**Лекция №4**

**Методология и технология разработки информационных систем**

Методология создания информационных систем заключается в организации процесса построения информационной системы и в управлении этим процессом для того, чтобы гарантировать выполнение требований, как к самой системе, так и к характеристикам процесса разработки.  
Основными задачами, решение которых должна обеспечивать методология создания информационных систем (с помощью соответствующего набора инструментальных средств), являются:

* соответствие создаваемой информационной системы целям и задачам предприятия, а также предъявляемым к ней требованиям по автоматизации бизнес-процессов;
* гарантирование создания системы с заданными параметрами в течение заданного времени в рамках оговоренного заранее бюджета;
* простота сопровождения, модификации и расширения системы с целью обеспечения ее соответствия изменяющимся условиям работы предприятия;
* соответствие создаваемой корпоративной информационной системы требова­ниям открытости, переносимости и масштабируемости;
* возможность использования в создаваемой системе разработанных ранее и при­меняемых на предприятии средств информационных технологий (программ­ного обеспечения, баз данных, средств вычислительной техники, телекомму­никаций).

Методологии, технологии и инструментальные средства проектирования (CASE-средства) составляют основу проекта любой информационной системы. Методология реализуется через конкретные технологии и поддерживающие их стандар­ты, методики и инструментальные средства, которые обеспечивают выполнение процессов жизненного цикла информационных систем.  
Основное содержание технологии проектирования составляют технологические инструкции, состоящие из описания последовательности технологических опера­ций, условий, в зависимости от которых выполняется та или иная операция, и опи­саний самих операций.  
Технология проектирования может быть представлена как совокупность трех составляющих:

* заданной последовательности выполнения технологических операций проектирования;
* критериев и правил, используемых для оценки результатов выполнения технологических операций;
* графических и текстовых средств (нотаций), используемых для описания проектируемой системы.

Каждая технологическая операция должна обеспечиваться следующими материальными, информационными и людскими ресурсами:

* данными, полученными на предыдущей операции (или исходными данными), представленными в стандартном виде;
* методическими материалами, инструкциями, нормативами и стандартами;
* программными и техническими средствами;
* исполнителями.

Результаты выполнения операции должны представляться в некотором стандартном виде, обеспечивающем их адекватное восприятие при выполнении следующей технологической операции (на которой они будут использоваться в качестве исходных данных).  
Можно сформулировать ряд общих требований, которым должна удовлетворять технология проектирования, разработки и сопровождения информационных си­стем:

* поддерживать полный жизненный цикл информационной системы;
* обеспечивать гарантированное достижение целей разработки системы с заданным качеством и в установленное время;
* обеспечивать возможность разделения (**декомпозиции**) крупных проектов на ряд подсистем — составных частей, разрабатываемых группами исполнителей ограниченной численности, с последующей интеграцией этих частей;

[http://mirea.kremlina.ru/templates/Default/dleimages/spoiler-plus.gif](javascript:ShowOrHide('spbf8d31789664b95053669c204a6e168b')) [Примечание](javascript:ShowOrHide('spbf8d31789664b95053669c204a6e168b'))

* обеспечивать возможность ведения работ по проектированию отдельных подсистем небольшими группами, что обусловлено принципами управляемости коллектива и повышения производительности за счет миними­зации числа внешних связей;
* обеспечивать минимальное время получения работоспособной системы;
* предусматривать возможность управления конфигурацией проекта, ведения версий проекта и его составляющих, автоматического выпуска проектной документации и синхронизацию ее версий с версиями проекта;
* обеспечивать независимость выполняемых проектных решений от средств реализации системы — системы управления базами данных, операционной системы, языка и системы программирования.

## Методология RAD

На начальном этапе существования компьютерных информационных систем их разработка велась на традиционных языках программирования. Однако по мере возрастания сложности разрабатываемых систем и запросов пользователей (чему в значительной степени способствовал прогресс в области вычислительной техники, а также появление удобного графического интерфейса пользователя в систем­ном программном обеспечении) требовались новые средства, обеспечивающие значительное сокращение сроков разработки. Это послужило предпосылкой к созданию целого направления в области программного обеспечения — инструмен­тальных средств для быстрой разработки приложений. Развитие этого направления привело к появлению на рынке программного обеспечения средств автоматизации практически всех этапов жизненного цикла информационных систем.

