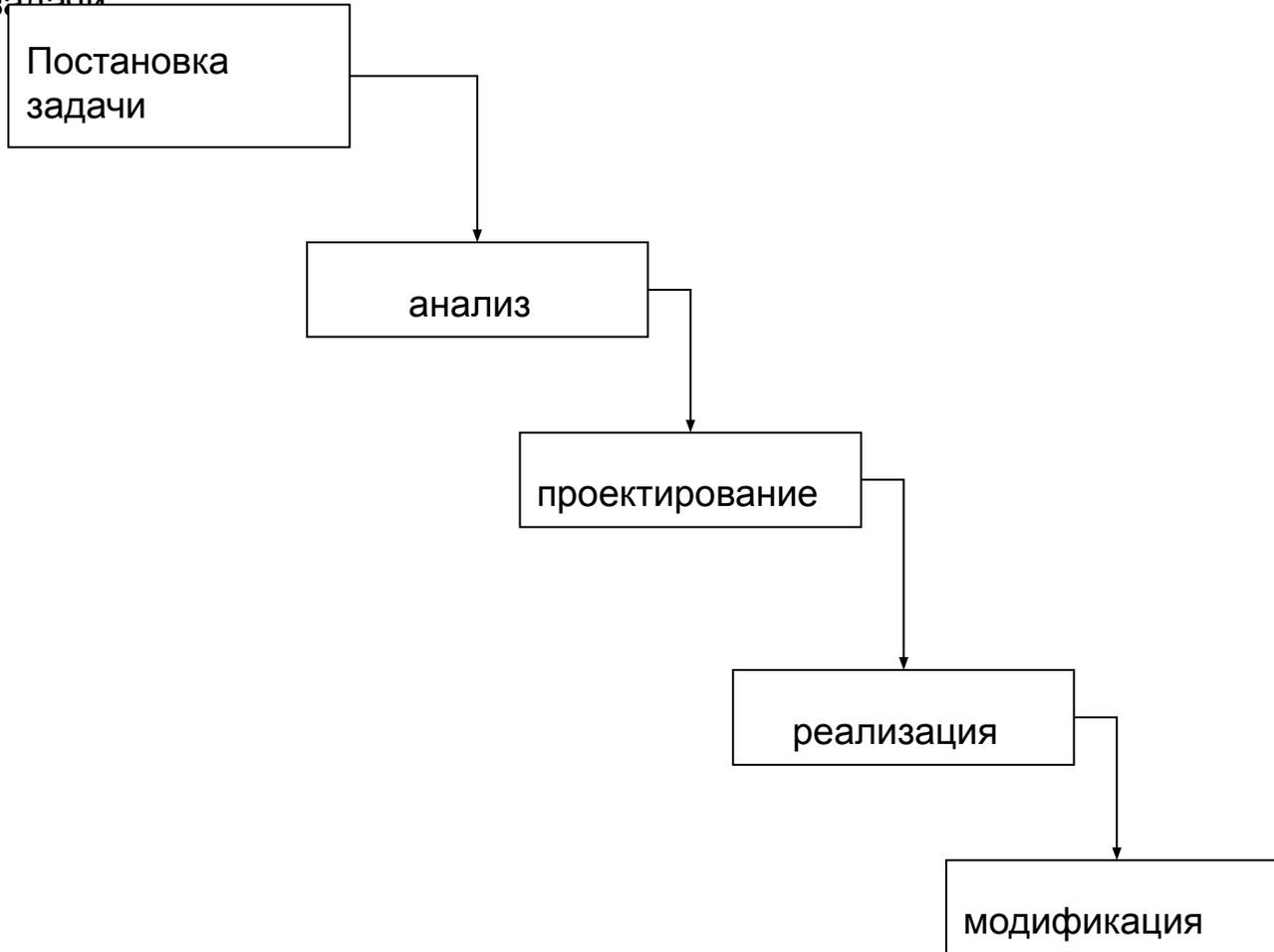
- **Реализация** – стадия рабочего проекта – процесс поэтапного написания кода программы на выбранном языке программирования их тестирования и отладку.

Существует 5 этап – сопровождение (сейчас не входит) - создание и внедрение новых версий.

Для всех этапов разработки существует 3 модели жизненного цикла программного обеспечения.

• 1. Каскадная.

Постановка задачи



Такая модель предполагает переход на следующую стадию только после полного завершения предыдущей стадии.

Достоинства:

- получение в конце каждой стадии законченного набора проектирования документации
- простота планирования процесса разработки.

