
КОРОЛЬ Иван Андреевич –

заместитель начальника управления стратегических проектов Министерства связи и информатизации (*г. Минск, пр. Независимости, 10*)

к. ф.-м. н.; ст. н. с.; доцент

- E-mail: korol@mpt.gov.by; korol.minsk@gmail.com
 - Раб. тел. - (017) 287-87-22
 - Моб. тел. +375-296-39-42-79
 - Skype: Korol56
-

1	Основы теории проектирования
2	Жизненный цикл программных систем
3	Унифицированный язык визуального моделирования UML
4	Этапы проектирования программных систем с применением языка UML
5	Управление проектами. Определения и концепции
6	Инициация (запуск) проекта
7	Планирование проекта
8	Управление рисками проекта
9	Оценка трудоемкости и сроков разработки программного обеспечения
10	Формирование команды разработчиков проекта
11	Реализация проекта

Лекция

**ОСНОВЫ
ТЕОРИИ ПРОЕКТИРОВАНИЯ**

Король Иван Андреевич

-
1. Руководство к Своду знаний по управлению проектами. Третье издание (Руководство PMBOK). Американский национальный стандарт ANSI/PMI 99-001-2004 [Электронный ресурс]. – Режим доступа: http://eabanks.ucoz.ru/_Id/0/4_PMBOK3rd Russian.pdf. – Дата доступа: 11.08.2011.
 2. «Руководство к своду знаний по программной инженерии». The Guide to the Software Engineering Body of Knowledge, SWEBOOK, IEEE Computer Society Professional Practices Committee, 2004.
 3. Технологии программирования: Учебник/В.А. Камаев, В.В. Костерин. – 2-е изд., перераб. и доп. – М.: Высш. шк., 2006. - 454 с.
 4. С. Архипенков «Лекции по управлению программными проектами», М., 2009
-

■ **ВОПРОСЫ:**

- 1.1. Общие положения - термины и определения
 - 1.2. Системный подход и программирование
 - 1.3. Общесистемные принципы создания программ
 - 1.4. Стандарты и программирование
 - 1.5. Описание жизненного цикла программного обеспечения
 - 1.6. Стадии и этапы разработки программ
 - 1.7. Выводы
 - 1.8. Контрольные вопросы
-

Основы теории проектирования

- ❑ Как **без оформленного проекта** вполне можно построить скворечник, но невозможно строительство высотного здания или комплекса космодрома со строительной индустрией, жилыми, стартовыми и производственными комплексами, так и **без проекта** можно реализовать лишь небольшую программу, но не автоматизированное рабочее место специалиста, а тем более автоматизированную систему управления большого предприятия.
- ❑ **Что же производят программисты?**
- ❑ Программисты производят **программный продукт**.
- ❑ В терминах автоматизированных систем программисты создают **программное обеспечение**.

Термины и определения

- ❑ **Программный продукт** — программа, которую можно запускать, тестировать, исправлять и развивать. Такая программа должна быть написана в едином стиле, тщательно оттестирована до требуемого уровня надежности, сопровождается подробной документацией и подготовлена для тиражирования.
- ❑ **Программная продукция [Software Product]** – программный объект, предназначенный для поставки пользователю (ГОСТ Р ИСО/МЭК 9126-93).
- ❑ **Программа; компьютерная программа [Program; Computer Program]** – синтаксическая единица, подчиняющаяся правилам специфического языка программирования и состоящая из описаний и операторов или команд, необходимых для решения определенной функции, задачи или проблемы.
- ❑ **Программный модуль [Software Unit]** – отдельно компилируемая часть программного кода (программы).

-
- ❑ **Программное обеспечение [Software]** – программы, процедуры, правила и любая соответствующая документация, относящиеся к работе вычислительной системы (ГОСТ Р ИСО/МЭК 9126-93).
 - ❑ **Программное обеспечение автоматизированных систем** – совокупность программ на носителях данных и программных документов, предназначенная для отладки, функционирования и проверки работоспособности автоматизированных систем.
 - ❑ **Система [System]** – комплекс, состоящий из процессов, технических и программных средств, устройств и персонала, обладающий возможностью удовлетворять установленным потребностям или целям.
 - ❑ **Информационная система** – организованная совокупность информационных технологий, объектов и отношений между ними, образующая единое целое (СТБ 982-94).
-

