

## ПРОГРАММНАЯ ИНЖЕНЕРИЯ — НАУЧНАЯ И ИНЖЕНЕРНАЯ ДИСЦИПЛИНА

**Ключевые слова:** *программная инженерия, научная дисциплина, инженерная дисциплина, теория, методы программирования, программные объекты, ядро знаний SWEBOOK (Software Engineering Body of Knowledge) и PMBOOK (Project Management Body of Knowledge), базовый процесс, инфраструктура, стандарты.*

### ВВЕДЕНИЕ

Термин программная инженерия впервые произнесен на научной конференции НАТО в 1968 г. С тех пор на протяжении 40 лет это понятие постепенно менялось и развивалось в среде программирования. Программирование вначале считалось искусством одиночек, потом техникой и наукой, а теперь оно переходит к инженерной деятельности, приближающей процесс разработки к конвейерному производству по принципу сборки из готовых «деталей» типа reuse, application, asset provision и др. В качестве средств автоматизации производства программных продуктов выступают инструментально–технологические системы и среды.

В результате многолетней деятельности коллективов программистов накопилось большое количество программ, средств, а также знаний и опыта создания компьютерных программ. Эти знания отображены в программных продуктах массового применения и представлены множеством теоретических и прикладных методов, средств, принципов, правил, а также процессов производства систем с участием программистов и инженеров (плановиков, контролеров программных продуктов и процессов, тестовиков, инженеров по качеству и др.). В рамках деятельности теоретиков и практиков сформировались формальные методы доказательства, верификации и тестирования программ, математические модели надежности и методы оценивания показателей качества программных продуктов.

Все это обусловило необходимость систематизировать полученные знания и определить их в виде самостоятельной дисциплины (предмета) для понимания ее всеми компьютерными сообществами, отдельными программистами и инженерами программного обеспечения (ПО). Специально созданный комитет специалистов по информатике при ассоциации вычислительной техники ACM (Association for Computing Machinery) и IEEE Computer Society, компьютерное сообщество института инженеров по электротехнике и электронике IEEE (Institute of Electrical and Electronics Engineers), сформировали базовое ядро знаний по программной инженерии SWEBOOK (Software Engineering body of Knowledge), в котором в концентрированном виде представлено концептуальное содержание десяти базовых областей (knowledge areas) и дефиниция программной инженерии (ПИ) [1–4].

**Определение 1.** Программная инженерия (Software Engineering) — система методов, способов и дисциплины планирования, разработки, эксплуатации и сопровождения ПО, способная к массовому производству. Это инженерная сторона дисциплины, она охватывает все аспекты создания ПО, начиная от формулирования требований, разработки продукта, сопровождения и до его снятия с эксплуатации.

На наш взгляд, эта дефиниция SWEBOOK сужает сущность понятия программное обеспечение как объекта ПИ и сам предмет. Поэтому попытаемся определить программную инженерию и ее объекты в более широком смысле, привлекая аспекты и аргументы, которые характеризуют глубинные вопросы ПИ и позволяют определить ее как научную и инженерную дисциплину.



Рис. 1. Теоретический фундамент ПИ

помощью базовых понятий и принципов, которые приведены ниже по каждой из фундаментальных наук:

- в теории алгоритмов — нормальные алгоритмы, вычислительные функции, машина Тьюринга, алгоритмические алгебры, графы-схемы, модели алгоритмов и программ и т.п.;
- в математической логике — правила логических исчислений и высказываний, способствующих формальному определению правильных умозаключений, а также логико-алгебраические спецификации программ;
- в теории доказательства — математический вывод теорем и программ с помощью аксиом и утверждений, теория непротиворечивости и алгоритмически неразрешимых проблем, теория верификации программ (VDM, RAISE, Z, методы Хора, Дейкстры и др.);
- теория множеств — кванторы всеобщности, существования и операции над множествами, которые применяются для формального представления аксиом для разных совокупностей программных объектов;
- в теории управления — принципы, методы и общие законы планирования и управления в целях инженерного получения и обработки информации в любых кибернетических и управленческих системах.

