**Конспект 1 Лекции по Программной Инженерии.**

## Что такое инженерия и назначение инженера

**Инженер** – это специалист, осуществляющий инженерную деятельность. В свою очередь, **инженерная деятельность** – область технической деятельности, включающая в себя ряд специализированных областей и дисциплин.

Конечной целью инженерной деятельности являются **изобретение, разработка, создание, внедрение, обслуживание и/или улучшение техники, материалов или процессов**.

**Проектирование следует отличать от конструирования.**

Продукт проектировочной деятельности выражается в особой знаковой форме – в виде текста, чертежей, графиков, расчётов, моделей в памяти ЭВМ и, как правило, технико–экономического обоснования.

Результаты конструкторской деятельности должен быть обязательно материализован в виде опытного образца.

**Социальная значимость результата инженерной деятельности**

Инженер (коллектив инженеров) все более и более удаляется от непосредственного потребителя его изделия, конструируя артефакт (техническую систему) отделенным от конкретного человека, служить которому прежде всего и призван инженер. Непосредственная связь изготовителя и потребителя, характерная для ремесленной технической деятельности, нарушается. У него создаётся иллюзия, что внедрение сконструированного артефакта реализуется автоматически.

Отмеченные особенности привели к возникновению **социотехнического проектирования**, необходимости подготовки специалистов соответствующего профиля.



**Сущность и логика становления программной инженерии**

*«****Малое» (простое****) программное обеспечение* имеет следующие характеристики:

● решает **одну несложную, четко поставленную задачу**;

**● размер исходного кода** не превышает **нескольких сотен строк**;

● **скорость** работы программного обеспечения и необходимые ему **ресурсы не играют большой роли**;

● **ущерб** от неправильной работы не имеет большого значения;

● **модернизация** программного обеспечения, дополнение его возможностей требуется редко;

**● как правило**, разрабатывается одним программистом или небольшой группой (**5 или менее человек**);

● **подробная документация** не требуется, ее может заменить исходный **код**, который **доступен**.

***«Большое» (сложное****) программное обеспечение* имеет **2-3 или более** характеристик из следующего перечня:

● решает **совокупность** взаимосвязанных задач;

● использование приносит **значимую выгоду**;

● **удобство** его использования играет **важную роль**;

● **обязательно** наличие **полной и понятной документации**;

● **низкая скорость** работы приводит к **потерям**;

● сбои, **неправильная работа**, наносит **ощутимый ущерб**;

● программы в составе ПО во время работы **взаимодействует** с другими **программами** и **программно-аппаратными комплексами**;

● работает на **разных** платформах;

● требуется **развитие**, **исправление ошибок**, добавление новых возможностей;

● группа разработчиков состоит из **более 5 человек**.

Классификация программных проектов по созданию сложного ПО может быть проведена по размеру группы разработчиков и длительности проекта:

● ***небольшие проекты*** – проектная команда менее 10 человек, срок от 3 до 6 месяцев;

● ***средние проекты*** – проектная команда от 20 до 30 человек, протяженность проекта 1-2 года;

● ***крупномасштабные проекты*** – проектная команда от 100 до 300 человек, протяженность проекта 3-5 лет;

● ***гигантские проекты***– **армия** разработчиков от 1000 до 2000 человек и более (включая консультантов и соисполнителей), протяженность проекта от 7 до 10 лет.

***системный подход*** – это методология исследования объекта любой природы как системы, а ***система***– это совокупность взаимосвязанных частей, работающих совместно для достижения некоторого результата.

**Предпосылки возникновения программной инженерии**

**первый кризис программирования:**

стоимость программного обеспечения стала приближаться к стоимости аппаратуры («железа»), а динамика роста этих стоимостей позволяла прогнозировать, что к середине 90-годов все человечество будет заниматься разработкой программ для компьютеров. Тогда и заговорили о программной инженерии.

Аналитики процессов в индустрии производства программного обеспечения сформулировали основные причины неудач при разработке ПО. Это:

● нечеткая и неполная формулировка требований;

● недостаточное вовлечение пользователей в работу над проектом;

● отсутствие необходимых ресурсов;

● неудовлетворительное планирование и отсутствие грамотного управления проектом;

● частое изменение требований и спецификаций;

● новизна и несовершенство используемой технологии;

● недостаточная поддержка со стороны высшего руководства;

● недостаточно высокая квалификация разработчиков, отсутствие необходимого опыта.

посчитанной, как если бы они работали в одиночку).

