**Лекция 3**

**Документирование программных средств и систем**

## 1. Вопросы разработки программного обеспечения

Когда программист-разработчик получает в той или иной форме за­дание на программирование, перед ним, перед руководителем проекта и перед всей проектной группой встают вопросы:

• что должно быть сделано, кроме собственно программы?

• что и как должно быть оформлено в виде документации?

• что передавать пользователям, а что — службе сопровождения?

• как управлять всем этим процессом?

• что должно входить в само задание на программирование?

Кроме упомянутых вопросов есть и другие.

На эти и массу других вопросов когда-то отвечали государственные стандарты на программную документацию — комплекс стандартов 19-й серии ГОСТ ЕСПД. Но уже тогда у программистов была масса претензий к этим стандартам. Что-то требовалось дублировать в документации мно­го раз (как казалось, — неоправданно), а многое не было предусмотрено, как, например, отражение специфики документирования программ, ра­ботающих с интегрированной базой данных.

В настоящее время остается актуальным вопрос о наличии системы, регламентирующей документирование программных средств.

Применить этот материал в лекции.

Этот материал взят из стандарта СФУ на ЭОР.

Стандарт базируется на основе следующих нормативных документов:

* Закон РФ «Об авторском праве и смежных правах» (в текущей редакции).
* Закон РФ «О правовой охране программ для электронных вычислительных машин и баз данных» (в текущей редакции).
* Федеральный закон «Об обязательном экземпляре документов» (в текущей редакции).
* ГОСТ Р 1.4-2004. Стандартизация в Российской Федерации. Стандарты организаций. Общие положения.
* ГОСТ Р 1.5-2004. Стандартизация в Российской Федерации. Стандарты национальные Российской Федерации. Правила построения, изложения, оформления и обозначения.
* ГОСТ 7.82-2001. Библиографическая запись. Библиографическое описание электронных ресурсов.
* ГОСТ 7.60-2003. Межгосударственный стандарт СИБИД. Издания. Основные виды, термины и определения.
* ГОСТ 7.83-2001. Межгосударственный стандарт СИБИД. Электронные издания. Основные виды и выходные сведения.
* ГОСТ Р ИСО/МЭК NJ 9294-93. Информационная технология. Руководство по управлению документированием программного обеспечения.
* ГОСТ Р ИСО/МЭК 12207-99. Информационная технология. Процессы жизненного цикла программных средств.
* ГОСТ Р ИСО/МЭК 12119-2000. Информационная технология. Пакеты программ. Требования к качеству и тестированию.
* ГОСТ Р ИСО 9127-94. Системы обработки информации. Документация пользователя и информация на упаковке для потребительских программных пакетов.
* ГОСТ Р ИСО/МЭК 8631-94. Информационная технология. Программные конструктивы и условные обозначения для их представления.
* СТТ 115.50-03-2005. Требования Системы добровольной сертификации «Росинфосерт». Информационные технологии. Сертификация средств и систем в сфере информатизации. Программные средства проверки знаний тестированием. Характеристики функционального назначения, информационной совместимости и безопасности. Технические требования.
* Положение об электронных образовательных ресурсах Сибирского федерального университета.
* СТП КГТУ 4.2.10-2006. Электронные образовательные ресурсы в формате *portable document format.* Порядок оформления, компоновки и технические параметры.

### 2. Общая характеристика состояния

Основу отечественной нормативной базы в области документирова­ния ПС составляет комплекс стандартов Единой системы программной документации (ЕСПД). Основная и большая часть комплекса ЕСПД была разработана в 70-е и 80-е годы. Сейчас этот комплекс представляет собой систему межгосударственных стандартов стран СНГ (ГОСТ), действую­щих на основе межгосударственного соглашения по стандартизации.

Стандарты ЕСПД в основном охватывают ту часть документации, которая создается в процессе разработки ПС, и связаны по большей час­ти с документированием функциональных характеристик ПС. Следует отметить, что стандарты ЕСПД (ГОСТ 19) носят рекомендательный ха­рактер. Впрочем, это относится и ко всем другим стандартам в области ПС (ГОСТ 34, Международному стандарту ISO/IEC и др.). Эти стандар­ты становятся обязательными на контрактной основе, т.е. при ссылке на них в договоре на разработку (поставку) ПС.

