

Лекция 12

Подходы разработки ПО

Эволюционные подходы

- В них явно видна та же суть, что и у неплан **Эволюционные подходы являются гибкими подходами, основанными на различных моделях прототипирования и связаны с эволюционным представлением разработки продукта.** ируемого подхода, но необходимость повышения характеристик создаваемого ПО привела в них к использованию ряда специальных методик и практик.
- Особенности эволюционных подходов являются:
 1. Использование прототипирования;

Выделяют эволюционные подходы следующих двух видов:

1. Подходы прототипирования.
2. Подходы быстрой разработки: Итеративная инкрементная разработка (IID), Быстрая разработка приложений (RAD), Эволюционная быстрая разработка (ERD), Метод разработки динамических систем (DSDM).

Подходы прототипирования являются вариациями Итеративной инкрементной разработки, на котором основываются и другие подходы быстрой разработки.

Непланируемый подход («кодирование – исправление»)

- *Непланируемый подход* («кодирование – исправление») основан на одноимённой модели. **Фактически это всего лишь способ написания кода программы, простой проверки полученной программы и его модификации, он не предполагает серьёзного проектирования.** Многие разработчики не считают его подходом, называя этот способ *кустарной разработкой*. Непланируемый подход используется при разработке небольших свободно распространяемых программ. Он рекомендуется в случае разработки простого демонстрационного прототипа или проверки некоторой программной концепции. Идея непланируемого подхода оказалась

Прототипируемые подходы

Прототипируемые подходы, или подходы прототипирования, являются одновременно развитием и альтернативой каскадных подходов и основаны, как следует из названия, на прототипировании.

Выделяют следующие основные подходы прототипирования:

1. Эволюционная доставка;
2. Итеративная доставка;
3. Постадийная доставка.

Модели ЖЦ для прототипируемых подходов являются вариантами прототипируемой модели с учётом каскадной и других моделей.

Эволюционная доставка

- **Эволюционная доставка – эволюционный подход, ориентированный в первую очередь на создание пользовательского интерфейса.** Основой модели ЖЦ служит эволюционная модель, так как в начале разработки нет чётко сформулированных требований. Принцип модели (рис.12.1) заключается в том, что первый прототип обычно уже включает развитый пользовательский интерфейс. Далее, пока заказчик не сочтёт продукт законченным, в него вносится необходимая функциональность; при этом возможно небольшое изменения интерфейса.

Рис.12.1. Модель ЖЦ для подхода Эволюционная доставка

Подход применяется в проектах с ярко выраженным преобладанием разработки пользовательского интерфейса. Существенным недостатком подхода является невозможность определить стоимость и продолжительность проекта.



