

Предисловие

За более чем шестидесятилетнюю эволюцию аппаратное обеспечение компьютеров достигло небывалого прогресса. Эмпирическое наблюдение, сделанное Г. Муром в 1965 году, в современной трактовке говорит об удвоении производительности компьютеров каждые два года. Современному специалисту (пользователю) доступна такая вычислительная мощность, которую 10 – 15 лет назад имели немногие научные учреждения. Однако эти вычислительные мощности невозможно использовать без программных систем (ПС). И в этой области, несмотря на значительный рост доступности аппаратных ресурсов, наблюдаются значительные проблемы.

Надо сказать, что компьютерная теория и практика с момента своего образования столкнулись с проблемами, связанными со сложностью программных систем. Первоначально проблемы сложности решались разработчиками путем правильного выбора структур данных, разработки алгоритмов и применения концепции разграничения полномочий. Хотя термин "архитектура программного обеспечения" является относительно новым для индустрии разработки ПС, фундаментальные принципы в этой области неупорядоченно применялись пионерами разработки программного обеспечения, начиная с начала 1980-х годов. Считается, что начало архитектуре программного обеспечения как концепции было положено в научно-исследовательской работе Э. Дейкстры в 1968 году и Д. Парнаса в начале 1970-х. Эти ученые подчеркнули, что структура системы программного обеспечения имеет существенное значение, и построение правильной структуры критически важно.

По данным американских исследователей, в эти годы только 14% проектов по созданию ПС завершались успешно (т.е. с удовлетворением требований заказчика, завершением в срок и соблюдением бюджета). Сегодня, после нескольких десятилетий эволюции языков программирования, инструментальных средств разработки, практически неограниченной доступности машинного времени, ситуация практически не изменилась. Согласно статистике о состоянии дел в программной индустрии в 2008 году, опубликованной компанией Standish Group, из 30 тыс. программных проектов 32% проектов завершились успешно, 44% проектов завершились с проблемами (превысили бюджет, сроки и пр.) и 24% проектов полностью провалились.

На сегодняшний день до сих пор нет согласия в отношении четкого определения термина "архитектура программного обеспечения". Являясь в настоящий момент своего развития дисциплиной без четких правил о "правильном" пути создания системы, проектирование архитектуры ПС все еще является смесью науки и искусства. Популярность изучения этой области возросла с начала 1990-х годов вместе с научно-исследовательской работой по исследованию архитектурных стилей (шаблонов), языков описания архитектуры, документирования архитектуры, и формальных методов проектирования. Известна книга М. Шоу и Д. Гэрлана из университета Carnegie Mellon под названием "Архитектура

программного обеспечения: перспективы новой дисциплины в 1996 году". В книге выдвинуты некоторые концепции архитектуры программного обеспечения, такие как компоненты, соединители (connectors), стили и др. В калифорнийском университете институт Ирвайна по исследованию ПО в первую очередь исследуются архитектурные стили, языки описания архитектуры и динамические архитектуры. Результатами подобных исследований являются популярные монографии, например, книга Л. Басса и др. "Архитектура программного обеспечения на практике". Однако таких трудов не так-то много, тем более в отечественной литературе. В этом свете данная монография, по мнению автора, в некоторой степени снижает дефицит таких публикаций.

Первая глава монографии по своей сути является вводной в проблему архитектурного проектирования. В ней рассматривается широкий круг вопросов, связанных с проблемами создания больших программных систем: особенности разработки сложных (больших) программных систем, инженерный подход к разработке ПС, становление и развитие программной инженерии, развитие технологий программирования, индустрия программного обеспечения. Завершается глава рассмотрением современного состояния ИТ-индустрии в России.

Вторая глава – одна из центральных в монографии. В ней рассматриваются все аспекты архитектур современных программных систем. С точки зрения пользователя программной архитектуры (заинтересованного лица (заказчика), разработчика ПС, специалистов по тестированию, специалистов по развертыванию и сопровождению ПС, а также конечных пользователей), эта архитектура дает направление для рассмотрения и решения задач, связанных со специальностью каждого такого пользователя. Тот факт, что эти несколько различных точек зрения могут быть объединены в архитектуре программного обеспечения, является аргументом в защиту необходимости и целесообразности создания архитектуры ПС еще до этапа разработки ПС. Именно в этом направлении в монографии излагаются архитектуры ПС. В ней дается понятие архитектуры программной системы, отмечаются причины ее важности, откуда и как появляется архитектура, кто и что на нее влияет, что определяет и на что влияет архитектура.

