Определение технологии конструирования программного обеспечения

- Технология конструирования программного обеспечения (ТКПО) — система инженерных принципов для создания экономичного ПО, которое надежно и эффективно работает в реальных компьютерах

Различают методы, средства и процедуры ТКПО.

Методы обеспечивают решение следующих задач:

- планирование и оценка проекта;
- анализ системных и программных требований;
- проектирование алгоритмов, структур данных и программных структур;
- кодирование;
- тестирование;
- сопровождение.

Средства (утилиты) ТКПО

- обеспечивают автоматизированную или автоматическую поддержку методов. В целях совместного применения утилиты могут объединяться в системы автоматизированного конструирования ПО. Такие системы принято называть CASE-системами. Аббревиатура CASE расшифровывается как Computer Aided Software Engineering (программная инженерия с компьютерной поддержкой).

Процедуры ТКПО

- Процедуры являются связующим звеном, которое соединяет методы и средства так, что они обеспечивают непрерывную технологическую цепочку разработки.

Процедуры определяют:

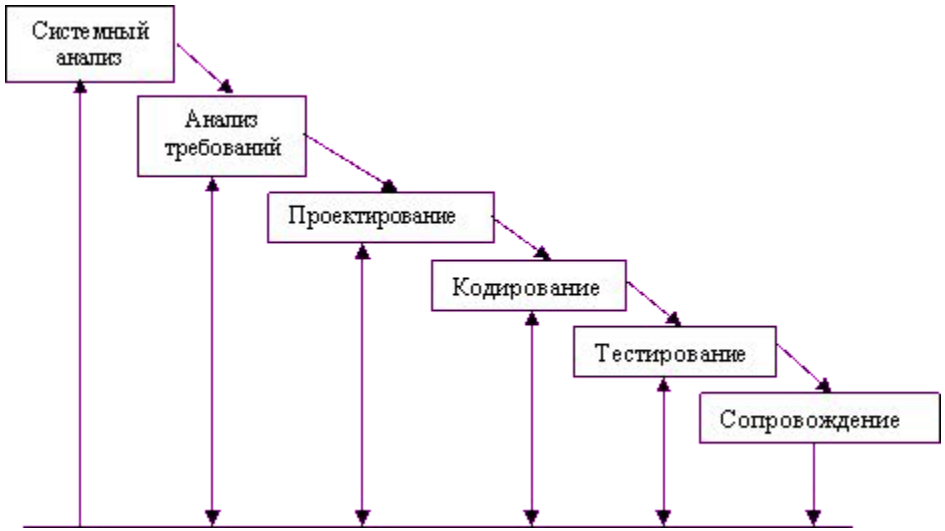
- порядок применения методов и утилит;
- формирование отчетов, форм по соответствующим требованиям;
- контроль, который помогает обеспечивать качество и координировать изменения;
- формирование основ, по которым руководители оценивают прогресс.

Процесс конструирования программного обеспечения состоит из последовательности шагов, использующих методы, утилиты и процедуры. Эти последовательности шагов часто называют парадигмами ТКПО.

Рассмотрим наиболее популярные парадигмы ТКПО.

Классический жизненный цикл

- классический жизненный цикл называют каскадной или водопадной моделью, подчеркивая, что разработка рассматривается как последовательность этапов, причем переход на следующий, иерархически нижний этап происходит только после полного завершения работ на текущем этапе


Системный анализ
Анализ требований
Проектирование
Кодирование
Тестирование
Сопровождение

Системный анализ

- Задает роль каждого элемента в компьютерной системе, взаимодействие элементов друг с другом.

Анализ требований

- Относится к программному элементу — программному обеспечению. Уточняются и детализируются его функции, характеристики и интерфейс.

Проектирование состоит в создании представлений:

- архитектуры ПО;
- модульной структуры ПО;
- алгоритмической структуры ПО;
- структуры данных;
- входного и выходного интерфейса (входных и выходных форм данных).

Кодирование

- Состоит в переводе результатов проектирования в текст на языке программирования.

Тестирование

- Это выполнение программы для выявления дефектов в функциях, логике и форме реализации программного продукта.

Сопровождение

Это внесение изменений в эксплуатируемое ПО. Цели сопровождения:

- исправление ошибок;
- адаптация к изменениям внешней для ПО среды;
- усовершенствование ПО по требованиям заказчика.

