

1. Общие понятия

1.1. Что такое информатизация бизнеса и зачем мне надо это знать?

Любая работающая на рынке компания создана для того, чтобы сделать ее владельцев богаче. Это главная цель практически любого бизнеса, за некоторыми исключениями.

Под словами «сделать... богаче» подразумеваются не только ежегодные дивиденды, которые получают владельцы бизнеса. Результат деятельности каждой организации можно измерить в трех показателях — прибыль, капитализация и репутация. Если деятельность предприятия не направлена на улучшение этих показателей, то ее генеральный директор управляет им неэффективно.

Основным (но не единственным) источником прибыли и капитализации является операционный поток наличности — выручка, которую приносит компании ее непосредственная деятельность. Поскольку у любой компании существуют конечные потребители, количество которых ограничено, и она практически никогда не работает в сегменте рынка одна, у конкурентов необходимо забирать максимально возможный операционный поток наличности. Эта задача достигается путем определения и реализации стратегии бизнеса. Иными словами, если цель бизнеса — экономические, социальные или иные стремления, ради которых существует предприятие, то стратегия бизнеса — это единый набор планов и действий, направленных на достижение основных целей бизнеса организации.

Бизнес-стратегия часто разрабатывается по нескольким направлениям: стратегии роста (продуктовая, повышение эффективности, капитализация, партнерство, построение сети), инновационные стратегии, конкурентные стратегии («гонка за лидером», дифференциации, лидерства в издержках, «война стандартов», выживания, поглощения).

Все эти стратегии преследуют свои цели. Цели бизнеса и цели бизнес-стратегии компании должны быть согласованы. Кроме того, бизнес-стратегия зависит от текущего и перспективного состояния рынка.



Например, если цель бизнеса — увеличить выручку до 150 млн руб., а текущая выручка равна 100 млн руб. (7 % на рынке), то стратегии на стабильном рынке и растущем рынке будут разными. На стабильном рынке необходимо наращивать производственную мощность, чтобы дать продукцию для генерации дополнительных 50 млн выручки, и, кроме того, расширять сегмент рынка (отбирать у конкурентов). На растущем рынке нужно просто наращивать производственную мощность.

После определения бизнес-стратегии соответствующих дивизионов компании, выполняющих свои функции бизнеса (маркетинг, производство и т. д.), устанавливают способ достижения целей конкурентной стратегии. Такие стратегии называются функциональными и чаще всего создаются по направлениям «Маркетинг», «Финансы», «Производство/операционная стратегия», «Стратегия НИОКР», «ИТ».

Разработка стратегии ИТ — важный и сложный процесс, который должен ответить на главный вопрос: как должна измениться архитектура ИТ в соответствии с архитектурой (моделью) бизнеса, которая, в свою очередь, корректируется с целью достижения новых требований бизнес- и функциональных стратегий.



В рассмотренном примере, исходя из определенных бизнес-стратегий, будут разработаны разные стратегии ИТ.

<p>На стабильном рынке необходимо наращивать производственную мощность, чтобы дать продукцию для генерации дополнительных 50 млн руб. выручки, и, кроме того, расширять сегмент рынка (отбирать у конкурентов)</p>	<p>Наращивание производственной мощности, возможно, потребует замены существующих ИС более оперативными и масштабными ИС уровня ERP, разработки/доработки ИС собственного производства, внедрения новых ИС сторонних производителей.</p> <p>Расширение сегмента рынка, возможно, приведет к изменению сайта компании, открытию коммерческого портала, внедрению CRM-системы, найму digital-агентства для проведения разовой маркетинговой кампании</p>
<p>На растущем рынке нужно просто наращивать производственную мощность</p>	<p>Очевидно, что в данном случае потребуется только первая часть мероприятий, описанных выше</p>

Естественно, приведенный пример не отражает реального процесса разработки ИТ-стратегии, но демонстрирует основной принцип построения интегрированной с бизнесом ИТ-стратегии: направления развития ИТ должны быть полностью интегрированы с направлениями развития бизнеса.