Кроме этого фундамента, к системе знаний ПИ относятся [5, 6]:

- формальные методы программирования — спецификация программ, их доказательство, верификация и тестирование, а также математические модели надежности и т.п.;
- прикладные методы, а именно: средства, принципы, правила и процессы жизненного цикла (ЖЦ) производства компьютерных систем как инструментов коллективной разработки больших программных проектов;
- методы управления коллективами, а именно: планирование сетевых графиков, контроль работ на процессах ЖЦ, измерение и оценивание качества разработки промежуточного и конечного продуктов производства, регулирование сроков и стоимости их изготовления и сертификации.

ПИ входит в состав компьютерной науки (Computer science), преемницы науки программирования включила все ее накопленные теоретические и прикладные достижения, развила новые методы управления коллективным трудом разработчиков и методы измерения и оценивания результатов их деятельности. Она сформировалась как научно-инженерная дисциплина. Поэтому приведенное определение 1 есть узким как в смысле предмета разработки ПО, так и самой ее дефиниции. Дадим определение 2, которое более широкое и способствует раскрытию смысла научной и инженерной части ПИ.

Известно, что любая наука — это система проверенных практикой знаний, которые отображают ее общие вопросы, понятия и закономерности их развития. Она устанавливает связи с другими науками и влияет на их развитие. Относительно ПИ можно сказать, что она интегрировала в себе принципы математики и основные положения таких фундаментальных наук: теория алгоритмов, математическая логика, теория управления, теория множеств, доказательств и др. (рис. 1).

Эти науки создают теоретический фундамент программной инженерии, необходимый для построения абстракций программ с

**Определение 2.** Программная инженерия — это система методов и средств программирования, инженерии планирования и управления коллективным процессом производства компьютерных программных систем (ПО, приложений, семейств систем, программных проектов), методы измерения и оценивания различных их характеристик на соответствие требованиям заказчика. Производство ориентировано на использование базовых объектов (модулей, компонентов, аспектов, сервисов и т.п.) и автоматизированные операции, близкие к конвейерному производству.

При данном толковании ПИ имеется в виду сложившаяся теория программирования и образовавшаяся инженерия управления изготовлением программных продуктов (ПП) путем адаптации к задачам изготовления ПП общих методов управления (критического пути, PERT, план-графиков), планирования и распределения работ между исполнителями проекта, оценки вложенных ими трудозатрат и достигнутого качества. Поэтому естественно программную инженерию рассматривать как научную и инженерную дисциплины изготовления программных продуктов (рис. 2).

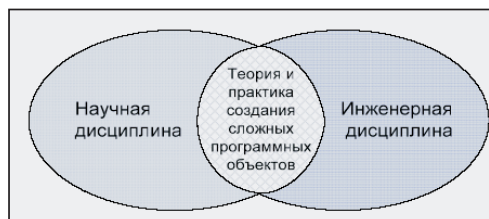


Рис. 2. Научная и инженерная дисциплины ПИ

На пересечении областей дисциплин (овалов на рис. 2) находится теория и практика построения сложных программных объектов. Теория построения — это теоретическое программирование с помощью средств аппарата абстрактных спецификаций (графовых и структурных схем, функций и композиций, дескрипторов и номинаций данных, use case-диаграмм и др.), а также формальная проверка правильности спецификаций методами доказательства, верификации и оценки надежности. Практика построения — это применение теоретических и системных методов программирования на процессах проверки (верификация, валидация, тестирование) спецификаций объектов, последовательного их преобразования к выходному коду и инженерия оценивания и сертификации разных показателей качества (надежность, производительность, эффективность и т.п.) ПП.

### ПРОГРАММНАЯ ИНЖЕНЕРИЯ КАК НАУЧНАЯ ДИСЦИПЛИНА

В отличие от математической или другой фундаментальной науки, целью которых есть получение новых знаний для решения соответствующих задач, в ПИ знания — это общая теория построения программ для компьютеров, ориентированная на изготовление продукта для получения от него результата.

