

Тема 1. СЛАЙД Введение в программную инженерию

СЛАЙД План:

1. Обзор истории программной инженерии, её предмет
2. Место программной инженерии в процессе разработки и эксплуатации информационных систем

СЛАЙД 1. Обзор истории программной инженерии, её предмет

Рост масштабов и усложнение способов организации деятельности по созданию инженерных объектов, повышение степени ответственности за ее результаты; быстрое возрастание сложности возникающих при этом научных, технических и управленческих проблем привели к появлению в середине XX века новой прикладной программной методологии - программной инженерии.

СЛАЙД В современных разработках зарубежных специалистов программная инженерия рассматривается как комплексный, мультидисциплинарный подход и методика создания сложных систем и признается в качестве фундамента, на основе которого можно обеспечить и гарантированно поддерживать надежную и устойчивую связь между миссией, стратегическими целями, конкретными задачами и измеримыми результатами инженерной деятельности.



Недаром один из видных зарубежных специалистов по программной инженерии Дерек Хитчинс (Derek K.Hrtchins) назвал программную инженерию программной методологией XXI века.

В истории развития программной инженерии можно выделить два крупных этапа.

Этап становления, занятый формированием ядра методологии и основополагающих практик программной инженерии, начался на рубеже 40-х и 50-х годов и завершился к середине 90-х годов XX века. Основным результатом этого периода можно считать создание научно методических и нормативно-технических основ проектирования и разработки сложных инженерно-технических объектов.

За этапом становления последовал период, который можно назвать **этапом координации**. Он продолжается до сегодняшнего дня; его характерной особенностью является повышенное внимание к

увязке и гармонизации положений программной инженерии с достижениями и рекомендациями, полученными в сфере управления качеством, управления проектами, программной инженерии и в других областях.

На этой основе формируются комплексные, в основном программные, инструменты управления и поддержки инженерно-технической и инженерно-управленческой деятельности, пригодные для использования на протяжении полного жизненного цикла создаваемых систем.

Кроме того, на современном этапе развития программной инженерии в центре внимания программных инженеров оказываются не столько классические, в основном аппаратные системы с сосредоточенными параметрами, сколько распределенные системы, насыщенные разнообразным программным обеспечением, а также инженерия крупномасштабных программных систем и вопросы создания социотехнических систем и мегасистем.

В качестве важнейшей особенности первого из упомянутых этапов можно выделить сосредоточенность специалистов того времени на проблемах борьбы с постоянно нарастающей сложностью инженерно-технических объектов, создаваемых людьми, и, как следствие, выход на первый план вопросов совершенствования методологии и инструментов проектирования технических систем.

В 1957 году Г. Гуд (H. Good) и Р. Макол (Robert E. Machol) в своей работе определили **программную инженерию** как **системный метод проектирования технического оборудования**, а в качестве основной проблемы, стоящей перед инженерами, выделили сложность создаваемых систем.

Характеризуя методы проектирования сложных систем, авторы книги сделали акцент на использовании достижений математической науки, в частности математической статистики, дискретной математики и теории игр. При этом они подчеркивали, что сложность и разнообразие проблем, встающих при создании систем большого масштаба, требуют слаженной планомерной работы специалистов многих профилей, подключающихся к работе на разных ее этапах и выполняющих разные функции.

СЛАЙД В качестве **основных инструментов** достижения успеха при **создании крупномасштабных систем** они видели:

- согласование во времени решения программных, технологических и организационно-управленческих задач на основе сбалансированного и многоаспектного описания создаваемых систем;

- применение коллективных, командных методов работы;
- широкое использование электронных цифровых вычислительных систем для поддержки инженерной деятельности;
- применение комплексного подхода к проектированию с выделением внешнего (рассмотрение системы в целом в ее окружении) и внутреннего (моделирование и проектирование составных частей системы) проектирования.