Говоря о состоянии ЕСПД в целом, можно констатировать, что боль­шая часть стандартов ЕСПД морально устарела.

К числу основных недостатков ЕСПД можно отнести:

• ориентацию на единственную, «каскадную» модель жизненного цикла (ЖЦ) ПС;

• отсутствие четких рекомендаций по документированию характе­ристик качества ПС;

• отсутствие системной увязки с другими действующими отечествен­ными системами стандартов по ЖЦ и документированию продук­ции в целом, например ЕСКД;

• нечетко выраженный подход к документированию ПС как товар­ной продукции;

• отсутствие рекомендаций по самодокументированию ПС, напри­мер, в виде экранных меню и средств оперативной помощи пользо­вателю (хэлпов);

• отсутствие рекомендаций по составу, содержанию и оформлению перспективных документов на ПС, согласованных с рекомендаци­ями международных и региональных стандартов.

Итак, ЕСПД нуждается в полном пересмотре на основе стандарта ИСО/МЭК 12207-95 на процессы жизненного цикла ПС, об этом стан­дарте далее будет сказано подробнее.

Надо сказать, что наряду с комплексом ЕСПД официальная норма­тивная база России и стран СНГ в области документирования ПС и в смежных областях включает ряд перспективных стандартов (отечествен­ного, межгосударственного и международного уровней).

Международный стандарт ISO/IEC 12207: 1995-08-01 на организацию ЖЦ продуктов программного обеспечения (ПО) — казалось бы, весьма неконкретный, но вполне новый и отчасти модный стандарт.

Стандарты комплекса ГОСТ 34 на создание и развитие автоматизи­рованных систем (АС) — обобщенные, но воспринимаемые как весьма жесткие по структуре ЖЦ и проектной документации. Но эти стандарты многими считаются бюрократическими до вредности и консервативны­ми до устарелости. Насколько это так, а насколько ГОСТ 34 остается работающим с пользой — полезно разобраться.

#### 3. Краткое представление стандартов ЕСПД

Тем не менее до пересмотра всего комплекса многие стандарты ЕСПД могут с пользой применяться в практике документирования ПС. Эта по­зиция основана на следующем:

• стандарты ЕСПД вносят элемент упорядочения в процесс докумен­тирования ПС;

• предусмотренный стандартами ЕСПД состав программных доку­ментов вовсе не такой жесткий, как некоторым кажется, стандар­ты позволяют вносить в комплект документации на ПС дополни­тельные виды;

• стандарты ЕСПД позволяют вдобавок мобильно изменять структу­ры и содержание установленных видов ПД исходя из требований заказчика и пользователя.

При этом стиль применения стандартов может соответствовать со­временному общему стилю адаптации стандартов к специфике проекта: заказчик и руководитель проекта выбирают уместное в проекте подмно­жество стандартов и ПД, дополняют выбранные ПД нужными разделами и исключают ненужные, привязывают создание этих документов к той схеме ЖЦ, которая используется в проекте.

Стандарты ЕСПД (как и другие ГОСТы) подразделяют на группы, приведенные в таблице.

|  |  |
| --- | --- |
| *Код группы* | *Наименование группы* |
| 0 | Общие положения |
| 1 | Основополагающие стандарты |
| 2 | Правила выполнения документации разработки |
| 3 | Правила выполнения документации изготовления |
| 4 | Правила выполнения документации сопровождения |
| 5 | Правила выполнения эксплуатационной документации |
| 6 | Правила обращения программной документации |
| 7 | Резервные группы |
| 8 |
| 9 | Прочие стандарты |

Обозначение стандарта ЕСПД строят по классификационному при­знаку.

Обозначение стандарта ЕСПД должно состоять из:

• числа 19 (присвоенных классу стандартов ЕСПД);

• одной цифры (после точки), обозначающей код классификационной группы стандартов, указанной в таблице;

• двузначного числа (после тире), указывающего год регистрации стандарта.

*Перечень документов ЕСПД*:

1) ГОСТ 19.001-77 ЕСПД. Общие положения.

2) ГОСТ 19.101-77 ЕСПД. Виды программ и программных документов.

3) **ГОСТ 19.102-77 ЕСПД. Стадии разработки.**

4) ГОСТ 19.103-77 ЕСПД. Обозначение программ и программных до­кументов.