- Сопровождение ПО состоит в повторном применении каждого из предшествующих шагов (этапов) жизненного цикла к существующей программе но не в разработке новой программы.

Достоинства и недостатки

Достоинства классического жизненного цикла: дает план и временной график по всем этапам проекта, упорядочивает ход конструирования.

Недостатки классического жизненного цикла:
1) реальные проекты часто требуют отклонения от стандартной последовательности шагов;
2)цикл основан на точной формулировке исходных требований к ПО (реально в начале проекта требования заказчика определены лишь частично);
3) результаты проекта доступны заказчику только в конце работы.

Макетирование

Часто заказчик не может сформулировать подробные требования по вводу, обработке или выводу данных для будущего программного продукта. С другой стороны, разработчик может сомневаться в приспосабливаемости продукта под операционную систему, форме диалога с пользователем или в эффективности реализуемого алгоритма. В этих случаях целесообразно использовать макетирование.

Цель макетирования

Основная цель макетирования — снять неопределенности в требованиях заказчика.

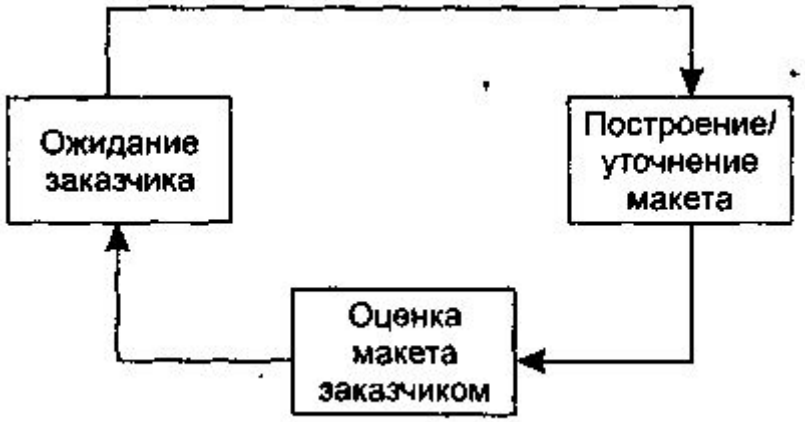
Понятие макетирования

Макетирование (прототипирование) — это процесс создания модели требуемого программного продукта.

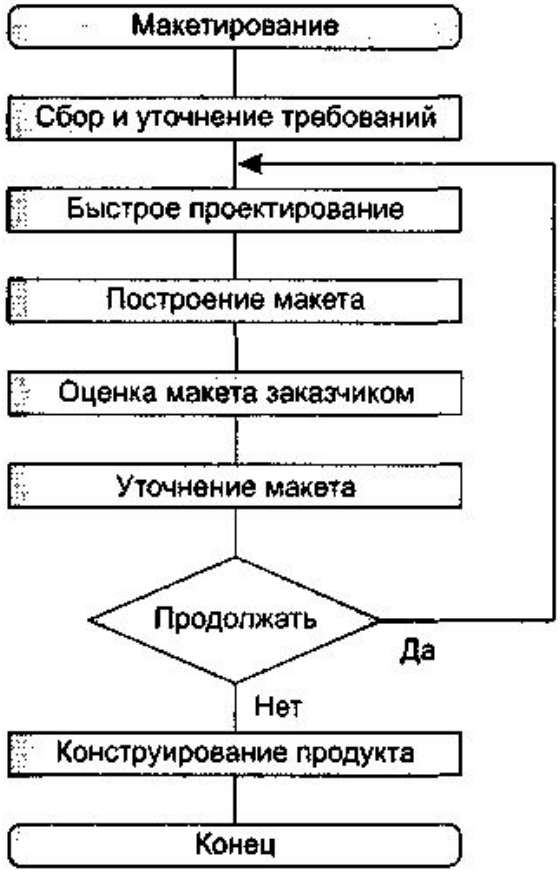
Модель может принимать одну из трех форм:

- Бумажный макет или макет на основе ПК (изображает или рисует человеко-машинный диалог);
- Работающий макет (выполняет некоторую часть требуемых функций);
- Существующая программа (характеристики которой затем должны быть улучшены).