Гуд и Макол, по-видимому, первыми предложили выполнять «макропроектирование» бригадой, состоящей из специально подготовленных программных инженеров, а также специалистов, обеспечивающих эффективное взаимодействие с другими участниками разработки, например с группами, выполняющими «внутреннее» проектирование элементов системы, с группами, ответственными за проведение испытаний, и с другими специалистами.

На этапе становления программная инженерия наряду с такими дисциплинами, как исследование операций и инженерная психология, рассматривалась специалистами в области программных исследований в качестве прикладной составляющей общей теории систем, где программной инженерии отводилась роль дисциплины, занятой научным планированием, проектированием, оценкой и конструированием систем «человек - машина».

В 1962 году А. Холл в своей книге «Опыт методологии для системотехники» сосредоточил внимание на целостном рассмотрении методологии **программной инженерии** и определил ее как **организованную творческую технологию**, выделив в качестве основных следующие положения: **СЛАЙД**

Первое: программная инженерия многоаспектна, и этот факт должен быть обязательно отражен при определении ее предмета.

Второе: в основу деятельности программного инженера должно быть положено понимание, что целью всего процесса программной инженерии является оптимальное проведение функциональных границ между человеческими интересами, системой и ее окружением. В самом же окружении выделяются три главных ее составных части: физическое и техническое окружение, деловое и экономическое окружение, социальное окружение.

Третье: программная инженерия уделяет первостепенное внимание исследованию потребностей, в основе которых должны лежать использование передовых экономических теорий, учет потребностей рынка и возможность изменения этих потребностей как сейчас, так и в

будущем.

Основы методологии программной инженерии, заложенные А. Холлом, остаются актуальными и сегодня. По мере развития методологии возникла потребность в разработке **рекомендации по практике применения рекомендаций программной инженерии**. Среди первых здесь можно выделить работы С. Шиннерса и Г. Чесгната.

СЛАЙД Шиннерс первоочередное внимание уделял пониманию проблемы, для решения которой создается система, и предлагал **семь общих взаимосвязанных** (включая обратные связи) **процедур**, которые должны быть неотъемлемой частью **инженерной деятельности по созданию систем**, а именно:

1. обоснование необходимости создания системы,
2. рассмотрение альтернативных решений,
3. выбор наиболее подходящей альтернативы,
4. синтез системы,
5. проверка соответствия,
6. сравнение требований и результатов испытаний и
7. корректировка характеристик оборудования и данных.

Эти рекомендации и сегодня широко используются в практике инженеров-разработчиков систем. В свою очередь **Г. Чесгнат** рассматривал **программную инженерию как комплексный подход, включающий всеобъемлющее рассмотрение различных методов достижения желаемого результата**.

При этом результат рассматривался как интегрированное целое, которое может включать ряд вспомогательных частей или функций. Выбор системы из серии решений этой многовариантной задачи, в которой характеристики составляющих оцениваются в терминах их вклада в оптимальный взвешенный результат всею целого, и составляли, по мнению Г. Чесгната, основу программной инженерии. Кроме того, развивая идеи программной инженерии о понимании проблемы и исследовании потребностей, Г. Чесгнат указывал на необходимость выявления требований к системе на основе всестороннего анализа потребностей всех категорий пользователей, заложив тем самым основы современной инженерии требований.

Следует отметить, что в своих работах Г. Чесгнат сосредоточился на проблемах проектирования систем и анализе их экономическом эффективности, а также на учете возникающих при этом неопределенностей; другие важные задачи создания сложных систем им по существу не рассматривались.

Начиная с конца 60-х годов быстро развиваются два основополагающих подхода программной инженерии - системный и жизненного цикла, которые и по сей день являются основой организации и осуществления деятельности по созданию сложных инженерных объектов. В частности, весной 1971 года в Калифорнийском технологическом институте (California Institute of Technology) была прочитана серия лекций под общим названием «Системные концепции для частного и государственного секторов».