### Основные особенности методологии RAD

Методология создания информационных систем, основанная на использовании средств быстрой разработки приложений, получила в последнее время широкое распространение и приобрела название *методологии быстрой разработки приложений*(Rapid Application Development, RAD). Данная методология охватывает все этапы жизненного цикла современных информационных систем.  
**Методология RAD** — это комплекс специальных инструментальных средств, позволяющих оперировать с определенным набором графических объектов, функционально отображающих отдельные информационные компоненты приложе­ний.  
Под методологией быстрой разработки приложений обычно понимается процесс разработки информационных систем, основанный на трех основных элементах:

* небольшой команде программистов (обычно от 2 до 10 человек);
* тщательно проработанном производственном графике работ, рассчитанном на сравнительно короткий срок разработки (от 2 до 6 мес.);
* итерационной модели разработки, основанной на тесном взаимодействии с заказчиком — по мере выполнения проекта разработчики уточняют и реализуют в продукте требования, выдвигаемые заказчиком.

Основные принципы методологии RAD можно свести к следующим:

* используется итерационная (спиральная) модель разработки;
* полное завершение работ на каждом из этапов жизненного цикла не обязательно;
* в процессе разработки информационной системы обеспечивается тесное взаимодействие с заказчиком и будущими пользователями;
* применяются CASE-средства и средства быстрой разработки приложений;
* применяются средства управления конфигурацией, облегчающие внесение из­менений в проект и сопровождение готовой системы;
* используются прототипы, позволяющие полнее выяснить и реализовать потреб­ности конечного пользователя;
* тестирование и развитие проекта осуществляются одновременно с разработ­кой;
* разработка ведется немногочисленной и хорошо управляемой командой профессионалов;
* обеспечиваются грамотное руководство разработкой системы, четкое планирование и контроль выполнения работ.

### Объектно-ориентированный подход

Средства RAD позволили реализовать совершенно иную по сравнению с традиционной технологию создания приложений: информационные объекты формируются как некие действующие модели (прототипы), чье функционирование согласу­ется с пользователем, а затем разработчик может переходить непосредственно к формированию законченных приложений, не теряя из виду общей картины проектируемой системы.  
Возможность использования подобного подхода в значительной степени является результатом применения принципов объектно-ориентированного проектирования. Эти принципы позволяют преодолеть одну из главных трудностей, возникающих при разработке сложных систем, — колоссальный разрыв между реальным миром (предметной областью) и имитирующей средой.  
Использование объектно-ориентированных принципов позволяет создать описа­ние (модель) предметной области в виде совокупности объектов — сущностей, объе­диняющих данные и методы обработки этих данных (процедуры). Каждый объект обладает собственным поведением и моделирует некоторый объект реального мира. С этой точки зрения объект является вполне осязаемым и демонстрирует опреде­ленное поведение.  
В объектном подходе акцент переносится на конкретные характеристики физической или абстрактной системы, являющейся предметом программного моделирования. Объекты обладают целостностью, которая не может быть нарушена. Таким образом, свойства, характеризующие объект и его поведение, остаются неизменными. Объект может только менять состояние, управляться или становиться в определенное отношение к другим объектам.  
Широкое распространение объектно-ориентированное программирование (ОПП) получило с появлением средств визуального программирования, которые обеспе­чивают слияние (инкапсуляцию) данных с процедурами, описывающими поведе­ние реальных объектов, в объекты программ, которые могут быть отображены определенным образом в графической пользовательской среде. Это позволило приступить к созданию программных систем, максимально похожих на реальные, и добиваться наивысшего уровня абстракции.  
При разработке приложений с помощью инструментов RAD используются множество готовых объектов, сохраняемых в общедоступном хранилище. Однако имеется также возможность разработки новых объектов. При этом новые объекты мо­гут разрабатываться как на основе существующих, так и «с нуля».  
Инструментальные средства RAD обладают удобным графическим интерфейсом пользователя и позволяют на основе стандартных объектов формировать простые приложения без написания кода программы. Это является большим преимуществом RAD, так как в значительной степени сокращает рутинную работу по разработке интерфейсов пользователя (при использовании обычных средств разработка интерфейсов представляет собой достаточно трудоемкую задачу, решение которой отнимает много времени). Высокая скорость разработки интерфейсной части приложений позволяет быстро создавать прототипы и упрощает взаимодействие с конечными пользо­вателями.  
Таким образом, инструменты RAD позволяют разработчикам сконцентрировать усилия на сущности реальных бизнес-процессов предприятия, для которого создается информационная система. В итоге это приводит к повышению качества разрабатываемой системы.