Наука ПИ — это теоретические, формальные методы и средства построения сложных программных объектов. Построение — это анализ предметной области, проектирование и образование выходного кода для его выполнения на компьютере. Интегрированные в программную инженерию фундаментальные науки, которые приведены на рис. 1, а также наука программирования образуют ее общий теоретический фундамент, который предоставляет базовые понятия, объекты и формальные механизмы, необходимые при создании общих и специфических особенностей программных продуктов в соответствии с заданными к нему требованиями. Как наука ПИ включает:

- 1) основные понятия и объекты;
- 2) теорию программирования и методы управления изготовлением ПП;
- 3) средства и инструменты процессов разработки продукта.

**1. Основные понятия ПИ** — это данные и их структуры, функции и композиции, базовые объекты (модуль, компонент, каркас, повторно используемый компонент (ПИК) и др.) и целевые объекты (ПО, программная система, семейство систем, программный проект, сложные программные системы).

Простые объекты разрабатываются с помощью элементарных операций формального их описания, а изготовление из них целевых объектов осуществляется применением инженерных методов управления коллективом, их работами, сроками и стоимостью.

Приведем определение целевых объектов ПИ [3, 6, 7].

**Программная система** (ПС, Application) — комплекс интегрированных программ и средств, которые реализуют набор функций некоторой предметной области (ПрО) в заданной среде. В комплекс могут входить: ПО, прикладные системы (зарплата, учет и др.), общесистемные компоненты (транслятор, редактор, система управления базами данных (СУБД) и т.п.), системы защиты, обеспечения безопасности функционирования и др. Способ изготовления — инженерия ПС (application engineering), которая включает процессы ЖЦ, методы разработки, управления, оценивания продуктов и процессов для их возможного усовершенствования.

**Программное обеспечение** — совокупность программных средств, которые реализуют функции компьютерной системы (или функции аппаратно-программной системы), включая общесистемные средства (например, операционная система (ОС), СУБД, системы защиты и т.п.), построенные подсистемы контроля (показателей технологических процессов, обработки сигналов) и прикладные программные системы. Так, функции некоторой ОС — это управление задачами, программами, данными и т.п. ПО может входить в состав ПС или быть идентичным функциям программной системы. Способ изготовления — инженерия реализации задач ПО.

**Семейство систем** (systems family) — совокупность ПС с общим (неизменным для всех членов семейства) и управляемым изменяемым набором ее характеристик, которые удовлетворяют определенным потребностям прикладной области (домену). Способ изготовления — инженерия домена (Domain Engineering) или конвейерное производство однотипных ПП с единой схемой на основе каркаса и специально разработанных отдельных членов семейства и других готовых программных ресурсов с помощью базового процесса или линейки продукта (product line).

**Программный проект** — уникальный и интегрированный продукт, в котором отображены совокупности реализованных целей, задач и результатов деятельности, удовлетворяющих требованиям заказчика проекта. Способ изготовления — инженерия разработки и менеджмента проекта.

**Сложные программные объекты** — совокупность взаимосвязанных целевых объектов разных типов, которые выполняют необходимые функции в такой системе, представленные вновь разработанными простыми объектами или выбранными с репозитория готовых ресурсов ПИ.

**2. Теория программирования** — это множество методов, языков и средств описания (спецификации) и проектирования целевых объектов, а также методов их доказательства, верификации и тестирования. Кроме того, в ПИ привлечены формальные методы управления проектом (персоналом, материальными и финансовыми ресурсами) и отдельными его показателями [6–8]. Методы программирования в ПИ представлены на рис. 3 и включают:

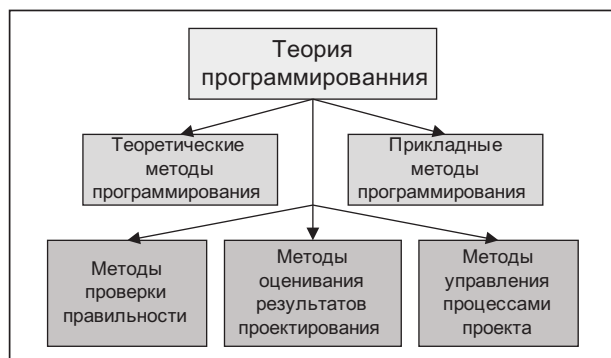


Рис. 3. Совокупность методов ПИ

- методы теоретические (алгебраический, алгоритмический, VDM, RAISE и др.) и прикладные (объектный, компонентный, аспектный и др.), предназначенные для проектирования разных типов целевых объектов;

- методы проверки правильности программ формальными процедурами (утверждения, вывода, доказательства);

- методы оценивания результатов последовательного проектирования (промежуточных рабочих продуктов) на процессах проекта и оценки показателей качества конечного продукта (надежности, точности, производительности и т.п.);

- методы управления планами работ и контроля промежуточных результатов на процессах проекта, а также дополнительные расчетные методы оценки трудозатрат на каждую работу, ее стоимость и др.