Для чтения этих лекций были привлечены выдающиеся специалисты того времени, включая Ч. Черчмена (Charles West Churchman), Р. Макола (Robert E. Machol), Ф. Морса (Philip M. Morse), С. Рамо (Simon Ramo) и других известных ученых и практиков; позднее тексты этих лекций вошли в книгу Р. Майлса (Ralph F. Miles) «Системные концепции».

Можно считать, что начиная именно с этих лекций в качестве первоосновы инженерной деятельности по созданию сложных систем были признаны системный подход (systems approach) и системное мышление (systems thinking). Характеризуя в своей лекции системный подход, С. Рамо указывал, что этот подход является способом применения научного подхода к решению комплексных проблем в сфере инженерной деятельности и сосредотачивает внимание на анализе и проектировании системы в целом, а не ее компонентов или частей. Это предполагает, что при анализе возможных инженерно-технических решений во внимание принимаются все стороны и все имеющиеся возможности в отношении как социальных, так и технических аспектов проблемы. С. Рамо подчеркивал, что применение программного подхода к решению комплексных программных проблем требует от членов команды, создающей систему, большого объема знаний и способности к гармонизации подходов, сложившихся при решении программных проблем в рамках различных дисциплин. В целом специалисты согласились тогда с тем, что применительно к инженерной деятельности **системный подход включает шесть основных шагов:**

1. Определение цели или описание проблем.
2. Разработка требований и критериев.
3. Синтез программных решений.
4. Анализ программных решений.
5. Выбор системы.
6. Реализация системы.

С. Рамо указывал, что системный подход может дать хорошие результаты только в том случае, когда требования к системе ясно и недвусмысленно определены, а технологии и научные достижения, необходимые для их реализации, достаточно зрелы. Более подробно

представления С. Рамо о системном подходе и инженерной деятельности были в дальнейшем описаны в книге С. Рамо и Р. Сент-Клера, посвященной практике использования программного подхода при принятии сложных решений.

Одним из первых примеров целенаправленного использования программного подхода при создании сложного инженерного объекта стал проект «Аполлон» (одним из руководителей проекта был С. Рамо), для успешной реализации которого в качестве ключевых использовались четыре зрелых программных технологии - тяжелая ракета-носитель, корабль для перемещения в космическом пространстве, система траекторного анализа и измерений и система связи. Отметим, что настоящая книга содержит множество методических и практических рекомендаций по использованию программного подхода на разных этапах существования системы; кроме того, в ней имеется множество примеров и задач на эту тему. Собственно, шесть шагов, перечисленных выше, явились рамочной основой для определения содержания процесса программной инженерии, описанию и анализу различных особенностей которого в привязке к модели жизненного цикла посвящена большая часть книги.

Важным шагом на этапе становления программной инженерии стала разработка подхода жизненного цикла систем (system life cycle approach). Этот подход был определен в качестве фундаментальной основы успешной реализации процесса программной инженерии Б. Бланчардом (Benjamin Blanchard) и У. Фабрицким (Wolter Fabrycky) в 1981 году². Стадии и этапы жизненного цикла систем, выделенные авторами, были подобны фазам выбора системы, которые в 1962 году описал в своей книге А. Холл. Однако Б. Бланчард и У. Фабрицкий впервые обратили внимание на необходимость использования программными инженерами понятия жизненного цикла системы в качестве рамочной, организационной основы инженерного мышления, что, по мнению авторов, позволяет при создании сложных инженерных объектов рассматривать все системные аспекты в их полноте и взаимосвязи. Идея последовательного использования подхода жизненного цикла на практике является одним из стержневых положений, лежащих в основе представлений авторов настоящей книги о деятельности программного инженера. По существу все 15 глав этого учебника в той или иной мере позволяют читателю продвинуться на пути овладения методами и практиками подхода жизненного цикла в его современном понимании.