### Визуальное программирование

Применение принципов объектно-ориентированного программирования позволи­ло создать принципиально новые средства проектирования приложений, называе­мые средствами *визуального программирования.*Визуальные инструменты RAD позволяют создавать сложные графические интерфейсы пользователя вообще без написания кода программы. При этом разработчик может на любом этапе наблю­дать то, что закладывается в основу принимаемых решений.  
Визуальные средства разработки оперируют, в первую очередь, со стандартными интерфейсными объектами — окнами, списками, текстами, которые легко можно связать с данными из базы данных и отобразить на экране монитора. Другая группа объектов представляет собой стандартные элементы управления — кнопки, переключатели, флажки, меню и т. п., с помощью которых осуществляется управление отображаемыми данными. Все эти объекты могут быть стандартным образом описаны средствами языка, а сами описания сохранены для многократного использования в будущем.  
В настоящее время существует довольно много различных визуальных средств разработки приложений, но все они могут быть разделены на две группы — универсальные и специализированные.  
Среди **универсальных** систем визуального программирования сейчас наиболее распространены такие, как Borland Delphi и Visual Basic. Универсальными мы их называем потому, что они не ориентированы исключительно на разработку при­ложений баз данных — с их помощью могут быть разработаны приложения почти любого типа, в том числе и информационные приложения. Причем программы, разрабатываемые с помощью универсальных систем, могут взаимодействовать практически с любыми системами управления базами данных.  
**Специализированные** средства разработки ориентированы только на создание приложений баз данных. Причем, как правило, они привязаны к вполне определенным системам управления базами данных. В качестве примера таких систем можно привести системы Power Builder фирмы Sybase (естественно, предназначенную для работы с СУБД Sybase Anywhere Server) и Visual FoxPro фирмы Microsoft.  
Поскольку задачи создания прототипов и разработки пользовательского интер­фейса, по существу, слились, программист получил непрерывную обратную связь с конечными пользователями, которые могут не только наблюдать за созданием приложения, но и активно участвовать в нем, корректировать результаты и свои требования. Это также способствует сокращению сроков разработки и является важным психологическим аспектом, который привлекает к RAD все большее число разработчиков.

### Событийное программирование

Логика приложения, построенного средствами RAD, является событийно-ориентированной. Это означает, что каждый объект, входящий в состав приложения, может генерировать события и реагировать на события, генерируемые другими объектами. Примерами событий могут быть открытие и закрытие окон, щелчок на кнопке, нажатие клавиши клавиатуры, движение мыши, изменение данных в базе данных и т. п.  
Разработчик реализует логику приложения путем определения обработчика каждого события — процедуры, выполняемой объектом при наступлении соответствующего события. Например, обработчик события «щелчок на кнопке» может открыть диалоговое окно. Таким образом, управление объектами осуществляется с помощью событий.

### Фазы жизненного цикла в рамках методологии RAD

При использовании методологии быстрой разработки приложений жизненный цикл информационной системы состоит из четырех фаз:

* анализа и планирования требований;
* проектирования;
* построения;
* внедрения.

Рассмотрим каждую из них более подробно.

#### Фаза анализа и планирования требований

На фазе анализа и планирования требований определяются:

* функции, которые должна выполнять разрабатываемая информационная система;
* наиболее приоритетные функции, требующие разработки в первую очередь; Q информационные потребности;
* масштаб проекта;
* временные рамки для каждой из последующих фаз;
* сама возможность реализации данного проекта в установленных рамках финансирования на имеющихся аппаратных и программных средствах.

Если реализация проекта принципиально возможна, то результатом фазы анализа и планирования требований будет список функций разрабатываемой информационной системы с указанием их приоритетов, а также предварительные функциональные и информационные модели системы.