**3. Языки, средства и инструменты ПИ.** Все объекты, как артефакты ПП, включают разного рода описания: требования к разработке, согласованные с заказчиком, архитектуру, структуры данных, спецификации программ и т.п.

Проектирование целевых объектов осуществляется с помощью современных языков, в том числе визуальных (например, C++, Java, Pascal, UML и др.), и соответствующих инструментальных сред, которые содержат необходимые языковые преобразователи и инструменты поддержки артефактов процесса разработки. К таким средствам также относятся шаблоны, каркасы, диаграммы потоков данных, классов, поведения и т.п.

Проверка правильности целевых объектов осуществляется методами программирования и соответствующими инструментами, приспособленными для разработки разных задач проекта в соответствующей среде проектирования. Готовый продукт проверяется на соответствие реализованных функций заданным требованиям, тестируется с помощью специальных методик, а также подвергается измерению и оцениванию для получения показателей качества ПП.

В роли сред проектирования целевых объектов используются передовые, современные технологии и соответствующие пакеты инструментально-технологических систем (например, Microsoft Visual Studio, MSF, RUP, Rational Rose и др.). Их инструменты поддерживают проектирование разных типов целевых объектов, а также управление менеджментом проекта, в частности персоналом, планами и качеством продуктов. Эти средства и инструменты удобно использовать в коллективной разработке проекта.

#### ПРОГРАММНАЯ ИНЖЕНЕРИЯ КАК ИНЖЕНЕРНАЯ ДИСЦИПЛИНА

Инженерная дисциплина (или инженерия) в ПИ — это способы выполнения деятельности, связанной с изготовлением ПП для разных видов целевых объектов с применением методов, средств и инструментов научной дисциплины ПИ [6–8]. Ее основу составляют следующие базовые элементы (рис. 4):

1) ядро знаний SWEBOK ([www.swebok.com](http://www.swebok.com)), как набор теоретических концепций и формальных правил, которые могут применяться в ПИ;

2) базовый процесс ПИ, как стержень процессной деятельности в организации-разработчике ПП;

3) стандарты, как набор регламентированных правил конструирования промежуточных артефактов на процессах ЖЦ;

4) инфраструктура — это условия среды и методического обеспечения базового процесса ПИ и исполнителей, которые занимаются изготовлением ПП;

5) менеджмент проекта ([www.pmi.com](http://www.pmi.com)) — это стандартные положения и процессы, а также научные принципы, методы планирования и контроля работ проекта.

С инженерной точки зрения, в ПИ реализуются задачи изготовления ПП, как технологические процессы формирования требований, проектирования и сопровождения продукта, а также проверки операций базового процесса на правильность реализации разных функциональных задач проекта и вкладывания его работ в заданный заказчиком срок.

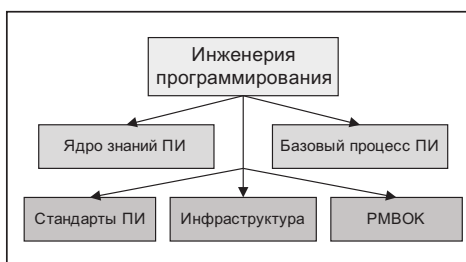


Рис. 4. Состав базовых элементов инженерной дисциплины

ПИ можно рассматривать с двух точек зрения:

— как инженерную деятельность, в которой инженеры разных категорий осуществляют работы проекта, учитывают соответствующие теоретические методы и средства ПИ, которые рекомендованы в ядре знаний SWEBOK, а также положения процессов ЖЦ стандартов и избранных для процессов методов;

— как систему управления проектом, качеством и рисками с привлечением правил и положений стандартов ЖЦ, качества и менеджмента проекта [7, 8].