Следует отметить, что на всех стадиях своего развития программная инженерия уделяла большое внимание формированию и совершенствованию нормативного обеспечения деятельности по созданию систем. По-видимому, первым нормативным документом по программной инженерии стало опубликованное в 1966 году

Руководство 375-5. Оно было разработано при поддержке Военно-воздушных сил США (United States Air Force - USAF) и содержало описание основных деталей процесса разработки систем.

В дальнейшем это руководство было заменено на военный стандарт MIL-STD 499, который начиная с 1969 года непрерывно развивался до начала 90-х годов. Заключительная, но не принятая к практическому применению версия этого стандарта MIL-STD 499B, легла в основу таких важнейших современных стандартов программной инженерии, как ANSI/EIA 632, ISO/IEC 26702 и ISO/IEC 15288. В упомянутом военном стандарте содержались описание процессов разработки систем и рекомендации по управлению созданием системы с привязкой полученных результатов к плану управления программной инженерией (systems engineering management plan - SEMP).

В этом стандарте был впервые определен ряд **ключевых понятий**, вошедших, с небольшими изменениями, во все последующие нормативно-технические документы по программной инженерии:

1. **Система** - интегрированная совокупность персонала, продуктов и процессов, обладающая способностью к удовлетворению установленных потребностей или достижению целей.
2. **Элемент системы** - базовая структурная составляющая, входящая в состав системы и удовлетворяющая одному или нескольким требованиям, связанным с нижележащими уровнями функциональной архитектуры.
3. **Ключевые функции** - жизненно важные работы, способности или действия, которые должны быть реализованы для того, чтобы система могла гарантированно обеспечить удовлетворение потребностей клиента на протяжении полного жизненного цикла системы.

В дополнение к упомянутым стандартам армейское руководство США в 1979 году опубликовало боевой устав (Field Manual) 770-78. Данный документ содержал описание процесса разработки систем и представлял собой всестороннее, предназначенное для широкого использования руководство по практическому применению типового процесса разработки систем и управлению этим процессом.

Начиная с конца 80-х годов работу по формированию профессиональной среды программных инженеров, включая нормативно-техническое обеспечение деятельности, помимо военных специалистов стали активно вести представители гражданской

индустрии и академического сообщества. В 1989 году Альянс предприятий электронной промышленности (Electronic Industry Alliance - EIA) предложил свой стандарт программной инженерии EIA-632, в котором процесс программной инженерии рассматривался в качестве основы при создании систем, машин или оборудования, проектируемых в соответствии с требованиями пользователей в отношении функционирования системы. В работе над этим стандартом EIA активно сотрудничал с Ассоциацией предприятий аэрокосмической промышленности (Aerospace Industries Association - AIA). Вышло несколько редакций EIA-632, последняя из которых была разработана в сотрудничестве с Американским национальным институтом стандартов (American National Standards Institute - ANSI). В настоящее время версия этого стандарта поддерживается американской ассоциацией компаний по продвижению инноваций TechAmerica.

Работа по совершенствованию стандартов активно продолжается и на современной стадии развития программной инженерии. В настоящее время одним из главных ее результатов является развитый комплекс стандартов программной и программной инженерии, разработанный Международной организацией стандартизации (International Standard Organization - ISO) и Международной электротехнической комиссией (International Electrotechnical Commission - IEC) при участии ведущих мировых некоммерческих профессиональных организаций, таких как Институт инженеров электротехники и электроники (Institute of Electrical and Electronics Engineers - IEEE) и Группа по управлению объектами (Object Management Group - OMG). Среди нормативных документов, входящих в состав этого комплекса, выделяются стандарты процессов жизненного цикла систем, которые группируются вокруг стандарта ISO/IEC15288 «Программная и программная инженерия. Процессы жизненного цикла систем» и стандарта ISO/IEC 12207 «Программная и программная инженерия. Процессы жизненного цикла программного обеспечения». В состав этого комплекса входят также стандарты качества и зрелости процессов, стандарты описания архитектуры систем и ряд других документов.