-
- **Информационная система** [*Information System*] – система обработки информации в совокупности с относящимися к ней ресурсами организации, такими как люди, технические и финансовые ресурсы, которая предоставляет и распределяет информацию (ГОСТ ИСО/МЭК 2382-1-99).
 - **Автоматизированная система** (АС) – комплекс технических, программных, других средств и персонала, предназначенный для автоматизации различных процессов.
 - **Автоматизированная или автоматическая информационная система** – совокупность информационных технологий и комплекса программно-технических средств, осуществляющих информационные процессы в человеко-машинном или автоматическом режиме.
-

-
- Несмотря на то, что нет точного определения **программной системы**, можно сказать, что она представляет собой набор решений множества различных, но связанных между собой задач, и далее положиться на интуицию в случаях, когда надо отличить программу от системы.
 - **Примеры систем**: ОС, СУБД, система продажи авиабилетов и др.
 - **Примеры программ**: редактор текстов, компилятор, программы посылки запросов от кассира и др.
-

-
- ❑ **Программа** — очень *сложный объект*, содержащий до сотен тысяч и даже нескольких миллионов мыслей.
 - ❑ Сложность программного продукта — отнюдь не случайное свойство, скорее *необходимое*.
 - ❑ Его ***сложность определяется четырьмя основными причинами:***
 - ❑ сложностью поставленной задачи,
 - ❑ сложностью управления процессом разработки,
 - ❑ сложностью описания поведения отдельных подсистем,
 - ❑ сложностью обеспечения гибкости конечного программного продукта.
-

-
- ❑ **Проектирование** — это разработка проекта, процесс создания спецификации, необходимой для построения в заданных условиях еще несуществующего объекта на основе первичного описания этого объекта.
 - ❑ Результатом проектирования является проектное решение или совокупность проектных решений, удовлетворяющих заданным требованиям.
 - ❑ **Спецификация** в сфере проектной деятельности — это какое-либо описание в точных терминах.
 - ❑ **Проектным документом** называют документ, выполненный по заданной форме, в котором представлено проектное решение.
 - ❑ В программировании проектные решения оформляются в виде программной документации.
 - ❑ Различают внешнюю программную документацию, которая согласуется с заказчиком, и внутреннюю промежуточную документацию проекта, которая необходима самим программистам для их работы.
-

- ❑ **Проект** (от лат. *projectus* — брошенный вперед) — совокупность проектных документов в соответствии с установленным перечнем, которая представляет результат проектирования.
- ❑ **Проектной ситуацией** называют реальность (ситуацию), в которой ведется проектирование. Паровоз и электровоз проектировались в разных проектных ситуациях, определенных уровнем знаний человечества. Именно поэтому XIX в. стал веком паровоза.

Любая задача характеризуется необходимостью преобразования некоторой исходной ситуации в ситуацию, называемую **решением**. Говоря о любой задаче, всегда имеем ее информационные элементы:

- ❑ информация об условии (условие задачи) — **что задано**;
- ❑ информация о решении (признаки исходной ситуации) — **что требуется получить**;
- ❑ информация о технологии преобразования условия в решение — **как решить**.

- **Алгоритм** — строго однозначно определенная для исполнителя последовательность действий, приводящих к решению задач. Современное значение слова «алгоритм» во многом аналогично таким понятиям, как рецепт, процесс, методика, способ. Согласно **Д. Кнуту**, алгоритм имеет пять важных свойств.
- **Конечность**. Алгоритм всегда должен заканчиваться после выполнения конечного числа шагов.
- **Определенность**. Каждый шаг алгоритма должен быть точно определен.
- **Наличие входных данных**. Алгоритм имеет некоторое число входных данных, задающихся до начала его работы или определяющихся динамически во время его выполнения.
- **Наличие выходных данных**. Алгоритм имеет одно или несколько выходных данных, имеющих определенную связь с входными данными.
- **Эффективность**. Алгоритм обычно считается эффективным, если его операторы достаточно просты для того, чтобы их можно было точно выполнить при помощи карандаша и бумаги в течение конечного промежутка времени.