Инженерная деятельность планируется, в ней проводится распределение проекта на отдельные работы, которые будут выполнять исполнители проекта. Менеджер проекта — это главное действующее лицо, ответственный за проектирование и контроль выполнения этих работ разными службами инфраструктуры проекта в организации (верификации, тестирования, оценки качества и др.). Продукт коллективного изготовления передается заказчику для сопровождения. Найденные в нем ошибки и недостатки устраняются разработчиками.

Эта деятельность практически уже отработана и по своей сути близка к инженерной деятельности в промышленности, где инженерия — это способ применения научных результатов в изготовлении некоторых технических изделий с помощью технологических правил и процедур, методик измерения, оценки и сертификации изготовленного продукта.

Далее дается общая характеристика базовых 1)–5) элементов инженерной дисциплины (см. рис. 4).

**1. Ядро знаний SWEBOK** — это главная содержательная трактовка и перечень концептуальных основ разделов ПИ. Структурно ядро включает в себя 10 разделов (knowledge areas). Их условно поделим на две категории: проектирование и организационная, инженерная деятельность. Первая категория — это методы и средства разработки (формирование требований, проектирование, конструирование, тестирование, сопровождение). Вторая категория включает методы управления проектом, конфигурацией, качеством и базовым процессом организации-разработчика.

Методы ядра знаний SWEBOK менеджер проекта сопоставляет с задачами стандартных процессов ЖЦ и обеспечивает наполнение базового процесса этими методами. Тем самым создается базис инженерной дисциплины изготовления продукта, включающий регламентированную последовательность процессов разработки и сопровождения ПП. На начальных процессах определяются требования к продукту, проектные решения и каркас (абстрактная архитектура) будущего продукта. На основе требований и каркаса разрабатываются новые или подбираются готовые объекты для «заполнения» каркаса конкретным содержанием и последующей его доработкой до конечного ПП.

**2. Базовый процесс (БП)** — это метуровень для процессирования изготовлением продукта. Он содержит основные понятия, которые относятся к оснастке, организационной структуре коллектива разработчиков и методологии оценки, измерения, управления изменениями и совершенствованием самого процесса. В целом БП — это множество логически связанных с ним видов инженерной деятельности организации-разработчика и набор средств и инструментов, предназначенных для использования при изготовлении ПП.

**3. Инфраструктура** — это набор технических, технологических, программных (методических) и человеческих ресурсов организации-разработчика, необходимых для выполнения БП, ориентированного на выполнение договора с заказчиком программного проекта. К техническим ресурсам проекта относятся: компьютеры, устройства (принтеры, сканеры и т.п.), серверы и т.п., а к программным — ПО, информационное и системное обеспечение. Технологические и методические ресурсы — это методики, процедуры, правила, рекомендации стандартов для процесса и управления персоналом. Человеческие ресурсы — это группы разработчиков и служб управления проектом, планами, качеством, риском, конфигурацией и про-

верки правильности выполнения проекта разработчиками. Вместе это базовый процесс и комплект документов по регламентации, выполнению и регулированию процессов ЖЦ под конкретные цели проекта [7]. Средства, промежуточные результаты разработки на процессах ЖЦ, а также методики руководства ресурсами БП и результаты применения соответствующих методов программирования сохраняются в базе знаний проекта (рис. 5).

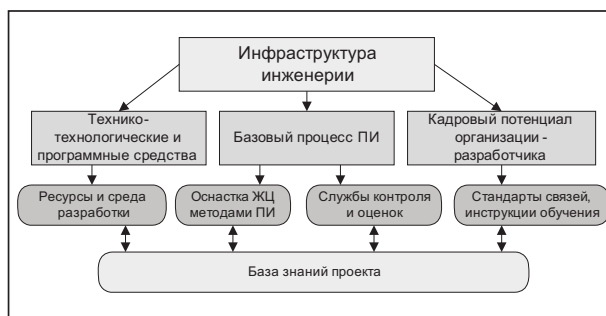


Рис. 5. Элементы инфраструктуры проекта

После выполнения проекта и приобретения опыта построения конкретного проекта базовый процесс и его элементы могут совершенствоваться (доработка, замена, добавление новых средств) соответственно стандарту ISO/IEC TR 15504 (Information Technology — Software Process Assessment, Part 2: A reference model for processes and process capability).