В условиях динамичного развития рынка информация становится для бизнеса таким же стратегическим ресурсом, как традиционные материальные, финансовые и энергетические ресурсы.

Информатизация — комплекс мероприятий, направленных на то, чтобы модель бизнеса компании наполнялась с помощью ИТ реальным информационным содержанием и получала долгосрочные конкурентные преимущества.

Архитектура ИТ — это согласованная система представлений об организации информационной среды компании. Данная система должна соответствовать модели бизнеса, определять целевое состояние ИТ, быть основой долгосрочного планирования, обеспечиваться процессами и инфраструктурой ИТ. Совокупность модели бизнеса и архитектуры ИТ представляет собой корпоративную архитектуру.

Зачем все это надо знать аналитику?



Аналитик должен всегда помнить, что любая разрабатываемая ИС существует не сама по себе, а в конкретной конечной ИТ-архитектуре, которая регламентируется ИТ-стратегией, согласована с ней и зависит от бизнес-архитектуры. Последняя, в свою очередь, постоянно изменяется под влиянием рынка и целей бизнеса.

Таким образом, разработчики системы всегда должны быть готовы к тому, что требования к разрабатываемой ИС будут постоянно меняться, и, следовательно, ее создатели должны контролировать ситуацию. Количество и степень изменений зависят от стабильности бизнеса и рынка в целом.

Чтобы разрабатывать эффективные приложения, аналитик должен понимать все аспекты, описанные выше, не только владеть информацией об ожиданиях от разрабатываемой системы, но и осознавать ее место в ИТ-архитектуре компании заказчика, тренды изменений этой архитектуры и бизнес-выгоды, которые принесет внедрение новой системы.

1.2. Что такое корпоративное лидерство?

Корпоративное лидерство — сочетание трех важных факторов: эффективной организации, бизнес-синергии и менеджерской команды.

Эффективная организация подразумевает создание корпоративного видения, управленческой команды, архитектуры бизнеса и корпоративной культуры, развитие и поддержание высоких корпоративных моральных стандартов во всех делах, чтобы компания могла создавать повышенную ценность для всех участников бизнеса.



Бизнес-синергия включает в себя создание синергии между корпоративными стратегиями, ресурсами и способностями, заполнение «белых пятен» между бизнес-единицами и дивизионами, создание эффективных связей между ними.

Менеджерская команда: создание здоровой команды корпоративных лидеров и поддержание ее эффективности на высоком уровне, выращивание

новых лидеров, создание системы мотивации сотрудников, наблюдение за тем, чтобы только лучшие люди принимались на работу, развивались дальше, вознаграждались и росли, вкладывая в бизнес свои лучшие силы и способности.

© <http://www.cecsi.ru/coach/synergy.html>

Каждый сотрудник компании должен вносить свой вклад в формирование эффективной организации всех ее составляющих. Менеджеры всех звеньев должны стимулировать участие рядовых сотрудников в этом процессе.

Конечно же, компания не станет лидером на рынке при развитии только этих факторов, но если перечисленные составляющие будут развиваться гармонично, у организации будет гораздо больше шансов стать лидером в своем сегменте. Кроме того, такая компания будет управляемой по целям, прозрачной, совершенствующейся как «сверху», так и «снизу», сможет иметь такие весомые коммерческие преимущества, как гибкость и готовность к изменениям.

К сожалению, на практике в российских компаниях вы редко встретите такой подход, но именно в ваших силах в будущем изменить эту картину.

1.3. Что такое архитектура, какой она бывает и зачем мне надо это знать?



Существует много различных определений архитектуры. Все они имеют свои плюсы и минусы. Мне импонирует определение архитектуры, данное в стандарте ANSI/IEEEStd 1471-2000: «Архитектура — это базовая организация системы, воплощенная в ее компонентах, их отношениях между собой и с окружением, а также принципы, определяющие проектирование и развитие системы».

Архитектура предприятия/корпоративная архитектура (Enterprise Architecture)

Это высокоуровневая архитектура всего предприятия, отражающая бизнес-потребности компании, реализуется с помощью ИТ-средств. Корпоративная архитектура фокусируется на определении потоков и бизнес-процессов, действий, функций, информации, данных и технологий предприятия, а также на вызовах, стоящих перед ИТ, которые необходимы для того, чтобы эффективно применить технологию в ответ на изменение бизнес-потребностей.