-
- ❑ Важными при разработке процессов проектирования являются такие понятия, как стратегия и тактика.
 - ❑ **Стратегия** (от греч. *stratos* — войско и *ago* — веду) — наука, искусство генерации наиболее существенных **общих долгосрочных целей и наиболее общего плана достижения преимущества**, курса действий и распределения ресурсов еще до выполнения реальных действий.
 - ❑ **Тактика** (от греч. *taktika* — искусство приводить в порядок) — фиксированная в своей последовательности **совокупность средств и приемов для достижения намеченной цели** и искусство ее применения, способы действия, ориентированные на достижение конкретных целей, являющиеся звеньями реализации стратегических целей.
 - ❑ Целью применения способа действия является совершение оптимальных действий, в заранее не предсказанных стратегическим планом ситуациях, уже в процессе выполнения реальных действий.
-

-
- **Стратегия** определяет, **куда, в каком направлении двигаться**, куда держать курс еще до начала проекта.
 - А **тактика** определяет, **как, каким способом двигаться**, какие конкретные действия предпринимать при затруднениях в ходе выполнения проекта.
 - Стратегия выполнения конкретного проекта описывается в программном документе — *техническом задании*.
-

-
- ❑ **Технология** (от греч. *techne* — искусство, мастерство, умение и *logos* — слово, учение) — совокупность производственных процессов в определенной отрасли производства, а также научное описание способов производства, совокупность приемов, применяемых в каком-либо деле, мастерстве, искусстве.
 - ❑ **Технология программирования** как наука изучает технологические процессы и порядок их прохождения (с использованием знаний, методов и средств).
-

-
- **Технологический процесс** — последовательность направленных на создание заданного объекта действий (технологических процедур и операций), каждое из которых основано на каких-либо естественных процессах и человеческой деятельности.
 - Знания, методы и средства могут использоваться в разных процессах и, следовательно, в технологиях.
 - **Технология программирования** — для инженера это научная и практически апробированная **стратегия создания программ**, содержащая описание совокупности методов и средств разработки программ, а также порядок применения этих методов и средств.
-

-
- ▣ **Технология автоматизированной разработки программного обеспечения** [*Computer Aided Software Engineering, CASE*],
 - ▣ **Технология автоматизированной разработки систем** [*Computer Aided System Engineering, CASE*],
 - ▣ **CASE-технология** [*CASE Technology*] – автоматизированная технология, обеспечивающая с помощью предназначенного для этих целей инструментария (CASE-систем) комплексную поддержку разработки либо поддержку отдельных стадий жизненного цикла сложных программных или информационных систем.
-

-
- По своей природе **программа** (т. е. набор инструкций) гораздо ближе к **технологии** (точнее, к описанию технологического процесса преобразования входной информации в выходную информацию), чем к **изделию**.
 - Это означает, что для оценки производительности труда программиста не нужно искать способ оценки количества продукции, выпускаемой им, поскольку никакая физическая продукция не производится и, следовательно, нет ее объема.
 - **Программный продукт** является разработанной программистом информационной технологией, которая материализуется у заказчика в виде изделия, становясь автоматизированными системами и инструментами их обслуживания.
 - Это объяснение, по-видимому, снимает многие правовые проблемы, а также проблемы ценообразования.
-

■ **Системный подход** — общенаучный обобщенный алгоритм, предусматривающий всестороннее исследование сложного объекта с использованием *компонентного, структурного, функционального, параметрического и генетического видов анализа*.

■ **Компонентный анализ** — рассмотрение объекта, включающего в себя составные элементы и входящего, в свою очередь, в систему более высокого ранга.

■ **Структурный анализ** — выявление элементов объекта и связей между ними.

■ **Функциональный анализ** — рассмотрение объекта как комплекса выполняемых им полезных и вредных функций.

■ **Параметрический анализ** — установление качественных пределов развития объекта — физических, экономических, экологических и др.

□ Применительно к программам параметрами могут быть: время выполнения какого-нибудь алгоритма, размер занимаемой памяти и т. д.

□ При этом выявляются ключевые технические противоречия, мешающие дальнейшему развитию объекта, и ставится задача их устранения за счет новых технических решений.

■ **Генетический анализ** — исследование объекта на его соответствие законам развития программных систем. В процессе анализа изучается история развития (генезис) исследуемого объекта: конструкции аналогов и возможных частей, технологии изготовления, объемы тиражирования, языки программирования и т. д.

При **блочно-иерархическом подходе** процесс проектирования и представления о самом объекте расчленяется на уровни.

Методология блочно-иерархического подхода базируется на **трех концепциях**:

- ❑ разбиения и локальной оптимизации,
 - ❑ абстрагирования,
 - ❑ повторяемости.
-

Концепция **разбиения** позволяет сложную задачу проектирования объекта или системы свести к решению более простых задач с учетом их взаимосвязи.

