

Разработка и использование компьютерных программ в настоящее время стали массовой деятельностью. Более семи миллионов человек занимаются их разработкой, а сотни миллионов активно их используют [1.1]. Практически нет ни одной сферы деятельности человека (экономика, медицина, бизнес, коммерция, промышленность и т. д.), где бы *ПО* не использовалось, как средство автоматизации и улучшения *работ*. Спрос на него постоянно увеличивается, сложность растет, а количество ошибок не уменьшается.

Знания разработчиков *ПО* отличаются разнообразием и, как правило, они являются неполными, несогласованными и разнородными, ориентированными на реализацию разных предметных областей, начиная от ОС и заканчивая прикладными бизнес-системами. И самое главное, знания в процессе инженерной деятельности постепенно уточняются, видоизменяются и пополняются, и их необходимо учитывать разработчикам нового *ПО*.

Примерно каждые 10 лет происходит смена языков программирования и операционных сред для описания и функционирования программ, что предполагает перевод ранее изготовленных и функционирующих программ на новые языки и операционные среды. На это тратятся огромные людские и финансовые ресурсы. Так, в 2000 году в изменении формата даты в программах и микросхемах на десятках млн. компьютеров участвовало более 2 миллионов программистов, а *затраты* составили сотни миллионов долларов. Перевод ранее созданных прикладных Фортран-программ на новые языки (C, Java и др.), и *развертывание* их в новых операционных средах требует больших капиталовложений и привлечения огромной армии программистов.

В связи с постоянным обновлением персональных компьютеров, соответственно, операционных сред и систем программирования, возникают сложности, связанные с адаптацией действующих программ и прикладных систем к новым условиям. В практике программирования не раз делались попытки облегчить труд *по* написанию программ, перейдя к "программированию без программистов". В связи с этим появлялись программные проекты, которые ставили своей целью заменить постановки задач математическим описанием и заставить машины их обрабатывать. К ним относятся компьютеризация математических знаний (машины серии "Мир", японский проект "IBM 5-го поколения" [1.2]), фабрики программ *по* изготовлению программной продукции *по* типу сборочного конвейера из готовых "деталей"-программ (завод для сборки АСУ, система АПРОП, ПРИЗ) и многие другие проектные решения, направленные на изменение стилей программирования. Например, формальные спецификации и математическое *доказательство* правильности программ (1980 г.), систематизация знаний в области инженерии программного обеспечения и создание общего ядра знаний SWEBOK (2001, 2003гг.), теория построения и верификации программ (проект 2005 г.) и др.

Переворот в программировании *по* высвобождению армии пользователей компьютеров от разработки и написания программ так и не произошел, наоборот, методы и средства программирования постоянно развиваются. Особенно это связано с новыми возможностями платформ и распределенных сред, в которых компоненты располагаются *по* разным узлам сети и взаимодействуют между собой через сетевые протоколы. Кроме того, появились новые методы и подходы к разработке *ПО*: структурный, объектно-ориентированный, компонентный, аспектный, визуальный, агентно-ориентированный, сервисный и др. [1.3–1.12].

Для поддержки новых методов разработано огромное количество разнообразных инструментальных средств и методов оценки качества, производительности, стоимости и т.п. Процесс разработки *ПО* и методы оценивания продуктов стандартизованы (ISO/IEC 12207, 15504, 9126 и др.) [1.14, 1.16]. Все это способствует повышению эффективности проектирования, тестирования, прогнозирования надежности и оценки качества *ПО*.

Новый программный проект разрабатывается 1–2 года, а эволюционирует 6–7 лет. На сопровождение проекта тратится 61% против 39% средств на его разработку. Эффективность разработчиков в зависимости от квалификации колеблется в отношении 1:10, а значит, требуется повышать уровень знаний разработчиков *ПО*. На сегодня *ядро* стабильных знаний *по* программной инженерии составляет 75% от тех знаний, которыми пользуются в практической деятельности. В связи с этим проведена систематизация накопленных знаний в программировании и ряде других областей информатики. Международным комитетом при американском объединении компьютерных специалистов ACM (Association for Computing Machinery) и институте инженеров *по* электронике и электротехнике IEEE Computer Society было создано *ядро* знаний SWEBOK. В этом ядре были систематизированы разнородные знания в области программирования, планирования и управления, сформулировано понятие программной инженерии и десяти областей, которые соответствуют процессам проектирования *ПО* и методам их поддержки.

**Программная инженерия (Software Engineering)** является отраслью информатики, которая изучает вопросы построения компьютерных программ, отражает закономерности развития программирования, обобщает *опыт* программирования в виде комплекса знаний и правил регламентации инженерной деятельности разработчиков *ПО*.

В этом определении выделим два основных аспекта.