Enterprise Architecture объединяет Business Architecture, Information Architecture, Solution Architecture и Technology Architecture [6] (рис. 1).

Business Architecture. Описывает все бизнес-процессы, бизнес-факторы, бизнес-сущности и бизнес-правила с точки зрения бизнеса. Business Architecture не зависит от применяемых ИТ-технологий.

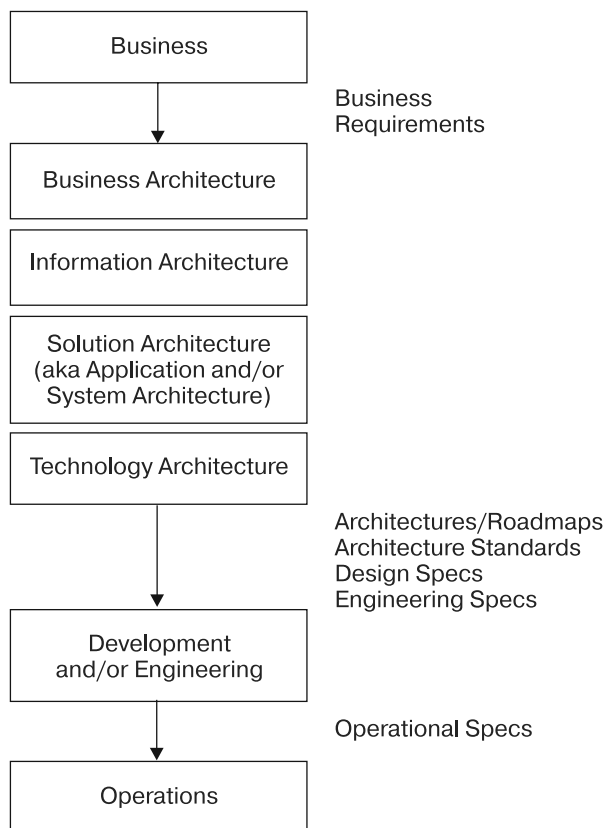


Рис. 1. Архитектура предприятия/корпоративная архитектура

Information Architecture. Определяет структуры данных и описывает все потоки данных, которые используются для поддержки Business Architecture. Такие операции, как идентификация, систематизация, категоризация, хранение данных, относятся к Information Architecture. Может представляться в виде Data Model.

Solution (System/Application) Architecture. Архитектура программного обеспечения, которое реализует функции Business Architecture.

Technology Architecture. Описывает архитектуру ИТ-окружения, которое используется для поддержки Information Architecture и Solution (System/Application) Architecture.

Различные представления архитектуры (Business Architecture, Information Architecture, Solution Architecture и Technology Architecture) тесно взаимосвязаны, и их интеграция – необходимое условие успешного построения Enterprise Architecture.

С точки зрения разработки ПО наиболее важное понятие – Solution (System/Application) Architecture, основными составными частями которой являются System Architecture, Data Architecture и Software Architecture.

System Architecture. Представление системы, которое показывает реализацию функциональных возможностей системы аппаратными средствами и компонентами программного обеспечения, устанавливает связь архитектуры программного обеспечения и архитектуры аппаратных средств, а также регламентирует взаимодействие пользователей с этими компонентами. Существуют и другие определения System Architecture, например: ряд взаимосвязанных шаблонов (паттернов), которые структурируют модули и данные и обеспечивают требуемое поведение системы (см. определение Data Architecture). System Architecture является составной частью Solution Architecture.

Software Architecture. Составная часть System Architecture. Описывает организацию системы с точки зрения программных компонентов, из которых она состоит, и связи между компонентами.

Data Architecture. Составная часть System Architecture. Описывает структуры данных и логические связи между данными (hierarchical model, network model, relational model, object model и т. д.).