- **Локальная оптимизация** подразумевает улучшение параметров внутри каждой простой задачи.
 - **Абстрагируемость** заключается в построении моделей, отражающих только значимые в данных условиях свойства объектов.
 - **Повторяемость** — в использовании существующего опыта проектирования.
-

При создании и развитии программного обеспечения (ПО) рекомендуется применять следующие общесистемные принципы:

- 1) **принцип включения**, предусматривающий, что требования к созданию, функционированию и развитию ПО определяются со стороны более сложной, включающей его в себя системы;
 - 2) **принцип системного единства**, состоящий в том, что на всех стадиях создания, функционирования и развития ПО его целостность будет обеспечиваться связями между подсистемами, а также функционированием подсистемы управления;
-

-
- 3) **принцип развития**, предусматривающий в ПО возможность его наращивания и совершенствования компонентов и связей между ними;
 - 4) **принцип комплексности**, заключающийся в том, что ПО обеспечивает связанность обработки информации как отдельных элементов, так и всего объема данных в целом на всех стадиях обработки;
 - 5) **принцип информационного единства**, т. е. во всех, подсистемах, средствах обеспечения и компонентах ПО используются единые термины, символы, условные обозначения и способы представления;
-

-
- 6) ***принцип совместимости***, состоящий в том, что язык, символы, коды и средства программного обеспечения согласованы, обеспечивают совместное функционирование всех подсистем и сохраняют открытой структуру системы в целом;
 - 7) ***принцип инвариантности***, предопределяющий, что подсистемы и компоненты ПО инвариантны к обрабатываемой информации, т. е. являются универсальными или типовыми.

Инвариантность –

- 1) свойство величин оставаться неизменными, сохраняться при тех или иных преобразованиях;
 - 2) неизменность, независимость от чего-либо.
-

Создание сложной системы немыслимо без **стандартов**.

Они нужны для борьбы с хаосом и неразберихой, но вместе с этим стандарт не должен быть слишком «узким» и мешать техническому прогрессу.

Сегодня используются обозначения **ГОСТ** (государственный стандарт) со времен Советского Союза и **СТБ** (стандарт Беларуси) новые стандарты Республики Беларусь.

Помимо государственных стандартов действуют отраслевые стандарты (**ОСТ**), стандарты предприятий (**СТП**).

Группа стандартов ГОСТ «Единая система программной документации» (ЕСПД) претерпела мало изменений с момента ее создания, пережила несколько поколений ЭВМ и революционных изменений технологий разработки программ.

При этом она до настоящего времени никогда не затрудняла новаций.

Помимо вышеизложенных **стандартов де-юре** имеются **стандарты де-факто**.

Стандарты де-факто появляются на основе идей какой-то широко известной разработки.

Выгодно делать продукты в стиле разработки какой-то фирмы, так как пользователи уже имеют навыки работы с меню в стиле «**Lotus**», **электронными таблицами, текстовыми редакторами**.

Обычно стандартом де-факто определяются используемые операционные системы, трансляторы с языков программирования

-
- В области программирования общепризнанной ведущей организацией по разработке стандартов является институт **ANSI** (**Американский национальный институт стандартов**).
 - Данный институт является лидером по установке стандартов языков программирования, кодовых таблиц клавиш и символов, выводимых на экран, и еще многих других.
 - Необходимо также отметить стандарты **Международной организация по стандартизации, ИСО** (International Organization for Standardization, **ISO**)
-

■ К сожалению, самое благородное дело стандартизации – **достижение всеобщей унификации и взаимозаменяемости** – может также стать тормозом развития.

■ **Вводя новый стандарт**, надо учитывать последствия ввода, особенно если стандарт является опережающим и опережает практику развития или если стандарт является слишком «узким» и тормозит эволюцию прогресса.

Во всем мире руководствуются **следующим отношением к стандартам:**

- или полностью им следуй,
- или делай свой собственный стандарт.