Макетирование основывается на многократном повторении итераций, в которых участвуют заказчик и разработчик.



Последовательность действий при макетировании



Макетирование начинается со сбора и уточнения требований к создаваемому ПО Разработчик и заказчик встречаются и определяют все цели ПО, устанавливают, какие требования известны, а какие предстоит доопределить.

Затем выполняется быстрое проектирование. В нем внимание сосредоточивается на тех характеристиках ПО, которые должны быть видимы пользователю.

Быстрое проектирование приводит к построению макета.

Макет оценивается заказчиком и используется для уточнения требований к ПО.

Итерации повторяются до тех пор, пока макет не выявит все требования заказчика и, тем самым, не даст возможность разработчику понять, что должно быть сделано.

Достоинство макетирования

Обеспечивает определение полных требований к ПО

Недостатки макетирования

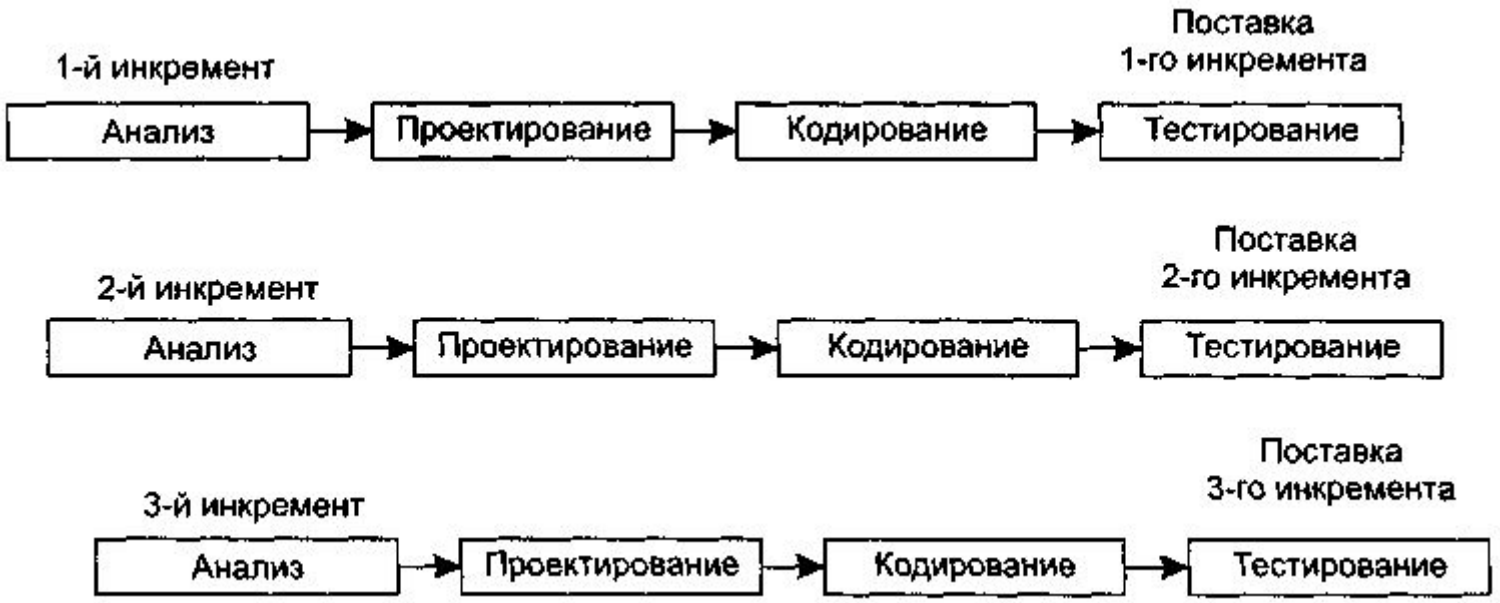
- заказчик может принять макет за продукт;
- разработчик может принять макет за продукт.

Суть недостатков

Когда заказчик видит работающую версию ПО, он перестает сознавать, что детали макета скреплены не надежно

Он забывает, что в погоне за работающим вариантом оставлены нерешенными вопросы качества и удобства сопровождения ПО. Когда заказчику говорят, что продукт должен быть перестроен, он начинает возмущаться и требовать, чтобы макет «в три приема» был превращен в рабочий продукт. Очень часто это отрицательно сказывается на управлении разработкой ПО.