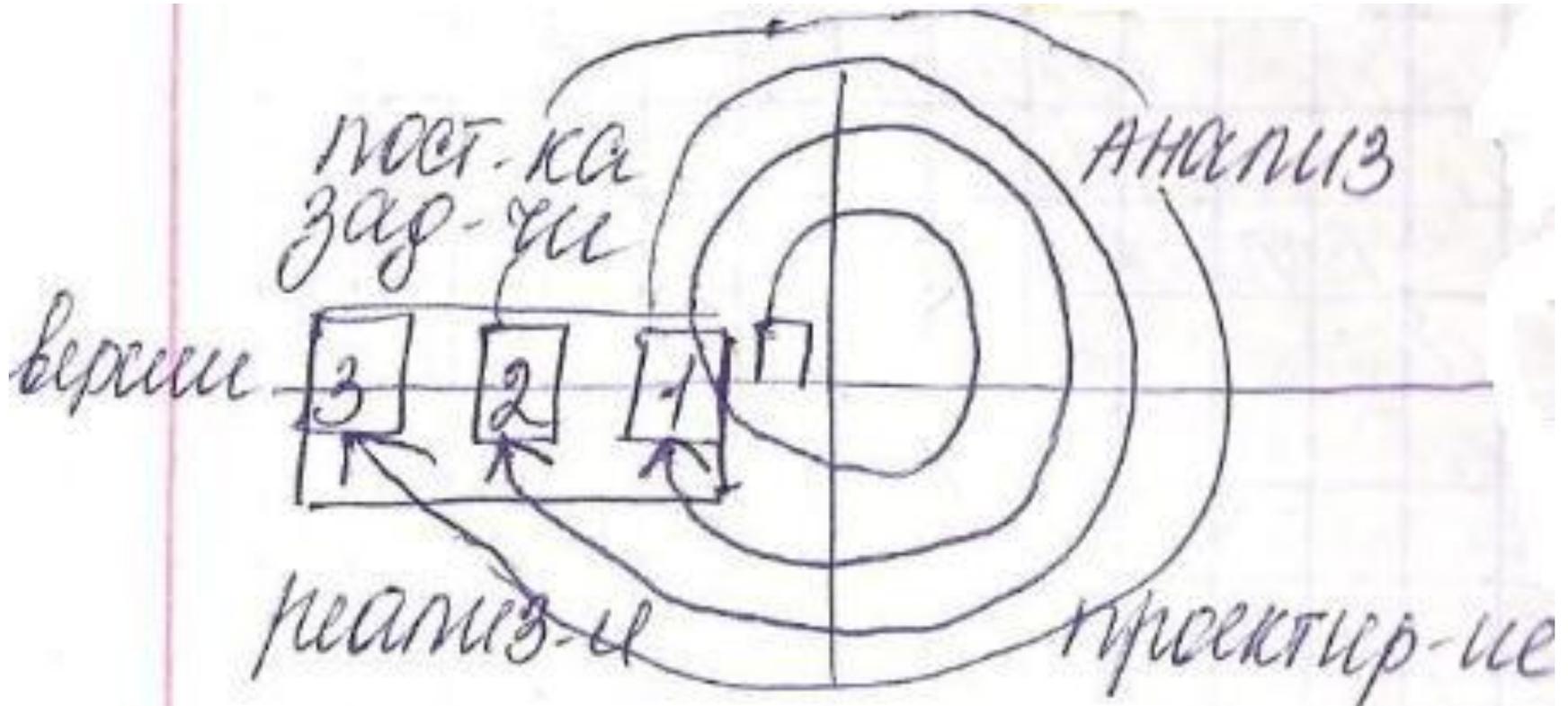
Недостатки:

- при неточных спецификациях приводят к пересмотру уже принятых решений.
- Изменение требования заказа непосредственно в процессе разработки.

2. Модель с промежуточным контролем.

В данной схеме можно вернуться на любой уровень.

- 3. Спиральная модель.



В соответствии с этой схемой ПО создается не сразу, а итерационно с использованием метода прототипирования. Прототипом называют реальный программный продукт, реализовывающий отдельные функции и внешние интерфейсы разрабатываемого ПО. На итерации как правило специфицируют, проектируют, реализуют и тестируют интерфейс пользователя. На 2) добавляют некоторый ограниченный набор функций, на последних этапах этот набор расширяет возможности разрабатываемого ПО. Основное достоинство - начало с некоторой итерации, на которой обеспечена определяемая функциональная полнота, и продукт можно предложить пользователю, что позволяет:

- сократить время до появления новых версий ПО
- заинтересовать большое количество пользователей за счет продвижения следующих версий.
- ускорить формирование и уточнение спецификаций за счет практики использования ПО
- уменьшить вероятность морального устарения ПО за время разработки.