#### Фаза проектирования

На фазе проектирования необходимым инструментом являются CASE-средства, используемые для быстрого получения работающих прототипов приложений.  
Прототипы, созданные с помощью CASE-средств, анализируются пользователями, которые уточняют и дополняют те требования к системе, которые не были выявлены на предыдущей фазе. Таким образом, на данной фазе также необходимо участие будущих пользователей в техническом проектировании системы.  
Далее на этой фазе проводится анализ и, если требуется, корректировка функциональной модели системы. Детально рассматривается каждый процесс системы. При необходимости для каждого элементарного процесса создается частичный прототип: экран, диалоговое окно или отчет (это позволяет устранить неясности или неоднозначности). Затем определяются требования разграничения доступа к данным.  
После детального рассмотрения процессов определяется количество функциональных элементов разрабатываемой системы. Это позволяет разделить информационную систему на ряд подсистем, каждая из которых реализуется одной командой разработчиков за приемлемое для RAD-проектов время (порядка полутора месяцев). С использованием CASE-средств проект распределяется между различными командами — делится функциональная модель.  
На этой же фазе происходит определение набора необходимой документации.  
Результатами данной фазы являются:

* общая информационная модель системы;
* функциональные модели системы в целом и подсистем, реализуемых отдельными командами разработчиков;
* точно определенные с помощью CASE-средства интерфейсы между автономно разрабатываемыми подсистемами;
* построенные прототипы экранов, диалоговых окон и отчетов.

#### Фаза построения

На фазе построения выполняется собственно быстрая разработка приложения. На данной фазе разработчики производят итеративное построение реальной системы на основе полученных ранее моделей, а также требований нефункционального ха­рактера. Разработка приложения ведется средствами визуального программиро­вания. Формирование программного кода частично выполняется с помощью автоматических генераторов кода, входящих в состав CASE-средств. Код генерируется на основе разработанных моделей.  
На фазе построения также требуется участие пользователей системы, которые оценивают получаемые результаты и вносят коррективы, если в процессе разработки система перестает удовлетворять определенным ранее требованиям. Тестирова­ние системы осуществляется непосредственно в процессе разработки. После окончания работ каждой отдельной команды разработчиков производится постепенная интеграция данной части системы с остальными, формируется полный программный код, выполняется тестирование совместной работы данной части приложения с остальными, а затем тестирование системы в целом.  
Завершается физическое проектирование системы, а именно:

* определяется необходимость распределения данных;
* производится анализ использования данных;
* производится физическое проектирование базы данных;
* определяются требования к аппаратным ресурсам;
* определяются способы повышения производительности;
* завершается разработка документации проекта.

Результатом реализации данной фазы является готовая информационная система, удовлетворяющая всем требованиям пользователей.

#### Фаза внедрения

Фаза внедрения в основном сводится к обучению пользователей разработанной информационной системы.  
Так как фаза построения достаточно непродолжительна, планирование и подготовка к внедрению должны начинаться заранее, еще на этапе проектирования системы.

### Ограничения методологииRAD

Несмотря на все свои достоинства, методология RAD (как, впрочем, и любая другая методология) не может претендовать на универсальность. Ее применение наиболее эффективно при создании сравнительно небольших систем, разрабатываемых для конкретного заказчика.  
При разработке типовых систем, не являющихся законченным продуктом, а представляющих собой совокупность типовых элементов информационной системы, большое значение имеют такие показатели проекта, как управляемость и качество, которые могут войти в противоречие с простотой и скоростью разработки. Это связано с тем, что типовые системы обычно централизованно сопровождаются и могут адаптироваться к различным программно-аппаратным платформам, системам управления базами данных, коммуникационным средствам, а также интегрироваться с существующими разработками. Поэтому для такого рода проектов необходимы высокий уровень планирования и жесткая дисциплина проектирования, строгое следование заранее разработанным протоколам и интерфейсам, что снижает скорость разработки.  
Методология RAD не подходит для создания не только типовых информацион­ных систем, но и сложных расчетных программ, операционных систем и программ управления сложными инженерно-техническими объектами, то есть программ, требующих написания большого объема уникального кода.  
Методология RAD не может быть использована для разработки приложений, в которых интерфейс пользователя является вторичным, то есть отсутствует наглядное определение логики работы системы. Примерами таких приложений могут служить приложения реального времени, драйверы или службы.  
Совершенно неприемлема методология RAD для разработки систем, от которых зависит безопасность людей, например систем управления транспортом или атомными электростанциями. Это обусловлено тем, что итеративный подход, являющийся одной из основ RAD, предполагает, что первые версии системы не будут полностью работоспособными, что в данном случае может привести к серьезнейшим катастрофам.