Инкрементная модель



Инкрементная стратегия. В начале процесса определяются все пользовательские и системные требования, оставшаяся часть конструирования выполняется в виде последовательности версий. Первая версия реализует часть запланированных возможностей, следующая версия реализует дополнительные возможности и т. д., пока не будет получена полная система;

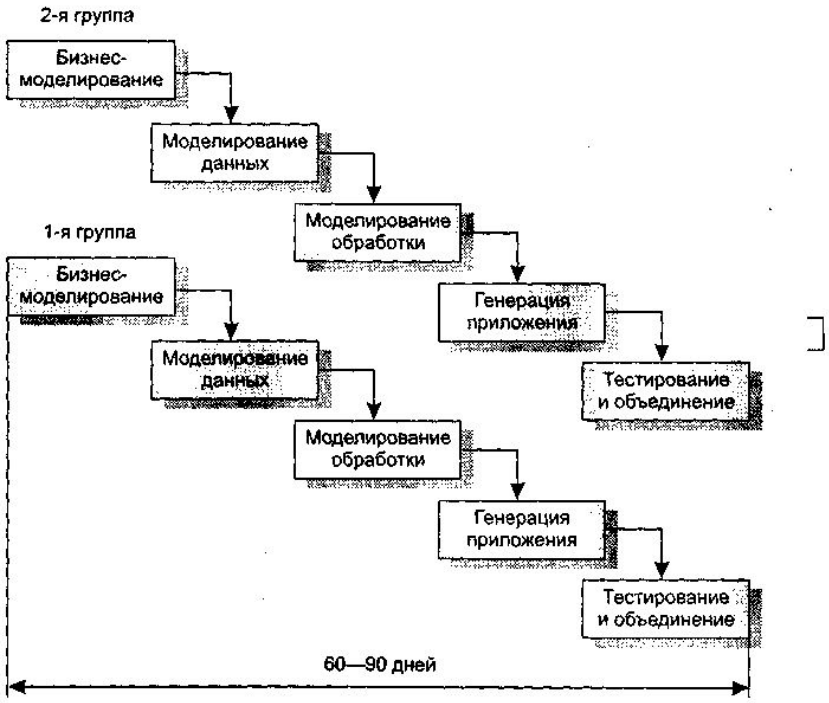
Она объединяет элементы последовательной водопадной модели с итерационной философией макетирования.

Каждая линейная последовательность здесь вырабатывает поставляемый инкремент ПО. Например, ПО для обработки слов в 1-м инкременте реализует функции базовой обработки файлов, функции редактирования и документирования; во 2-м инкременте — более сложные возможности редактирования и документирования; в 3-м инкременте — проверку орфографии и грамматики; в 4-м инкременте — возможности компоновки страницы.

- Первый инкремент приводит к получению базового продукта, реализующего базовые требования (правда, многие вспомогательные требования остаются нереализованными).
- План следующего инкремента предусматривает модификацию базового продукта, обеспечивающую дополнительные характеристики и функциональность.
- По своей природе инкрементный процесс итеративен, но, в отличие от макетирования, инкрементная модель обеспечивает на каждом инкременте работающий продукт.

Быстрая разработка приложений

- Модель быстрой разработки приложений (Rapid Application Development) — второй пример применения инкрементной стратегии конструирования



RAD-модель обеспечивает экстремально короткий цикл разработки. RAD — высокоскоростная адаптация линейной последовательной модели, в которой быстрая разработка достигается за счет использования компонентно-ориентированного конструирования. Если требования полностью определены, а проектная область ограничена, RAD-процесс позволяет группе создать полностью функциональную систему за очень короткое время (60-90 дней).