5) ГОСТ 19.104-78 ЕСПД. Основные надписи.

6) ГОСТ 19.105-78 ЕСПД. Общие требования к программным доку­ментам.

7) ГОСТ 19.106-78 ЕСПД. Требования к программным документам, выполненным печатным способом.

8) **ГОСТ 19.201-78 ЕСПД**. Техническое задание. Требования к содер­жанию и оформлению.

9) ГОСТ 19.202-78 ЕСПД. Спецификация. Требования к содержанию и оформлению.

10) ГОСТ 19.301-79 ЕСПД. Порядок и методика испытаний.

11) ГОСТ 19.401-78 ЕСПД. Текст программы. Требования к содержа­нию и оформлению.

12) ГОСТ 19.402-78 ЕСПД. Описание программы.

13) ГОСТ 19.404-79 ЕСПД. Пояснительная записка. Требования к со­держанию и оформлению.

14) ГОСТ 19.501-78 ЕСПД. Формуляр. Требования к содержанию и оформ­лению.

15) ГОСТ 19.502-78 ЕСПД. Описание применения. Требования к со­держанию и оформлению.

16) ГОСТ 19.503-79 ЕСПД. Руководство системного программиста. Тре­бования к содержанию и оформлению.

17) ГОСТ 19.504-79 ЕСПД. Руководство программиста.

18) ГОСТ 19.505-79 ЕСПД. Руководство оператора.

19) ГОСТ 19.506-79 ЕСПД. Описание языка.

20) ГОСТ 19.508-79 ЕСПД. Руководство по техническому обслужива­нию. Требования к содержанию и оформлению.

21) ГОСТ 19.604-78 ЕСПД. Правила внесения изменений в програм­мные документы, выполняемые печатным способом.

22) ГОСТ 19.701-90 ЕСПД. Схемы алгоритмов, программ, данных и си­стем. Условные обозначения и правила выполнения.

23) ГОСТ 19.781-90 ЕСПД. Обеспечение систем обработки информа­ции программное.

Из всех стандартов ЕСПД остановимся только на тех, которые могут чаще использоваться на практике.

Первым укажем стандарт, который можно использовать при форми­ровании заданий на программирование.

***ГОСТ (СТ СЭВ) 19.201-78(1626-79). ЕСПД. Техническое задание. Требования к содержанию и оформлению (переиздан в ноябре 1987 г. с изм. 1)***

Техническое задание (ТЗ) содержит совокупность требований к ПС и может использоваться как критерий проверки и приемки разработанной программы, поэтому достаточно полно составленное (с учетом возможности внесения дополнительных разделов) и принятое заказчиком и разработчи­ком ТЗ является одним из основополагающих документов проекта ПС.

Техническое задание должно содержать следующие разделы:

• введение;

• основания для разработки;

• назначение разработки;

• требования к программе или программному изделию;

• требования к программной документации;

• технико-экономические показатели;

• стадии и этапы разработки;

• порядок контроля и приемки;

• в техническое задание допускается включать приложения.

В зависимости от особенностей программы или программного изде­лия допускается уточнять содержание разделов, вводить новые разделы или объединять отдельные из них.

***ГОСТ (СТ СЭВ) 19.101-77 (1626-79). ЕСПД. Виды программ и про­граммных документов (переиздан в ноябре 1987 г. с изм. 1)***

Устанавливает виды программ и программных документов для вы­числительных машин, комплексов и систем независимо от их назначе­ния и области применения.

Виды программ

|  |  |
| --- | --- |
| *Вид программы* | *Определение* |
| Компонент | Программа, рассматриваемая как единое целое, выполняющая законченную функцию и применяемая самостоятельно или в составе комплекса |
| Комплекс | Программа, состоящая из двух или более компонентов и (или) комплексов, выполняющих взаимо­связанные функции, и применяемая самостоятельно или в составе другого комплекса |