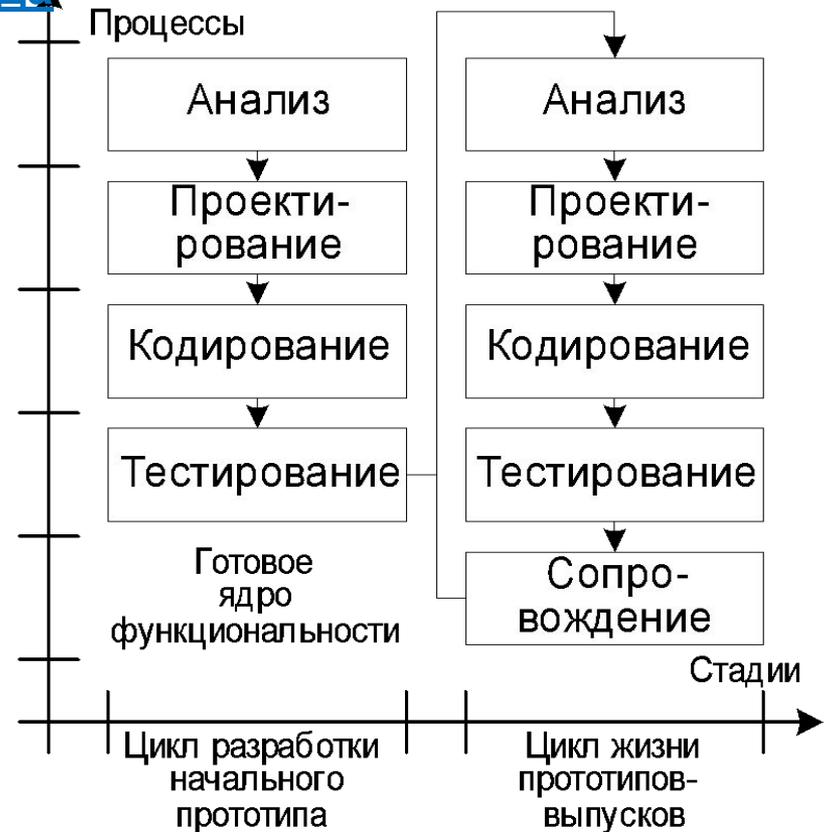
Итеративная доставка

- **Итеративная доставка** – эволюционный подход, ориентированный в первую очередь на создание необходимого ядра функциональности. Основой модели ЖЦ для подхода служит итеративная инкрементная модель, так как в начале разработки известны чётко сформулированные требования. **Принцип модели (рис.12.2) заключается в том, что первый прототип обычно уже включает большую часть необходимой функциональности.** Далее, пока заказчик не сочтёт продукт законченным, для него разрабатывается необходимый пользовательский интерфейс; при этом возможно небольшое изменение

Рис.12.2. Модель ЖЦ для подхода Итеративная доставка

Подход применяется в проектах с ярко выраженным преобладанием разработки функциональности.

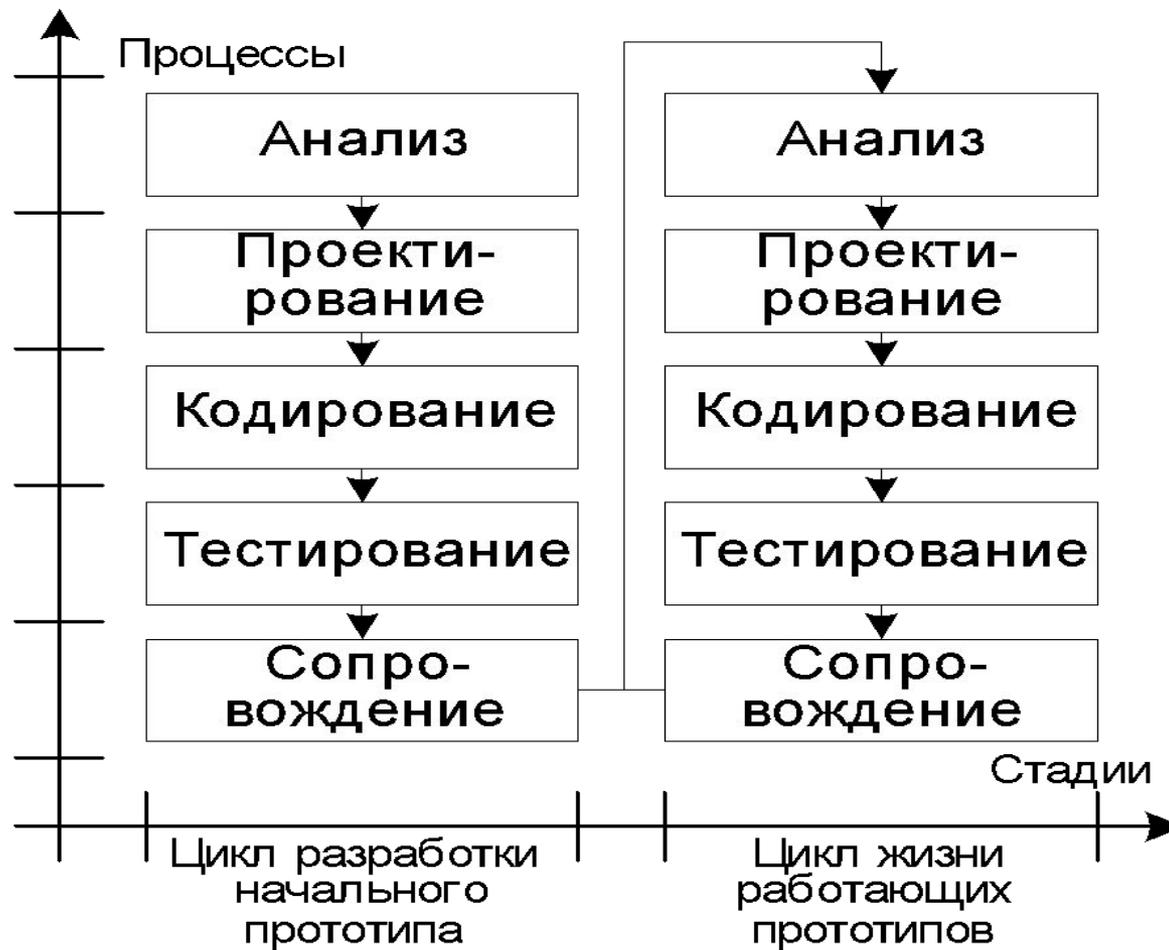
Существенным недостатком подхода также является невозможность определить стоимость и продолжительность проекта.