Разработка спиральной модели жизненного цикла программы и КЕЙС технологии позволили сформулировать условия выполнения которых сокращает сроки создания ПО. Современные технологии проектирования, разработки и сопровождения ПО должна отвечать следующим требованиям:

1. Поддержка полного жизненного цикла ПО
2. Гарантированное достижения цели разработки с заданным качеством и в установленное время.
3. Возможность выполнения крупных проектов виде подсистем, разрабатываемые группой производителей (3-7 человек) с последующей интеграцией составных частей и координации ведения общего проекта.
4. Минимальное время получения работоспособной системы
5. Возможность управления конфигурации проекта, ведения версий проекта и автоматического корректного выпуска документации по каждой версии.
6. Независимость выполняемых проектных решений от средств реализации.

Этим требованиям отвечает технология RAD (быстрая разработка приложений). RAD создана для достижения высокой скорости, высокого качества и низкой стоимости разработки ПО. В этой технологии процесс делится на этапы:

1. Анализ и планирование требования пользователя
2. Проектирование
3. Реализация
4. Внедрение

На 1-м этапе определяются приоритетные требования и направления ограничивающие масштаб проекта. На 2-м этапе детально описываются процессы ПО, требования к данным и доступа к ним и определяют состав необходимой документации. По результатам анализа этих процессов определяют количество функциональных точек и принимают решение о количестве подсистем и команд участвующих в разработке. Под функциональной точкой в понимают любой из функциональных элементов ПО, к которым относится: входной элемент приложения, выходной элемент приложения, запрос, совокупность записей, интерфейс приложения.

Нормы определяющие количество человек для разработки: менее 1000 функциональных точек – 1 человек, от 1-4000 точек – команда (3-7 человек). На каждые 4000 точек – 1 команда.

На этапе реализации выполняется интерактивное построение реальной системы. На этом этапе активно работают с заказчиками.

На этапе внедрения проводят обучение пользователей и осуществляя постепенный переход на новую систему. При этом эксплуатация старой версии продолжается до полного внедрения.

Оценка качества процессов создания ПО.

В текущий период рынок ПО характеризуется переходом от штучного производства ПО к промышленному.

Соответственно возрастает требования к качеству что требует совершенствование процессов разработки. На настоящий момент существует несколько стандартов связанных с оценкой качества этих процессов которые обеспечивает организацию разработок, к наиболее известным относятся:

- 1) **ISO-9000** к разработке ПО ISO-9000-9004 –в этой серии сформированы необычные условия для достижения некоторого минимального уровня организации процессов.
- 2) **LMM**- представляет собой совокупность критериев оценки зрелости организации разработки и рецепты улучшения существующих процессов.

Пять уровней технологической зрелости

Разработчики модели LMM (SEI) определили пять уровней технологической зрелости, по которым заказчики могут оценивать потенциальных поставщиков (претендентов на получение контракта), а поставщики могут совершенствовать процессы создания ПО. Каждому из показанных на рисунке уровней технологической зрелости внутри модели CMM дано следующее краткое определение:

- Уровень 1: Начальный (Initial) — технология разработки ПО характеризуется как произвольная (импровизированная), в некоторых случаях — даже хаотическая. Лишь некоторые процессы определены, успех всецело зависит от усилий отдельных сотрудников.
- Уровень 2: Повторяемый (Repeatable) — базовые процессы управления проектом ПО установлены для отслеживания стоимости, графика и функциональности выходного продукта. Необходимая дисциплина соблюдения установленных процессов имеет место и обеспечивает возможность повторения успеха предыдущих проектов в той же прикладной области.

- Уровень 3: Определенный (Defined) — управленческие и инженерные процессы задокументированы, стандартизованы и интегрированы в унифицированную для всей организации технологию создания ПО. Каждый проект использует утвержденную, адаптированную к особенностям данного проекта, версию этой технологии.
- Уровень 4: Управляемый (Managed) — детальные метрики (объективные данные) о качестве исполнения процессов и выходной продукции собираются и накапливаются. Управление процессами и выходной продукцией осуществляется по количественным оценкам.
- Уровень 5: Оптимизируемый (Optimized) — совершенствование технологии создания ПО осуществляется непрерывно на основе количественной обратной связи от процессов и пилотного внедрения инновационных идей. На этом уровне идёт постоянное улучшение существующих процессов, т.е. постоянно вводятся новые технологии для эффективности разработки ПО.

3) **стандарт ISO/IEC 15504 (SPACE)** – он определяет возможности для улучшения процесса создания ПО. В основе стандарта лежит оценка процесса. Оценка выполняется путём сравнения процесса разработки ПО в данной организации с описанной в стандарте модели. Этот анализ помогает определить сильные и слабые стороны процесса, а также внутренние риски присущие данному процессу. Это помогает оценить эффективность процессов, определить причины ухудшения качества и связанные с этим издержки во времени и стоимости. В основе стандарта так же лежит определения возможности процесса, т.е. возможности его улучшения.