Аналитик должен понимать смысл вышеприведенных понятий, их предназначение и взаимосвязь. С точки зрения *развития продукта* важно **понимание Enterprise Architecture, разработки продукта** — Software Architecture, System Architecture, Data Architecture.

1.4. Что такое требование?



Согласно определению Institute of Electrical and Electronics Engineers, Inc., ведущей мировой профессиональной ассоциации по развитию технологий, требование — это:

- (а) условие или способность, необходимая пользователю, чтобы решить проблему или достигнуть цели;
- (б) условие или способность, которыми должны обладать система либо компонент системы для удовлетворения контракта, стандарта, спецификации или другого формально установленного документа;
- (в) документированное представление условия или способности, показанной в определении «а» или «б».

© (IEEE Std 610.12-1990)

Под требованием заказчика понимается потребность или ожидание, которое:

- ◆ установлено;
- ◆ предполагается (подразумевается);
- ◆ является обязательным.

Рассмотрим это на примере. В беседе заказчик выразил ожидание, которое звучит так: «Сделайте мне редактор писем, чтобы я мог выделять разные слова разным цветом». В данном случае:

- ◆ **установлено:** программа должна уметь читать формат писем, редактировать письма и сохранять их;
- ◆ **подразумевалось:** «естественно, я хочу, чтобы это был встроенный в почтовый клиент редактор» (в идеале должно быть установлено, а значит, сформулировано аналитиком явно);
- ◆ **обязательно:** выделять разные слова разным цветом.

Приведенный пример показывает, что недостаточно просто зафиксировать пожелание заказчика так, как он его высказал. Необходимо провести анализ и добиться того, чтобы зафиксированная информация обладала следующими основными характеристиками требования.



Требование должно быть **тестируемым**.

Требование должно быть **реализуемым**.

Требование должно быть **ясным для понимания**.

Требование должно быть **законченным и полным**.

В разработку поступают не любые утверждения заинтересованных лиц. Требование — это результат анализа. Требование должно отвечать на вопрос «что должна делать система», а не как она будет это делать.

Требования бывают разных типов и уровней.

Так, например, по К. Вигерсу [8] структура уровней требований выглядит следующим образом (рис. 2).

В Rational Unified Process (RUP) структура уровней требований выглядит иначе (рис. 3). На данной иллюстрации архитектурные представления не показаны. Более подробно вы можете изучить их в работе [5].

Обычно в компании по созданию ПО разрабатывается своя методология разработки и управления требованиями, которая адаптируется из признанных мировых стандартов. В рамках определения этой методологии формируется и структура уровней требований, или иерархия типов требований. Подробно типы требований будут рассмотрены далее.

1.5. Почему надо знать теории и методологии разработки ПО?

Около десяти лет назад дисциплина системного анализа в том виде, в котором она существует сейчас, находилась в России в стадии становления. Нет, системным анализом занимались и раньше, причем не менее профессионально, чем сейчас. В Советском Союзе существовали специальности «инженер-системотехник», «инженер-математик», наиболее близкие к нынешнему «системному аналитику». Однако в области разработки ПО использовались в основном



Рис. 2. Структура уровней требований по К. Вигерсу

методы структурной и функциональной декомпозиции. Шло время. ИТ-технологии активно развивались, появлялись новые платформы, парадигмы и языки программирования. Некоторыми западными институтами и организациями были предприняты попытки создания и описания процесса разработки программного обеспечения. В настоящее время многие из этих описаний стали де-факто стандартами в области разработки ПО (RUP, CMMI, Swebook и др.).