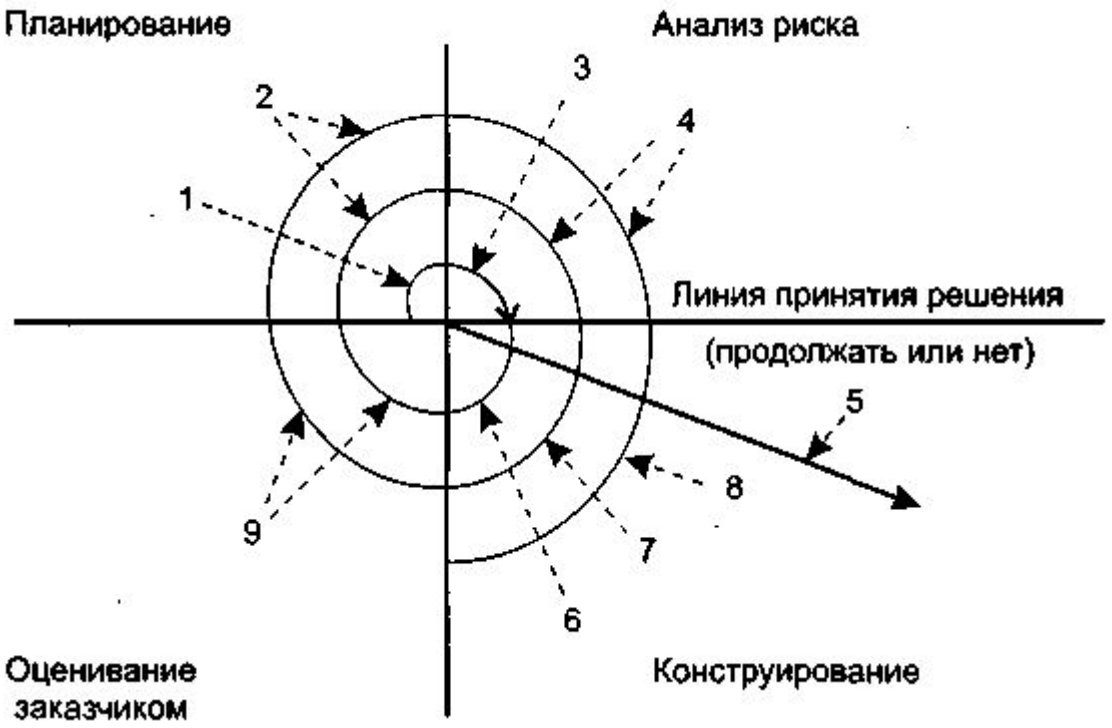
RAD-подход ориентирован на разработку информационных систем и выделяет следующие этапы:

- **бизнес-моделирование.** Моделируется информационный поток между бизнес-функциями. Ищется ответ на следующие вопросы: Какая информация руководит бизнес-процессом? Какая генерируется информация? Кто генерирует ее? Где информация применяется? Кто обрабатывает ее?
- **моделирование данных**. Информационный поток, определенный на этапе бизнес-моделирования, отображается в набор объектов данных, которые требуются для поддержки бизнеса. Идентифицируются характеристики (свойства, атрибуты) каждого объекта, определяются отношения между объектами;
- **моделирование обработки.** Определяются преобразования объектов данных, обеспечивающие реализацию бизнес-функций. Создаются описания обработки для добавления, модификации, удаления или нахождения (исправления) объектов данных;
- **генерация приложения.** Предполагается использование методов, ориентированных на языки объектно-ориентированного программирования. RAD-процесс работает с повторно используемыми программными компонентами или создает повторно используемые компоненты. Для обеспечения конструирования используются утилиты автоматизации;
- **тестирование и объединение.** Поскольку применяются повторно используемые компоненты, многие программные элементы уже протестированы. Это уменьшает время тестирования (хотя все новые элементы должны быть протестированы).

RAD имеет следующие недостатки и ограничения.

- 1. Для больших проектов в RAD требуются существенные людские ресурсы (необходимо создать достаточное количество групп).
- 2. RAD применима только для таких приложений, которые могут декомпозироваться на отдельные модули и в которых производительность не является критической величиной.

Спиральная модель



- Спиральная модель: *1* — начальный сбор требований и планирование проекта; *2* — та же работа, но на основе рекомендаций заказчика; 3 — анализ риска на основе начальных требований; *4* — анализ риска на основе реакции заказчика; 5 — переход к комплексной системе; 6 — начальный макет системы; 7 — следующий уровень макета; *8* — сконструированная система; 9 — оценивание заказчиком

- 1. Планирование — определение целей, вариантов и ограничений.
- 2. Анализ риска — анализ вариантов и распознавание/выбор риска.
- 3. Конструирование — разработка продукта следующего уровня.
- 4. Оценивание — оценка заказчиком текущих результатов конструирования.