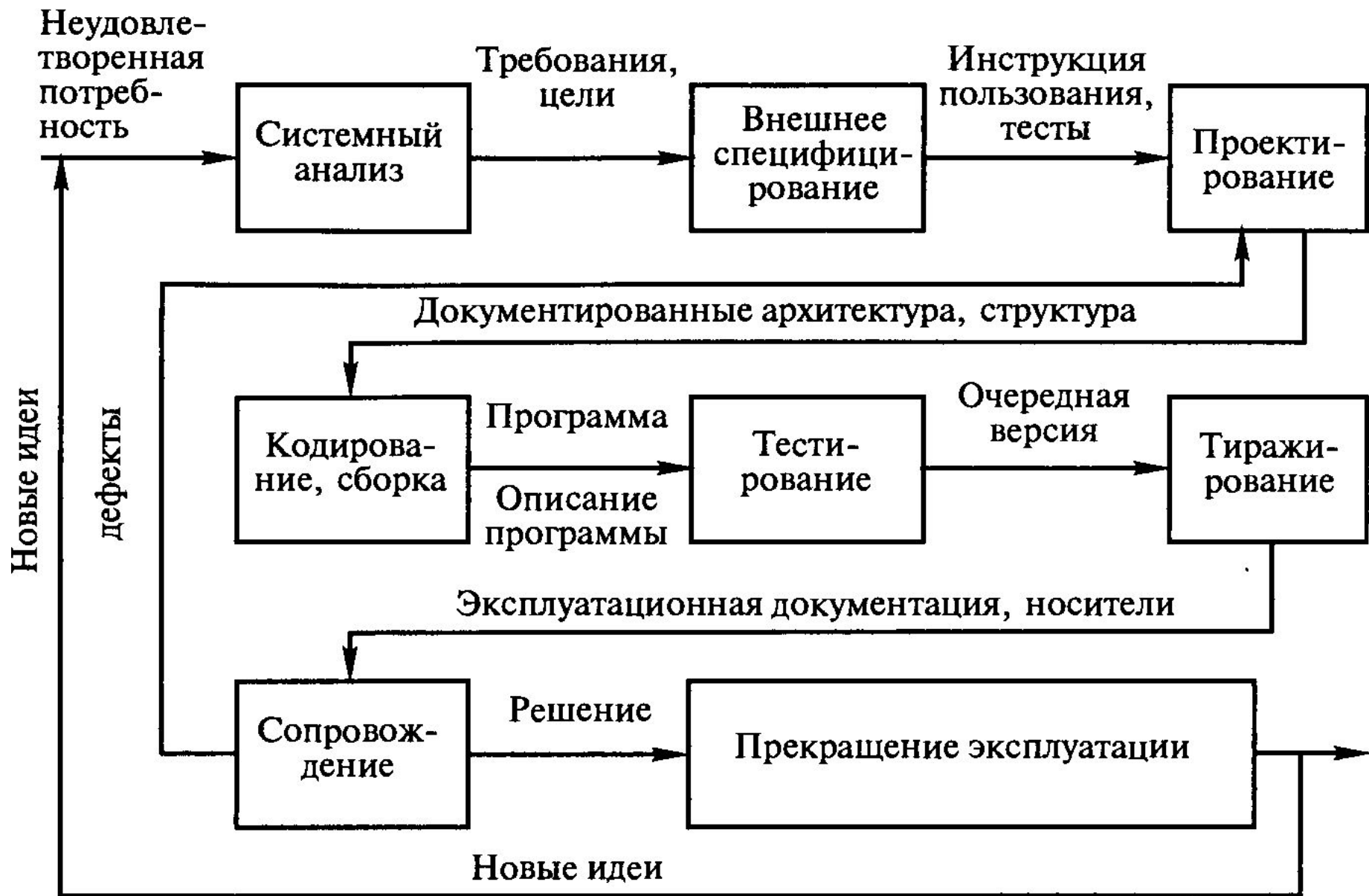
Стандарты дают дополнительные ограничения.

Программист должен уметь не только использовать готовые стандарты, но и разрабатывать новые.

Программы создаются, эксплуатируются и развиваются во времени.

Как и любые искусственные системы, они имеют свой жизненный цикл.

- **Жизненный цикл** – совокупность взаимосвязанных процессов создания и последовательного изменения состояния продукции от формирования к ней исходных требований до окончания ее эксплуатации или потребления.
-



ГОСТ 19.102-77. Единая система программной документации. **Стадии разработки.**

Стандарт регламентирует стадии и этапы программных разработок в течение всего жизненного цикла.

Данный стандарт сформировался на основе анализа удачных и неудачных программных разработок и содержит основные рекомендации по проведению новых разработок.

Стандарт уже пережил несколько технологий программирования.

При этом, практически не изменяясь, он не являлся тормозом прогресса.