Одним из первых таких стандартов, с которыми я ознакомился, был RUP. Кажется, это было в 1997 году. В то время я занимался разработкой ПО на Java и структурных языках и не мог не оценить, насколько эффективно данная методология позволяла создавать приложения на языках ООП. Передо мной была поставлена задача по созданию новой системы. Решение этой задачи потребовало взаимодействия с более опытными сотрудниками, создания и согласования ТЗ к системе, программирования и обсуждения полученного ре-

Traceability Overview

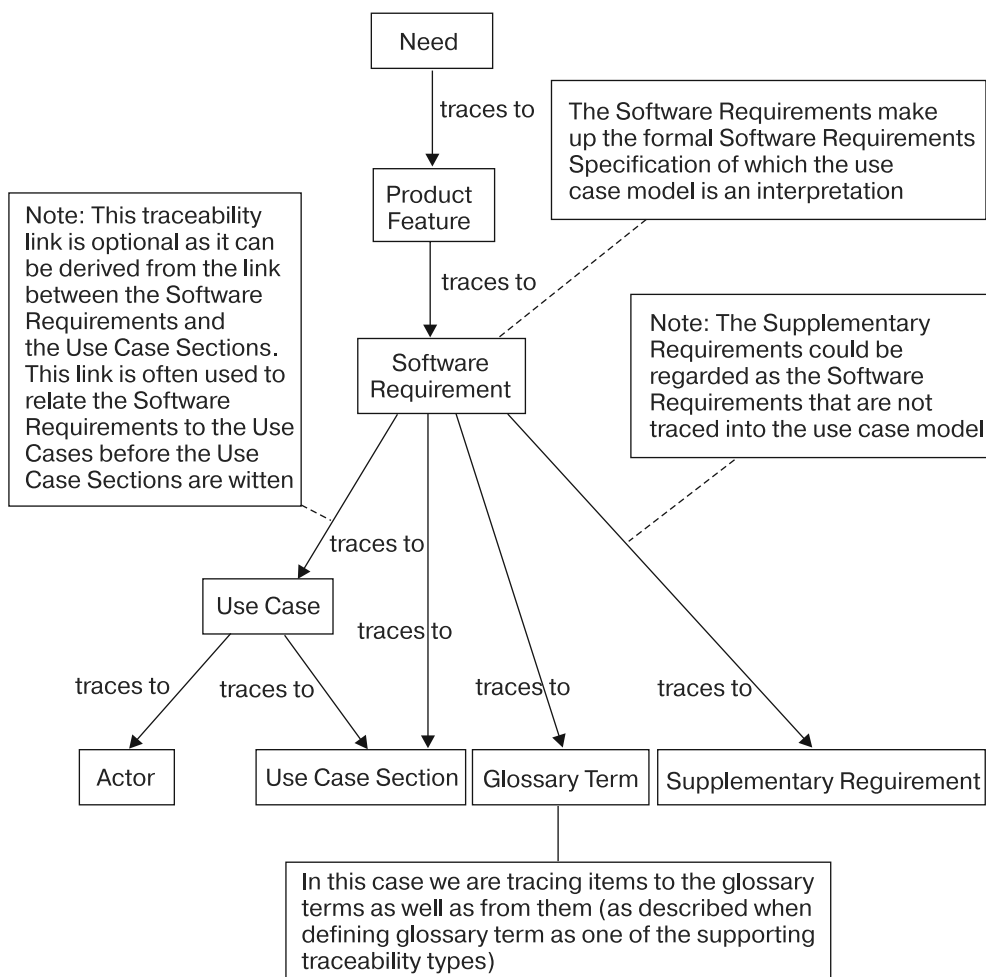


Рис. 3. Структура уровней требований в RUP (Rational Unified Process)

зультата с экспертами и коллегами. Специализированную литературу на русском языке тогда было практически невозможно найти. В стандарте RUP были описаны все роли в процессе разработки ПО и артефакты, посредством которых фиксируются все достигнутые договоренности с проектной командой. Мне стало понятно, какое количество ошибок я допустил, пытаясь выстроить взаимодействие и процесс разработки самостоятельно. Я осознал, почему у меня возникают сложности в согласовании ТЗ, в котором содержались требования к уровню дизайна (по-моему, там даже были какие-то фрагменты алгоритмов и структуры данных), с сотрудниками, выступающими от лица заказчика, но самое главное — я открыл для себя отличный инструмент для моделирования будущей системы и узнал о существовании языка UML.

Это открытие помогло мне построить процесс разработки системы максимально эффективно, а навыки моделирования поведения системы и объектно-ориентированного моделирования пригодились в общении с более опытными коллегами.