1. Программную инженерию можно рассматривать как инженерную дисциплину, в которой инженеры применяют теоретические идеи, методы и средства при разработке ПО, создают продукты в соответствии со стандартами, регламентирующими процессы их проектирования и разработки.
2. Программная инженерия описывает методы управления программным проектом, качеством и рисками. Применение таких методов позволяет достичь высокого качества программных продуктов. Эта инженерная дисциплина предоставляет всю необходимую информацию и стандарты для выбора наиболее подходящего метода и процессов жизненного цикла ПО для реализации конкретного проекта.

Как инженерная дисциплина, она охватывает все аспекты создания ПО, начиная от формирования требований до создания сопровождения и снятия с эксплуатации ПО, а также включает инженерные методы оценки трудозатрат, стоимости, производительности и качества. Т.е. речь идет именно об инженерной деятельности в программировании, поскольку ее сущность близка к определению инженерной деятельности в толковом словаре:

*инженерия* – это способ применения научных результатов, что позволяет получать пользу от свойств материалов и источников энергии;

*инженерия* – деятельность по созданию машин для предоставления полезных для потребителя услуг и изделий.

Инженеры в программной инженерии – это специалисты, выполняющие практические работы по реализации программ с применением теории, методов и средств компьютерной науки. Компьютерная наука охватывает теорию и методы построения вычислительных и программных систем, тогда как *программная инженерия* рассматривает вопросы практического построения ПО. Знание компьютерной науки необходимо специалистам в области программного обеспечения так же, как знание физики – инженерам-электронщикам. Если для решения конкретных задач программирования не существует подходящих методов или теории, инженеры применяют свои знания, накопленные ими в процессе разработок конкретных ПО, а также используют *опыт* применения соответствующих инструментальных программных средств.

Инженеры, как правило, работают в условиях заключенных контрактов и выполняют задачи проекта с учетом этих условий и ограничений на сроки, время, *стоимость* и др. В отличие от науки, цель которой – получение знаний, для инженерии *знание* – это способ получения некоторой пользы. Кроме программистов, занимающихся непосредственно разработкой ПО, в программной инженерии используются:

1. менеджеры, которые планируют и руководят проектом, отслеживают сроки и затраты;
2. инженеры службы ведения библиотек и репозитариев компонентов;
3. технологи, которые определяют инженерные методы и стандарты, создают для проекта модель ЖЦ, удовлетворяющую его целям и задачам;
4. тестировщики (контролеры), которые проверяют правильность выполнения процесса проектирования путем тестирования и на основе собранных данных проводят измерения разных характеристик качества, включая оценку надежности ПО;
5. верификаторы, которые проверяют правильность реализации функций в проекте;
6. валидаторы, проверяющие ПО на соответствие заданным требованиям.

Разработку программных систем можно считать инженерной деятельностью, но она имеет некоторые отличия от традиционной инженерии:

ветви инженерии имеют высокую степень специализации, а в программной инженерии специализация коснулась только отдельных областей (например, операционные системы, трансляторы, редакторы и т.п.);

программирование объектов основывается на стандартах, с помощью которых отражаются типовые требования заказчиков, т.е. типизация объектов и артефактов в сфере программирования;

технические решения классифицированы и каталогизированы, а в программной инженерии каждая новая разработка – это новая проблема, для реализации которой устанавливают аналогию с ранее разработанными системами. Одним из инженерных решений каталогизации в программировании является *паттерн* проектирования.

Для превращения программной инженерии в специальность мировая компьютерная общественность создала профессиональные комитеты, регламентирующие аспекты процесса программирования: *ядро* знаний SWEBOOK, этический кодекс программиста [1.13], учебные курсы (Curricula –2001, 2004) по подготовке специалистов в области программной инженерии, обучение специальности и *сертификация* специалистов.

Таким образом, возникновение программной инженерии как дисциплины разработки ПО определено следующими важными факторами:

- значительным объемом накопленных знаний в области создания ПО;
- появлением новых методов анализа, моделирования и проектирования ПО;

совершенствованием методов обнаружения ошибок в ПО;  
эффективной организацией коллективов разработчиков ПО и оценки их трудовой деятельности;  
использованием готовых программных компонентов, высокотехнологических средств и инструментов разработки ПО;  
необходимостью эволюционного развития компонентов и систем, а также их адаптацией к новым изменяющимся условиям операционных сред и компьютерных сетей.

*Программная инженерия* делает акцент на повышении качества и производительности *ПО* за счет применения: новых и усовершенствованных методов проектирования *ПО*; готовых компонентов и методов их генерации; методов эволюции, верификации и тестирования *ПО*; инструментальных средств; методов управления проектами, оценки качества и стоимости.

Внимание! Если Вы увидите ошибку на нашем сайте, выделите её и нажмите Ctrl+Enter.

---

© Национальный Открытый Университет "ИНТУИТ", 2022 | [www.intuit.ru](http://www.intuit.ru)