Готовность всех видов элементов процесса и персонала организации-разработчика производить качественный продукт — основа оценки зрелости как самой организации, так и выпущенного продукта. Для этого применяется модель зрелости CMM (Capability Maturity Model) Института программной инженерии SEI США. Для рассмотрения разной степени готовности организации создавать ПП модель рекомендует следующие оценки: удовлетворительно, средне, хорошо и очень хорошо. Уровень степени готовности определяется наличием в организации базового процесса, всех необходимых для него ресурсов, соответствующих стандартов и методик, а также профессиональных способностей (зрелости) членов коллектива изготавливать ПП в заданный срок и вкладываться в стоимость. Модель CMM — пятиуровневая. Первый и второй уровни определяют недостаточную подготовленность организации выполнять зрело продукт. Третий, четвертый и пятый уровни характеризуют степень готовности, зрелости и профессионализма специалистов организации изготавливать соответственно средний, хороший и отличный продукт. Кроме базовой модели, используются специализированная модель SW CMM (Software CMM) для оценки зрелости ПО, интегрированная модель зрелости CMMI (CMM Integrated) для учета потребностей государственных структур в ПО, Bootstrap — для оценки зрелости маленьких и средних коммерческих компаний.

**4. Стандарты ПИ** — это технологически отработанный набор разных видов процессов со строго определенным и регламентированным порядком проведения работ в ПИ, связанных с разработкой и оцениванием продукта на качество, риски и т.п. Стандарты в области ПИ регламентируют разные направления деятельности. Они стандартизируют термины и понятия, ЖЦ, качество, измерение, оценку продуктов и процессов. Наиболее важными среди них — стандарт ISO/IEC 12207 «Процессы жизненного цикла программного обеспечения», а также стандарты серии ISO/IEC 9000 — 1, 2, 3, ДСТУ 2844–94 и 2850–94, регламентирующие аспекты системы управления и обеспечения качества ПП.

Стандарт ISO/IEC 12207 содержит организационные процессы: планирование, управление и сопровождение. Процесс планирования включает действия по составлению планов и графиков проекта, по распределению работ между разными категориями специалистов, а также действия по контролю планов и выполненным работ. Процесс управления проектом определяет процессы управления работами на проекте, которые должны выполнять специалисты, владеющие теорией управления, способные планировать и следить за сроками выполнения проекта. Третий процесс — это сопровождение готового продукта проекта, выявление и устранение

найденных в нем недостатков, а также дополнение новых или удаление некоторых функций в продукте.

Ядро знаний SWEBOOK и стандарты ЖЦ взаимосвязаны. Они используются при определении структуры базового процесса ПИ и разработки необходимых методик и ограничений на изготовление продукта. При этом могут использоваться фундаментальные модели ЖЦ (водопадная, спиральная и т.п.), которые реализуют заложенный в них стиль проектирования и реализации для некоторых видов продуктов.

**5. Менеджмент проекта** — это управление разработкой проекта с использованием адаптированной теории управления и процессов ядра знаний PMBOK к типу проекта. В PMBOK представлены положения и правила руководства в соответствии с заданным временем производственного цикла построения уникального продукта в рамках некоторого проекта [9]. PMBOK — это стандарт IEEE Std.1490:2003, который разработан американским Институтом управления проектами ([www.pmi.org](http://www.pmi.org)) для создания ПО.

Ядро знаний PMBOK построено аналогично ядру знаний SWEBOOK и содержит описание лексики, структуры процессов и областей знаний, которые отображают современную практику управления проектами в некоторых областях промышленности и ПО в частности. Ядро содержит процессы и определяет принятие решений по ресурсам проекта в каждый момент времени выполнения и управления задачами проекта. Область знаний — управление содержанием проекта — это процессы, регламентирующие планирование, разбивку работ на более простые в целях упрощения процесса управления. Область управления качеством содержит процессы и операции достижения целей проекта, а также правила и процедуры для улучшения процесса обеспечения качества в соответствии с заданными требованиями заказчика и положениями стандарта качества (ISO/IEC 9000 –1, 2, 3). Область управления человеческими ресурсами включает распределение работ между исполнителями в соответствии с их квалификацией и профессионализмом.