Первый курс программной инженерии был прочитан в Массачусетском технологическом институте (Massachusetts Institute of Technology) в 1950 году тогдашним руководителем департамента программной инженерии корпорации Bell Labs Д. Гилменом (G. W. Gilman). В 1960 году профессор А. Уаймор (Albert Wayne Wymore) основал в университете Аризоны первую в мире кафедру программной инженерии, которая успешно работает до сегодняшнего дня. В настоящее время подготовку по программной инженерии в мире осуществляют около 250 университетов, среди которых примерно 60

европейских вузов, около 80 университетов из США и примерно 100 университетов из других стран мира. Сведения об образовательных программах по программной инженерии, реализуемых различными университетами, можно найти на сайте GradSchools.com

Повышение значимости для практикующих инженеров программной инженерии, включая системный подход и системное мышление, вызвало необходимость формирования новой академической среды, сосредоточенной на проблемах подготовки специалистов, способных создавать инженерные системы будущего.

Одним из шагов в направлении решения указанной задачи стало создание в 2004 году по инициативе Массачусетского технологического института Совета университетов, реализующих образовательные и исследовательские программы в области создания инженерно насыщенных систем (Council of Engineering Systems Universities - CESUN). Сегодня в этот Совет входит более 50 университетов из Северной Америки, Европы, Азии и Австралии; в центре их внимания находятся программы по созданию инженерно насыщенных систем, таких как транспортные, энергетические и коммуникационные системы. Члены CESUN полагают, что создание современных инженерно насыщенных систем - это область междисциплинарных исследований, где требуется по-новому учесть достижения технологий, а также управленческих и социальных наук. Для этого они сосредотачивают свои педагогические усилия в следующих областях:

- программная инженерия;
- технологии и стратегии;
- инженерный менеджмент, инновации и предпринимательство;
- системный анализ и принятие решений, исследование операций;
- проектирование и производство продукции, организация производства.

Основными партнерами CESUN при осуществлении профессиональной деятельности являются уже упоминавшийся Институт инженеров электротехники и электроники - крупнейшая в мире профессиональная организация по созданию и развитию передовых технологий, а также Международный совет по программной инженерии - INCOSE; кроме того, CESUN тесно сотрудничает с Институтом исследования операций и менеджмента (Institute for Operations Research and Management Science - INFORMS) и с

Институтом промышленных инженеров (Institute of Industrial Engineers - IIE).

В целом можно констатировать, что к концу первого десятилетия XXI века в мире сформировалась развитая и зрелая образовательная среда, в которой реализуются программы подготовки по программной инженерии различного уровня сложности. В эту среду включены бакалавриат, магистратура, а также система повышения квалификации и переподготовки кадров. Причем эта система постоянно совершенствуется и развивается при поддержке ведущих мировых компаний, занятых созданием сложных инженерных объектов.

При реализации образовательных программ по программной инженерии зарубежные университеты используют главным образом два сценария:

1. Программа фокусируется на программных проблемах создания сложных инженерных объектов - теория и практика программной инженерии становится в этом случае ядром всей образовательной программы, а ключевые разделы программной инженерии рассматриваются в отдельных, специально организованных курсах;
2. Программа фокусируется на проблемах инженерной деятельности в определенной предметной области (ИКТ, энергетика, транспорт и т. д.) - программная инженерия в этом случае является курсом, поддерживающим основную образовательную программу.

Принимая во внимание рекомендации международных экспертов, среди важнейших целей подготовки по программной инженерии можно выделить:

- **Владение подходом жизненного цикла**, включая способность на протяжении полного жизненного цикла (или на его отдельных этапах) успешно анализировать, проектировать или реализовывать пригодные к производству и использованию, эффективные, пригодные к сопровождению, экономически приемлемые комплексные системные решения применительно к продукции, услугам, предприятиям, а также к мегасистемам (системе систем). Это требование может быть адаптировано к конкретному типу или классу систем, с которыми придется иметь дело выпускнику, или к конкретной предметной области, например авиакосмической промышленности.
- **Профессионализм, включая способность к профессиональному развитию** на основе непрерывного обучения и активного участия в профессиональной деятельности, а также содействие развитию своей области профессиональной

деятельности. Кроме того, профессионализм подразумевает ответственное и этическое поведение на основе понимания общественной пользы;

- **Готовность использовать мультидисциплинарный подход**, включая способность успешно выполнять различные роли в мультидисциплинарных командах с различными формами членства, включая роль технического эксперта или роль руководителя на различных уровнях.