Постадийная доставка

- **Постадийная доставка – эволюционный подход, ориентированный в первую очередь на создание работающих прототипов.** Подход предназначен решить недостаток двух предыдущих подходов прототипирования – невозможность определения сроков завершения проекта. Это достигается обеспечением работоспособности всех создаваемых прототипов. **Основой модели ЖЦ для подхода служит прототипируемая модель, совмещающая итеративную инкрементную и эволюционную модели.** Это связано с тем, что в начале ЖЦ известны только основные сформулированные требования. Принцип модели (рис.12.3) заключается в том, что первый прототип обычно включает основную часть необходимой функциональности и при этом является работающим (готовым к эксплуатации). Далее, до тех пор, пока заказчик не сочтёт продукт приемлемым, в рамках отдельных проектов на основе имеющегося прототипа создаётся очередной работающий прототип, включающие в себя реализации новых требований заказчика.

Рис.12.3. Модель ЖЦ для подхода Постадийная доставка



Итеративная инкрементная разработка

Итеративная инкрементная разработка (ИИР, IID – Iterative and Incremental Development) – подход разработки, являющийся альтернативой (классическому) каскадному подходу и использующий прототипы.

- Идея повышения качества путём организации производства в виде коротких циклов **«Планирование – Выполнение – Проверка – Действие»** была предложена в 1939 г. в работе У. Шуарта, эксперта по качеству Bell Labs. Эта идея получила развитие в области разработки ПО и привела к созданию в середине 1950-х гг. подхода ИИР. ИИР использовался в ряде исследовательских проектов, выполненных подразделением FSD (Отделение федеральных систем) фирмы IBM по заказу Министерства обороны США.

ИИР стал одним из основных компонентов ряда современных строгих и гибких подходов, в том числе RAD, DSDM, RUP и многих живых подходов. Подход основан на одноимённой модели.

Быстрая разработка приложений

Быстрая разработка приложений (БРП, RAD – Rapid Application Development) – эволюционный подход, сформулированный Дж. Мартином в 1991 г.

- В 1980-х гг. Дж. Мартин, сотрудник фирмы IBM, разработал подход БРП. В 1991 г. он опубликовал книгу под названием «Быстрая разработка приложений». В дальнейшем подход БРП включил в себя другие методики, нацеленные на ускорение разработки приложений.
- **Разработка с использованием БРП часто связана с достижением компромисса между функциональностью и производительностью системы и ускоренной разработкой и обеспечением сопровождения. Он оказывается эффективным в средних проектах для конкретного заказчика при создании системы с ярко выраженным интерфейсом пользователя, наглядно демонстрирующим логику работы.**

БРП обладает следующими особенностями.

- **Команда разработчиков:** малочисленная проектная команда (от 2 до 10 человек) из опытных, разносторонне подготовленных специалистов, способных заменять друг друга.
- **Подход к управлению командой:** максимальная интеграция управленческого аппарата с командой разработчиков с коротким, но тщательно проработанным графиком проекта (от 2 до 6 месяцев).
- **Использование техники прототипирования:** повторяющийся цикл разработки и быстрая разработка прототипа перед каждой итерации для демонстрации пользователям и уточнения требований для этой итерации.