- В первом витке спирали определяются начальные цели, варианты и ограничения, распознается и анализируется риск. Если анализ риска показывает неопределенность требований, на помощь разработчику и заказчику приходит макетирование (используемое в квадранте конструирования). Для дальнейшего определения проблемных и уточненных требований может быть использовано моделирование. Заказчик оценивает инженерную (конструкторскую) работу и вносит предложения по модификации (квадрант оценки заказчиком). Следующая фаза планирования и анализа риска базируется на предложениях заказчика. В каждом цикле по спирали результаты анализа риска формируются в виде «продолжать, не продолжать». Если риск слишком велик, проект может быть остановлен.

- В большинстве случаев движение по спирали продолжается, с каждым шагом продвигая разработчиков к более общей модели системы. В каждом цикле по спирали требуется конструирование (нижний правый квадрант), которое может быть реализовано классическим жизненным циклом или макетированием. Количество действий по разработке (происходящих в правом нижнем квадранте) возрастает по мере продвижения от центра спирали.

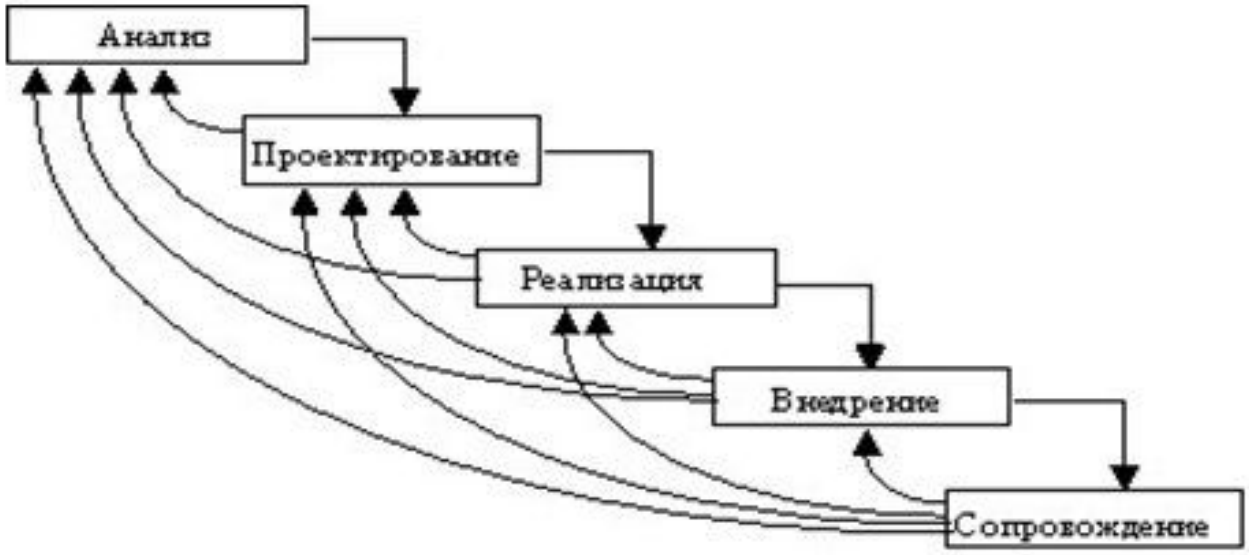
Достоинства

- 1) наиболее реально (в виде эволюции) отображает разработку программного обеспечения;
- 2) позволяет явно учитывать риск на каждом витке эволюции разработки;
- 3) использует моделирование для уменьшения риска и совершенствования программного изделия.

Недостатки

- 1) новизна (отсутствует достаточная статистика эффективности модели);
- 2) повышенные требования к заказчику;
- 3) трудности контроля и управления временем разработки.

Поэтапная модель с промежуточным контролем



- Итерационная модель разработки *ПО* с циклами обратной связи между этапами. Преимущество такой модели заключается в том, что межэтапные корректировки обеспечивают меньшую трудоёмкость *по* сравнению с каскадной моделью; однако время жизни каждого из этапов растягивается на весь период разработки