Сегодня в мире существует большое количество различных процессов и технологий для создания ПС, рассматривающих полный жизненный цикл (ЖЦ) проекта разработки ПС и сочетающих в себе научный подход, серьезную базу исследований и имеющих историю реального использования. В третьей главе монографии детально рассматриваются элементы ЖЦ ПС, обобщенные и сформулированные на основе анализа массы публикаций. Освещено само понятие жизненного цикла программных систем, основные, вспомогательные и организационные процессы ЖЦ ПС и их взаимосвязь. Большое внимание уделено современным прогрессивным видам моделей ЖЦ ПС, технологиям и инструментам создания программных систем, в том числе рациональному унифицированному процессу (RUP), SCRAM-методологии и Agile-методологии. Особое место в этом списке занимает технология и инструменты компании IBM Rational

Software. В этом плане интересна заключительная часть главы, в которой рассматривается управление жизненным циклом приложений и интегрированная среда поддержки создания программных систем (Application Lifecycle Management, ALM) на основе комплекса решений IBM Rational Software and Systems Delivery, являющихся наиболее полным по спектру реализованных компонентов ALM, и проекта IBM Jazz.

В четвертой главе рассмотрены вопросы проектирования программных систем в части определения требований и целей программного продукта. Материал этой главы представляет собой обобщение опубликованных монографий и статей, а также собственных исследований автора. Процесс проектирования рассматривается как последовательная трансляция требований, предъявляемых к ПС. Оригинальными являются формализованная схема процесса проектирования, а также предложенный автором подход, который может быть использован для выбора тех требований, предъявляемых к системе, которые наиболее совместимы с другими требованиями. Даны постановка задачи и принципы разработки требований, включая бизнес-моделирование, определение функциональных и нефункциональных требований. Рассмотрены вопросы анализа и управления требованиями, соотношения требований и рисков, проверка правильности требований, формирование целей программного продукта и проекта.

Пятая глава содержит материал, позволяющий решать вопросы разработки предварительного внешнего проекта программной системы. Здесь рассмотрено представление и анализ требований, роль моделирования в определении требований и спецификаций, разработка программных систем, управляемая моделями. Продолжена формальная схема предыдущей главы применительно к описанию спецификаций. Уделено внимание инструментам IBM Rational RequisitePro, Rational Software Architect и IBM Rational Software Modeler, а также другим средствам IBM. Дан структурный и объектный подход в анализе требований и определении спецификаций, в том числе: метод функционального моделирования, функциональные диаграммы, диаграммы потоков данных и др., достаточные сведения о языке UML как языке моделирования сложных систем. В заключение главы дана последовательность разработки предварительного внешнего проекта, включающая описание процесса внешнего проектирования, проектирование взаимодействия с пользователем, подготовку и проверку правильности внешних спецификаций.

В шестой главе содержится обширный материал по проектированию архитектуры программных систем. Здесь изложены теоретические основы методологии проектирования ПС, модульно-интерфейсный подход и модульное программирование, вопросы оценки сложности программных систем, представление архитектуры программных систем на основе модульно-интерфейсного, объектно-ориентированного и компонентного подходов. Интересным и во многом оригинальным, с элементами формализации является материал по оценке сложности ПС и представлению многослойного программного продукта. Показан подход к формальному определению слоев программной системы. Большое внимание уделено

рассмотрению методов структурного проектирования ПС. Показано, что структурный подход в ряде случаев правомочен при разработке программных систем на основе объектно-ориентированной и компонентной методологии. В заключение главы дано описание методики разработки модульной (компонентной) архитектуры программной системы на основе формализованной схемы процесса проектирования, изложенной в четвертой главе. Этот материал отличается новизной и оригинальностью.

Седьмая глава посвящена изложению вопросов технико-экономического анализа проектов программных систем. Содержание главы отвечает на ряд важных вопросов такого анализа при планировании жизненного цикла программных систем. В целом это добротный материал, обобщающий результаты исследований последних лет, опубликованных в трудах ведущих специалистов. В материале главы рассматриваются вопросы первичного технико-экономического обоснование разработки ПС, методы оценки размеров проектов на основе размерно-ориентированных и функционально-ориентированных метрик, на основе LOC- и FP-метрик, оценка по конструктивной модели стоимости. Рассмотрены известные модели композиции приложения, раннего этапа проектирования, этапа постархитектуры. В заключение дана оценка комплексной стоимости программной системы со стороны заказчика и исполнителя проекта.

Интересной, на взгляд автора, является восьмая заключительная глава, посвященная актуальной в последнее время тематике рефакторинга программных систем. В ней рассматривается большой круг вопросов, связанных с рефакторингом объектно-ориентированных программных систем. Объясняется, что такое рефакторинг, и как он связан с производительностью и проектированием ПС, подробно излагаются ситуации, в которых следует применять рефакторинг, а также методы рефакторинга. Вводится понятие уровня рефакторинга и высшего его уровня – архитектурного рефакторинга. Определяются ситуации, в которых необходим архитектурный рефакторинг. В заключение рассматриваются вопросы построения архитектуры ПС по ее программному коду, рефакторинг архитектуры многослойной иерархической ПС, слои модулей (компонентов) в архитектуре ПС и паттерн выделения слоев. Эта наиболее интересная часть главы, в направлении которой предполагается дальнейшая работа автора.

Автор