Таким образом, стандарт PMBOK, как и стандарт ISO/IEC 12207, имеет много общего, касающегося организационных процессов управления проектом и его ресурсами.

#### **ПРОГРАММНАЯ ИНЖЕНЕРИЯ В ПРАКТИЧЕСКОМ ИЗМЕРЕНИИ**

Общее назначение ПИ — это практика, т.е. конкретное построение компьютерных программ, систем и инструментов с применением теоретических и инженерных методов.

Главная особенность практики — это использование разработанных готовых программ и информационных ресурсов Интернет (MatLab, Greenstone, Grid-системы и др.). Доступ к ним может осуществить любой пользователь и получить бесплатно или на коммерческой основе готовый ресурс, как сервис. Он может однократно использоваться для решения соответствующей задачи либо как некоторая программа постоянного и многократного применения в некотором домене. Сегодня сформировалось три инженерных подхода по применению готовых ресурсов: reusing engineering, application engineering, domain engineering. Они применяют в качестве готовых ресурсов повторно используемые компоненты (ПИК), приложения и системы. Применение готовых ресурсов, как многократно использованного готового продукта, дает значительную экономию при производстве из них новых программных систем и семейств систем. Все виды ПИК сохраняются в хранилищах проекта — репозиториях [6, 7].

Инженерия ПИК — это систематическая и целенаправленная деятельность для подбора реализованных и представленных в репозитории ПИК. Система проектируется из них снизу вверх. Сначала создается общая структура — каркас продукта, дается его описание и по этому описанию готовые компоненты и ПИК интегрируются в систему.



Инженерия приложений базируется на многоразовом использовании ПИК, а также готовых программ. Проектирование одиночных, т.е. уникальных программ-приложений — это инженерия программирования из готовых ПИК приложений. Процесс построения начинается с анализа ПрО, определения концептуальной модели, разработки проектных решений и проведения композиции компонентов с использованием шаблонов или каркасов.

Инженерия ПрО — набор средств, инструментов, множества готовых ресурсов и базового процесса построения из них систем семейства (домена) на основе модели, содержащей общие и изменяемые характеристики представителей семейства. Выбранные ПИК или одиночные приложения встраиваются в модель домена. Технология разработки семейства планируется как в плане создания модели домена, так и членов семейства. Она включает процессы анализа, проектирования и встраивания ПИК в отдельные представители семейства. По своему характеру данная инженерия приближена к конвейерному производству продуктов из готовых ресурсов. Управление этой инженерией состоит в планировании и распределении работ для каждого участника процесса создания домена и контроля их выполнения.

Технологические приемы производства программных продуктов из готовых компонентов и программных систем воплощены в так называемые линейки продуктов, которые соответствуют конвейерному производству ПП на основе спроса рынка.

Практические инструменты производства — ядро знаний SWEBOOK, PMBOOK и стандарты, в которых представлен регламент описания процесса инженерии доменов (Domain engineering process) как нового процесса ЖЦ. Согласно этому стандарту процесс инженерии доменов включает: анализ домена (выявление связей, постоянных и изменяемых требований, понятий и моделей); проектирование домена (Domain design) — каркаса для ПИК, активов (asset) и интерфейсов, согласованных с моделью домена; технологию домена с подпроцессами формирования ресурсов (asset provision), разработкой базы ресурсов (asset-based development) и их сопровождения. В результате применения технологии инженерии домена в софтверной организации будет создаваться архитектурный базис производства ПП из многоразовых ПИК репозитория и языков спецификации доменов.

Таким образом, наука и инженерия, SWEBOOK, СТАНДАРТЫ, PMBOOK, как главные составные элементы программной инженерии, связаны между собой процессами ЖЦ, методами проектирования и управления разработкой проекта. Символически эта взаимосвязь представлена на рис. 7.