Основные принципы БРП

На основе БРП созданы и другие подходы: Адаптивная разработка ПО (ASD), Метод разработки динамических систем (DSDM) и т.д.

К основным принципам БРП следует отнести следующие:

1. Разработка системы итерациями, способствующая параллельному созданию подсистем отдельной группой разработчиков.
2. Использование прототипирования, позволяющее полнее выяснить и удовлетворить потребности конечного пользователя.
3. Обязательное вовлечение пользователей в процесс разработки системы.
4. Ведение разработки немногочисленной хорошо управляемой командой профессионалов.
5. Грамотное руководство разработкой системы, чёткое планирование и контроль выполнения работ.

Основные принципы БРП

6. Необязательность полного завершения работ на каждой из фаз ЖЦ.
7. Необходимое применение CASE-средств, обеспечивающих целостность проекта на всём протяжении ЖЦ.
8. Применение средств управления конфигурацией, облегчающих внесение изменений в проект и сопровождение готовой системы.
9. Непременное использование генераторов кода.
10. Тестирование и развитие проекта одновременно с разработкой.

Метрика – оценка размера приложения

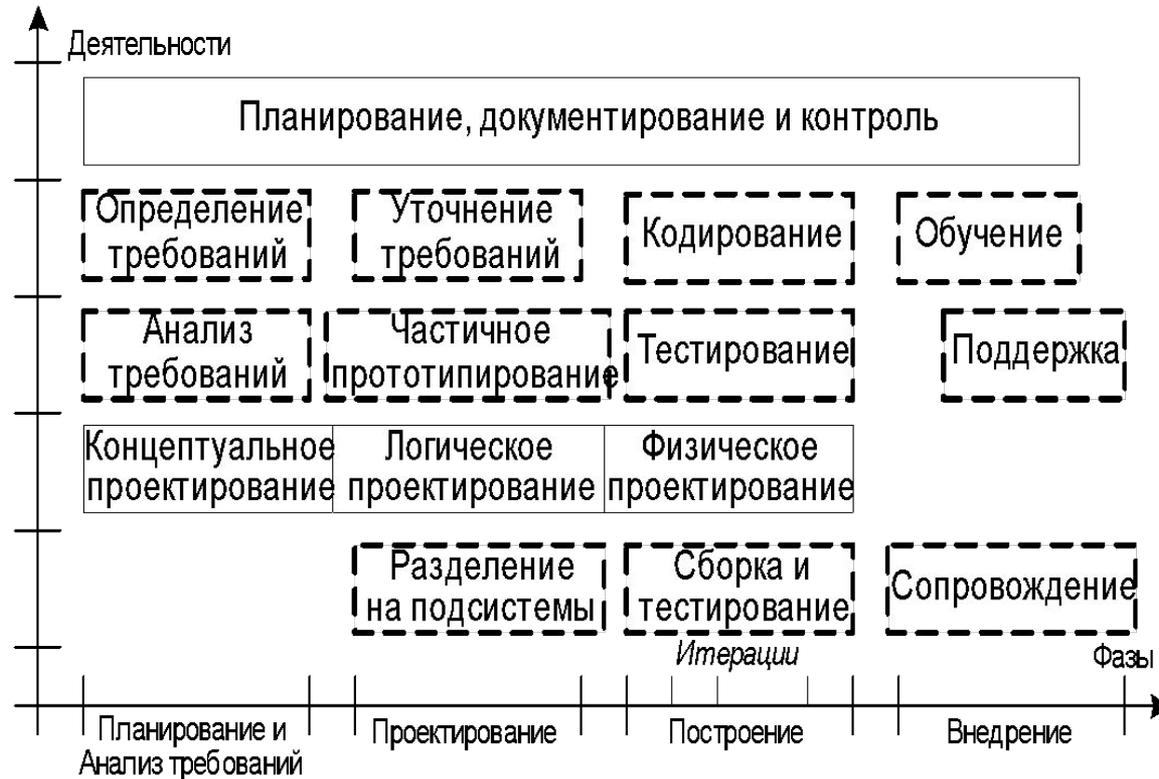
Для выполнения разработки согласно перечисленным принципам используется метрика – оценка размера приложения. **Эта метрика вычисляется на основе так называемых функциональных элементов (экранных форм, документов, отчётов и т.п.).** Размер приложения, разрабатываемое с помощью БРП, для хорошо отлаженной среды разработки с максимальным повторным использованием компонентов определяется следующим образом: Один человек – при числе функциональных элементов менее 1000, одна команда – при 1000 – 4000, 4000 функциональных элементов на одну команду разработчиков – при более 4000.