- **Коммуникабельность**, включая умение успешно общаться (читать, писать, говорить, слушать и иллюстрировать) устно и письменно, а также с использованием вновь появляющихся способов и средств коммуникации и массовой информации, особенно во взаимодействии с заинтересованными сторонами и с коллегами.

2. Место программной инженерии в процессе разработки и эксплуатации информационных систем

Назначение программной инженерии заключается в руководстве разработкой сложных, комплексных систем.

Система определяется как совокупность взаимосвязанных компонентов, которые работают совместно для достижения общей цели. Комплексная инженерно-насыщенная система:

- состоит из нескольких разнородных, разнотипных элементов, которые связаны между собой сложным образом
- требует применения программной инженерии для руководства разработкой.

Программная инженерия отличается от традиционных дисциплин тем, что:

- предметом ее рассмотрения является система в целом;
- ее интересуют потребности заказчика и условия эксплуатации;
- она направляет разработку концепции системы;
- она наводит мосты между традиционными инженерными дисциплинами и преодолевает непонимание между отдельными специалистами.

Программная инженерия является неотъемлемой частью руководства проектом, поскольку необходима для планирования и направления инженерных усилий.

Современная программная инженерия появилась в результате того, что передовые технологии в сочетании с ростом степени автоматизации принесли с собой риски и повышение сложности; при этом конкурентная борьба требовала идти на риск после тщательной оценки возможных последствий, а углубление специализации диктовало необходимость междисциплинарных связей и построения интерфейсов.

Примеры систем, нуждающихся в системном инженере

К числу инженерно-насыщенных комплексных систем относятся, в частности информационные системы:

- метеорологические спутники;
- системы управления воздушным движением в зоне аэропорта;
- системы слежения за грузовиками;
- системы бронирования авиабилетов;
- медицинские информационные системы;
- пассажирский самолет;
- современный уборочный комбайн;
- нефтеперерабатывающий завод;
- автосборочный завод;
- электростанция.

В настоящее время программная инженерия признана как профессия и играет все возрастающую роль в государственных учреждениях и промышленности. Существует специально созданная организация профессиональных программных инженеров - Международный совет по программной инженерии (INCOSE). Выпускники технических вузов имеют очень узкую специализацию. И лишь немногие интересуются задачами на стыке дисциплин; именно такие люди и становятся программными инженерами.

Профессия программного инженера трудна, но эти трудности окупаются. Типичный системный инженер с удовольствием берется за абстрактные, допускающие неоднозначное решение задачи, и наградой ему служит осознание своей ключевой роли в проекте. Следовательно, успешный системный инженер обладает следующими качествами:

- хорошо умеет решать задачи и приветствует сложные проблемы;
- имеет хорошую техническую подготовку и широкий кругозор;
- обладает аналитическим, системным складом ума и при этом проявляет творческие способности;
- прекрасно владеет навыками общения, прирожденный лидер.

На T-образной модели представлено сочетание опыта, образования, системы наставничества и глубины технических знаний, необходимое для становления успешного и влиятельного программного инженера.

В общем и целом, **программная инженерия - это дисциплина, обладающая высоким потенциалом, требующая мультидисциплинарных знаний и позволяющая агрегировать разнообразные системные элементы.** Системный инженер должен уметь выполнять приближенные расчеты применительно к сложным случаям, чтобы проверить результат на истинность. И наконец, он должен обладать скептическим, но вместе с тем позитивным складом ума - необходимое условие оправданного риска.