Сегодня существует несколько стандартов и методологий для разработки ПО: RUP, MFS, Iconix, спиральная разработка Boehm, Agile, XP и т. д. В книге вы найдете краткое описание и сравнительный анализ некоторых методологий. Не существует «серебряной пули» — методологии, максимально эффективной во всех случаях, — для всех видов проектов, систем любого типа и всех компаний. В каждом конкретном случае компания должна определить эффективную для себя методологию и применять ее при разработке ПО. Обычно за основу берется одна из «стандартных» методологий, которая затем адаптируется под специфику бизнеса организации — производителя ПО.

1.6. Почему бизнес- и системный анализ объединены в одной книге?

Строго говоря, бизнес- и системный анализ — разные дисциплины, и в идеале при разработке ПО должны выделяться две роли: бизнес-аналитик и системный аналитик, которые выполняют разные функции. Однако на практике эти роли часто совмещает один человек или одна из ролей может вообще отсутствовать. Уверен, что и вам в вашей профессиональной деятельности придется совмещать эти роли.

Я убежден в том, что профессиональный системный аналитик должен уметь проводить бизнес-анализ. Для разработки системы даже полезно, если системному аналитику удалось достаточно глубоко погрузиться в бизнес-область, поскольку хорошее понимание потребностей бизнеса приводит к созданию максимально полезных бизнесу продуктов.

В совмещении ролей системного и бизнес-аналитика нет ничего страшного, поскольку большинство методов и практик этих дисциплин пересекаются между собой. Конечно, в компетенции бизнес-аналитика останутся описание и поиск путей оптимизации бизнес-процессов, а в компетенции системного аналитика — моделирование поведения системы и выявление системных требований.

1.7. Что, кроме дисциплин анализа, должен знать хороший аналитик?

В современном мире разработки ПО сложность процессов и взаимодействий нарастает, при этом требования к скорости и качеству выполнения проектов по-

стоянно повышаются. Аналитик находится на переднем крае общения команды и заказчика, и от того, насколько точно и правильно он выявит и сформулирует требования к разрабатываемому продукту, зависит то, правильным путем пойдет вся команда или нет.

Аналитик работает с людьми, у которых есть свои привычки, убеждения, опыт. Для построения эффективных коммуникаций аналитик должен быть чуть-чуть психологом, обладать навыками предотвращения и разрешения конфликтов, уметь одинаково эффективно работать формально и неформально, самостоятельно и в команде. Развитие необходимых личностных факторов будет рассмотрено в главе «Младший аналитик».

Второй аспект работы аналитика — взаимодействие со всеми членами проектной команды: менеджером продукта, менеджером проекта, архитектором, тест-менеджером, разработчиками (программистами), тестировщиками. Здесь не обойтись без специальных технических навыков, иначе аналитику будет очень трудно найти общий язык с техническими специалистами. Аналитик должен не только владеть практиками анализа, но и постоянно расширять свой кругозор в предметной области разрабатываемого продукта и в технологиях, применяемых при производстве этого продукта. В главах «Аналитик» и «Старший/ведущий аналитик» мы поговорим о развитии необходимых профессиональных и специальных технических навыков.

Взаимодействуя с таким большим количеством людей, аналитик должен уметь планировать свою работу, управлять рисками и коммуникациями. Поэтому он должен обладать качествами лидера и немножко быть менеджером проекта. В принципе, работу аналитика можно назвать «проектом в проекте», результатом которого являются спецификации требований, потребителями результата — архитекторы, разработчики и тест-менеджеры, а заинтересованными лицами — практически все члены проектной команды. В главе «Начальник отдела анализа» мы подробнее поговорим о развитии необходимых лидерских и управленческих качеств.

Ну и самое главное: аналитик должен быть высоким профессионалом в технологиях и методах системного и бизнес-анализа, иначе цель его работы с точки зрения «проекта» может быть достигнута, но результат будет некачественным, что приведет к созданию некачественного и/или невостребованного продукта.

Если вас не испугал приведенный перечень всех «должен» — вы на правильном пути, в противном случае стоит подумать о других возможностях профессионального развития.