■ Некоторые отечественные и зарубежные источники предлагают выделять следующие **этапы**:

1) **анализ требований**, предъявляемых к системе (системный анализ). (Обычно проводится на основе первичного исследования потоков информации при традиционном проведении работ с фиксацией видов этих работ и их последовательности);

-
- 2) **определение целей**, достигаемых разрабатываемыми программами;
 - 3) **выявление аналогов**, обеспечивающих достижение подобных целей, их достоинств и недостатков;
 - 4) **постановка задачи** на разработку новых программ, определение внешних спецификаций (т. е. описаний входной и выходной информации, а иногда и их форм) и способов (алгоритмов, методов) обработки информации;
 - 5) **оценка достижения целей** разработки;

-
-
- (Далее, при необходимости, этапы 1-5 **могут быть итеративно повторены** до достижения удовлетворительного облика изделия с описанием выполняемых им функций и некоторой ясностью реализации его функционирования);
-

6) **рассмотрение возможных вариантов структурного построения** программного изделия:

- или в виде **нескольких программ**, или **нескольких частей одной программы**;
 - результатом этой работы являются **варианты архитектуры программной системы** и (или) требования к структуре отдельных программных компонент; организация файлов для межпрограммного обмена данными;
-

-
- 7) разработка **окончательного варианта архитектуры системы** и разработка окончательной структуры программных компонент;
 - 8) составление и проверка **спецификаций модулей**;
 - 9) составление **описаний логики модулей**;
 - 10) составление окончательного **плана реализации программ**;
-

-
- 11) **кодирование и тестирование** отдельных модулей и совокупности готовых модулей до получения готовой программы;
 - 12) **комплексное тестирование**;
 - 13) **разработка эксплуатационной документации** на программу;
 - 14) **проведение приемо-сдаточных** и других испытаний;
 - 15) **корректировка** программ по результатам испытаний;
-

-
- 16) **окончательная сдача** программного изделия заказчику;
 - 17) **тиражирование** программного изделия;
 - 18) **сопровождение** программы.

- Современные технологии проектирования программного обеспечения направлены на частичную автоматизацию этапов и на совмещение их во времени с целью сокращения сроков выполнения проектов.

-
- **Проектирование** — высокоинтеллектуальный процесс.
 - Для понятия теории проектирования необходимо оперировать множеством терминов и определений, такими как **проектная ситуация, технология, оптимизация программных разработок**.
 - Все это говорит о необходимости тщательно подходить к изучению словарного аппарата теории проектирования.
-

■ Программы в основном представляют собой сложные системы из миллионов машинных инструкций. **Сложность** определяется **четырьмя основными причинами:**

■ сложностью задачи;

■ сложностью управления процессом разработки;

■ сложностью описания поведения отдельных подсистем;

■ сложностью обеспечения гибкости конечного программного продукта.

□ При разработке программного обеспечения следует использовать следующие **общие принципы**:

- частотный;
 - модульности;
 - функциональной избирательности;
 - генерируемости;
 - функциональной избыточности;
 - «по умолчанию».
-

-
-
- Одной из важнейших составляющих успешного проектирования является **системный подход**, предусматривающий всестороннее исследование сложного объекта.
-

□ При создании и развитии ПО рекомендуется применять следующие **общесистемные принципы**:

□ включения;

□ системного единства;

□ развития;

□ комплексности;

□ информационного единства;

□ совместимости;

□ инвариантности.

-
- Необходимо помнить, что проектирование неотъемлемо от различных **стандартов** (ГОСТ, ANSI, проекта) и их следует соблюдать как при оформлении документации, так и для унификации вашего проекта.
 - Программы создаются, эксплуатируются и развиваются во времени, проходя свой **жизненный цикл.**
 - Характерная особенность жизненного цикла ПО — отсутствие этапа утилизации.
-

-
- В процессе выполнения проекта предусматриваются отдельные моменты времени, которые характеризуются законченным оформлением результатов всех работ, выполненных разработчиками до данного момента.
 - Согласно ГОСТ возможны следующие **стадии разработки**: ТЗ; ЭП; ТП; РП; внедрение.
 - Возможны также и нестандартные этапы и стадии.
 - Набор этапов и стадий отражает результаты проектирования самого процесса проектирования.
-

-
1. Дайте определения: программная продукция, программное обеспечение, информационная и автоматизированная система.
 2. Какими основными причинами определяется сложность задачи?
 3. Дайте определения: проектирование, проект, проектная задача.
 4. Что такое метод проектирования?
 5. Какими свойствами характеризуется алгоритм?
 6. Какие виды анализа используются при системном подходе?
 7. Что такое принцип совместимости?
-

-
8. Для чего необходима стандартизация проектирования и программирования?
 9. Назовите основные этапы жизненного цикла программных изделий.
 10. Назовите основные стадии и этапы разработки программ по ГОСТ 19.102-77.
-

Лекция
ОСНОВЫ ТЕОРИИ ПРОЕКТИРОВАНИЯ

Король Иван Андреевич