- **Основой модели ЖЦ для подхода служит итеративная инкрементная модель, ориентированная на разработку работающих прототипов (рис.12.4).**

Рис.12.4. Схема модели ЖЦ для подхода RAD

В БРП выделены следующие 4 фазы:

1. Планирование и Анализ требований;
2. Проектирование;
3. Построение;
4. Внедрение.



Фаза 1

- **На фазе 1 пользователи системы определяют функции, которые она должна выполнять, выделяют наиболее приоритетные из них, описывают информационные потребности.**
Определение требований выполняется в основном силами пользователей под руководством специалистов-разработчиков. Ограничивается масштаб проекта, определяются временные рамки для каждой из последующих фаз. Кроме того, определяется сама возможность реализации данного проекта в установленных рамках финансирования, на данном аппаратном обеспечении и т.п.
Результатами фазы должны быть список и приоритетность функций системы, предварительные информационные и функциональные модели этой системы

Фаза 2

На фазе 2 часть пользователей принимает участие в логическом проектировании системы под руководством разработчиков. Для быстрого получения работающих прототипов приложений используется CASE-средство. Пользователи уточняют и дополняют требования к системе, которые не были выявлены на предыдущей фазе.

- Более подробно рассматриваются процессы системы.
- Анализируется и корректируется функциональная модель.
- Каждый процесс рассматривается детально.
- Определяются требования разграничения доступа к данным.
- На этой же фазе происходит определение набора необходимой документации.
- После детального определения состава процессов оценивается количество функциональных элементов системы и принимается решение о разделении её на подсистемы, поддающиеся реализации одной командой разработчиков за приемлемое для RAD-проектов время – порядка 60 – 90 дней.
- С использованием CASE-средства проект распределяется между различными командами.

Результатами фазы должны быть: общая информационная модель системы, функциональные модели системы в целом и подсистем, точно определённые с помощью CASE-средства интерфейсы между автономно разрабатываемыми подсистемами, построенные прототипы экранов, отчётов, диалогов.

Фаза 3

На фазе 3 выполняется собственно быстрая разработка приложения.

На этой фазе

- разработчики производят итеративное построение реальной системы на основе полученных в предыдущей фазе моделей, а также требований нефункционального характера.
- Код частично формируется при помощи генераторов, получающих информацию из репозитория CASE-средства.
- Конечные пользователи на этой фазе оценивают получаемые результаты и вносят коррективы, если в процессе разработки система перестаёт удовлетворять определённым ранее требованиям.
- Тестирование осуществляется непосредственно в процессе разработки.
- После окончания работ каждой отдельной команды разработчиков производится постепенная интеграция части системы с остальными, формируется полный код, выполняется тестирование совместной работы этой части с остальными, а затем тестирование системы в целом.
- Завершается физическое проектирование системы: определяется необходимость распределения данных, производится анализ использования данных, производится физическое проектирование базы данных, определяются требования к аппаратным ресурсам, определяются способы увеличения производительности.
- Завершается разработка документации.

Результатом фазы является готовая система, удовлетворяющая всем согласованным требованиям.

Фаза 4

- **На фазе 4 производится обучение пользователей, выполняются организационные изменения.**

Параллельно с внедрением новой системы осуществляется работа с существующей системой (до полного внедрения новой). Так как фаза построения достаточно непродолжительна, подготовка к внедрению должна начинаться заранее, как правило, в фазе проектирования.

Приведённая схема разработки не является абсолютной. Возможны различные её варианты в зависимости от целей проекта.