Рассмотренные базовые теории и составные элементы ПИ ориентированы на разработку прикладных систем разного назначения. При их разработке необходимы инструментально-технологические системы и среды с набором необходимых инструментов. Пример такой среды и реализации в ней инженерного подхода к проектированию — система Visual Studio Teams Systems фирмы Microsoft. В ней реализована технология проектирования, кодирования, тестирования и внедрения программных продуктов. В этой среде определена технология подбора, распределения и выполнения работ между разными группами специалистов, имеющими разные уровни знаний и навыков в области программирования. Они разделены на четыре категории специалистов, в которой каждый получает работу по своим способностям. Переход в более квалифицированную группу зависит от качества выполненных работ предыдущей задачи и от полученного опыта. Четвертая группа — это специалисты высокой квалификации, они отвечают за правильность разработки проекта в данной среде.

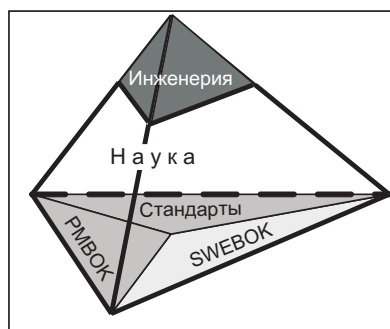


Рис. 7. Составные элементы ПИ

## ЗАКЛЮЧЕНИЕ

В настоящей работе представлен системный анализ ПИ в научном, инженерном и практическом измерении. Дано ее общее толкование, приведены аргументы и определения как объектов, так и самой дефиниции. Проведена ее структуризация в виде научной, инженерной и практической дисциплин, сформулированы содержание и их назначение. Предложен инженерный и практический аспекты построения целевых программных объектов (программного обеспечения, приложений и доменов) из готовых ПИК. Показана роль базового процесса ПИ, как процессной основы разработки компьютерных программ с использованием ядра знаний, стандартов ЖЦ, инфраструктуры и менеджмента в организации-разработчике.

Рассмотренный подход к определению ПИ обсуждался, дискутировался и уточнялся в отделе Института программных систем, на Всеукраинской конференции преподавателей и студентов «Программная инженерия — направления обучения», проведенной Национальным авиационным университетом (3–5 декабря 2007 г.), на семинаре преподавателей и студентов (6 декабря 2007 г.), чтении курса лекций студентам на факультете кибернетики Киевского национального университета имени Тараса Шевченко и в филиале МФТИ при Институте кибернетики имени Глушкова НАН Украины. Автор выражает благодарность всем, кто принял участие в обсуждении предмета ПИ.

## СПИСОК ЛИТЕРАТУРЫ

1. Jacobson I. Object-oriented Software Engineering. A use case Driven Approach, Revised Printing. — New York: Addison Wesley, Publ. Co, 1994. — 529 p.
2. Андон Ф.И., Лаврищева Е.М. Методы инженерии распределенных компьютерных приложений. — Киев.: Наук. думка, 1998. — 228 с.
3. Бабенко Л.П., Лаврищева К.М. Основи програмної інженерії. — Київ: Знання, 2001. — 269 с.
4. Сомервилл И. Инженерия программного обеспечения. — М.; Санкт-Петербург; Киев: Изд. дом «Вильямс», 2002. — 623 с.
5. Грищенко В.Н., Лаврищева Е.М. Методы и средства компонентного програмування // Кибернетика и системный анализ. — 2003. — № 1. — С. 39–55.
6. Лаврищева Е.М. Методы программирования. Теория, инженерия, практика. — Киев: Наук. думка, 2006. — 450 с.
7. Основы инженерии качества программных систем / Ф.И. Андон, Г.И. Коваль, Т.М. Коротун, Е.М. Лаврищева, В.Ю. Сулов: 2-е изд. — Киев: Академперіодика, 2007. — 672 с.
8. Лаврищева Е.М., Коваль Г.И., Коротун Т.М. Подход к управлению качеством программных систем обработки данных // Кибернетика и системный анализ. — 2006. — № 5. — С. 174–185.
9. Задорожна Н.Т., Лаврищева К.М. Менеджмент документообігу в інформаційних системах освіти. — Київ: Пед. думка, 2007. — 220 с.

*Поступила 15.02.2007*