Отметим **особенности современных проектов ПО:**

● сложность – неотъемлемая характеристика создаваемого ПО;

● отсутствие полных аналогов и высокая доля вновь разрабатываемого ПО;

● наличие унаследованного ПО и необходимость его интеграции с разрабатываемым ПО;

● территориально распределенная и неоднородная среда функционирования;

● большое количество участников проектирования, разобщенность и разнородность отдельных групп разработчиков по уровню квалификации и опыту.

Разработка ПО имеет следующие **специфические** особенности:

● **неформальный** характер требований к ПО и формализованный основной объект разработки – программы;

● **творческий характер** разработки;

● **дуализм ПО**, которое, с одной стороны, является статическим объектом – совокупностью текстов, с другой стороны, – динамическим, поскольку при эксплуатации порождаются процессы обработки данных;

● при своем **использовании** (эксплуатации) ПО не **расходуется и не изнашивается, но морально устаревает**;

● «**неощутимость**», «воздушность», «квазинематериальность» ПО, что подталкивает к **безответственному переделыванию**, поскольку легко стереть и переписать, чего не сделаешь при проектировании зданий и аппаратуры.

**Фундаментальная идея программной инженерии: проектирование ПО является формальным процессом, который можно изучать и совершенствовать.**

Основными целями программной инженерии являются:

● Системы должны создаваться в короткие сроки и соответствовать требованиям заказчика на момент внедрения.

● Качество ПО должно быть высоким.

● Разработка ПО должна быть осуществлена в рамках выделенного бюджета.

● Системы должны работать на оборудовании заказчика, а также взаимодействовать с имеющимся ПО.

● Системы должны быть легко сопровождаемыми и масштабируемыми.

**Этапы развития программной инженерии**

* 70-е и 80-е годы XX века– систематизация и стандартизация процессов создания ПО (структурный подход);
* 90-е годы- начало 21-го века – переход к сборочному, индустриальному способу создания ПО (объектно-ориентированный подход);
* с середины 90-х годов до настоящего времени – развитие компонентного подхода и сетевых технологий, создание CASE-технологий проектирования ПО

***Первый этап*** *– «стихийное» программирование*(от появления первых вычислительных машин до середины 60-х годов XX в). Первые программы имели простейшую структуру. Они состояли из собственно программы на машинном языке и обрабатываемых ею данных (рис. 1.2). Сложность программ в машинных кодах ограничивалась способностью программиста одновременно мысленно отслеживать последовательность выполняемых операций и местонахождение данных при программировании.

***Второй этап*** *– структурный подход к программированию* (60-70-е годы XX в.). В основе структурного подхода лежит *декомпозиция* (разбиение на части) сложных систем с целью последующей реализации в виде отдельных небольших (**до 40-50 операторов**) подпрограмм. При таком подходе задача представляется в виде иерархии подзадач простейшей структуры. Проектирование осуществляется «сверху вниз» и подразумевает реализацию общей идеи, обеспечивая проработку интерфейсов подпрограмм (*нисходящее проектирование*). Одновременно вводятся:

- ограничения на конструкции алгоритмов;

- формальные модели их описания;

- метод пошаговой детализации проектирования алгоритмов.

**Структурное программирование — методология разработки программного обеспечения, в основе которой лежит представление программы в виде иерархической структуры блоков.**

Предложена в 1970-х годах Э. Дейкстрой[[1]](#footnote-1) и другими.

В соответствии с данной методологией любая программа строится **без использования оператора goto** из трёх базовых управляющих структур:

**последовательность**, **ветвление**, **цикл**; кроме того, используются **подпрограммы**.

**Цель структурного программирования** — повысить производительность труда программистов, в том числе при разработке больших и сложных программных комплексов, сократить число ошибок, упростить отладку, модификацию и сопровождение программного обеспечения.

Таким образом была (до известной степени) решена **проблема возрастания сложности программных комплексов**.

К числу таких сложных программ относятся: **системы управления космическими объектами, управления оборонным комплексом, автоматизации крупного финансового учреждения и т.д**. Сложность таких комплексов оценивалась следующими показателями:

1. Большой объем кода (миллионы строк)
2. Большое количество связей между элементами кода
3. Большое количество разработчиков (сотни человек)
4. Большое количество пользователей (сотни и тысячи)
5. Длительное время использования

Для таких сложных программ оказалось, что **основная часть их стоимости приходится не на создание программ, а на их внедрение и эксплуатацию**. По аналогии с промышленной технологией стали говорить о **жизненном цикле** программного продукта, как о **последовательности определенных этапов: этапа проектирования, разработки, тестирования, внедрения и сопровождения.**

1. <http://samag.ru/uart/more/16> [↑](#footnote-ref-1)