#### Виды программных документов

|  |  |
| --- | --- |
| *Вид программного документа* | *Содержание программного*  *документа* |
| Спецификация | Состав программы и документации на нее |
| Ведомость держателей подлинников | Перечень предприятий, на которых хранят подлинники программных документов |
| Текст программы | Запись программы с необходимыми ком­ментариями |
| Описание программы | Сведения о логической структуре и функ­ционировании программы |
| Программа и методика испытаний | Требования, подлежащие проверке при испытании программы, а также порядок и методы их контроля |
| Техническое задание | Назначение и область применения програм­мы, технические, технико-экономические и специальные требования, предъявляемые к программе, необходимые стадии и сроки разработки, виды испытаний |
| Пояснительная записка | Схема алгоритма, общее описание алгорит­ма и (или) функционирования программы, а также обоснование принятых технических и технико-экономических решений |
| Эксплуатационные документы | Сведения для обеспечения функционирования и эксплуатации программы |

#### Виды эксплуатационных документов

|  |  |
| --- | --- |
| *Вид эксплуатационного документа* | *Содержание эксплуатационного документа* |
| Ведомость эксплуатациионных документов | Перечень эксплуатационных документов на программу |
| Формуляр | Основные характеристики программы, комплектность и сведения об эксплуатации программы |
| Описание применения | Сведения о назначении программы, обла­сти применения, применяемых методах, классе решаемых задач, ограничениях для применения, минимальной конфигурации технических средств |
| Руководство системного программиста | Сведения для проверки, обеспечения функционирования и настройки программы на условия конкретного применения |
| Руководство программиста | Сведения для эксплуатации программы |
| Руководство оператора | Сведения для обеспечения процедуры об­щения оператора с вычислительной систе­мой в процессе выполнения программы |
| Описание языка | Описание синтаксиса и семантики языка |
| Руководство по техническому обслуживанию | Сведения для применения тестовых и диагностических программ при обслуживании технических средств |

В зависимости от способа выполнения и характера применения про­граммные документы подразделяются на подлинник, дубликат и копию (ГОСТ 2.102-68), предназначенные для разработки, сопровождения и экс­плуатации программы.

#### Виды программных документов, разрабатываемых

**на разных стадиях, и их коды**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Код вида докумен-та* | *Вид документа* | *Стадии разработки* | | | |
| *Эскизный проект* | *Технический проект* | *Рабочий проект* | |
| *компонент* | *комплекс* |
| **-** | Спецификация | - | - | ! | + |
| 05 | Ведомость держателей подлинников | - | - | - | ? |
| 12 | Текст программы | - | - | + | ? |
|  |  |  |  |  |  |
|  |  | Окончание | | | |
| *Код вида докумен-та* | *Вид документа* | *Стадии разработки* | | | |
| *Эскизный проект* | *Технический проект* | *Рабочий проект* | |
| *компонент* | *комплекс* |
| 13 | Описание программы | - | - | ? | ? |
| 20 | Ведомость эксплуатацион­ных документов | - | - | ? | ? |
| 30 | Формуляр | - | - | ? | ? |
| 31 | Описание применения | - | - | ? | ? |
| 32 | Руководство системного программиста | - | - | ? | ? |
| 33 | Руководство программиста | - | - | ? | ? |
| 34 | Руководство оператора | - | - | ? | ? |
| 35 | Описание языка | - | - | ? | ? |
| 46 | Руководство по техническому обслуживанию | - | - | ? | ? |
| 51 | Программа и методика испытаний | - | - | ? | ? |
| 81 | Пояснительная записка | ? | ? | - | - |
| 90-99 | Прочие документы | ? | ? | ? | ? |

*Условные обозначения*:

+ документ обязательный;

! документ обязательный для компонентов, имеющих самостоятельное при­менение;

? необходимость составления документа определяется на этапе разработки и утверждения технического задания;

- документ не составляют.

Допускается объединять отдельные виды эксплуатационных докумен­тов (за исключением ведомости эксплуатационных документов и форму­ляра). Необходимость объединения этих документов указывается в тех­ническом задании. Объединенному документу присваивают наименова­ние и обозначение одного из объединяемых документов. В объединенных документах должны быть приведены сведения, которые необходимо включать в каждый объединяемый документ.

**ГОСТ 19.102-77. ЕСПД. Стадии разработки**

Устанавливает стадии разработки программ и программной докумен­тации для вычислительных машин, комплексов и систем независимо от их назначения и области применения.

**Стадии разработки, этапы и содержание работ**

|  |  |  |
| --- | --- | --- |
| *Стадии разработки* | *Этапы работ* | *Содержание работ* |
| Техническое задание | Обоснование необходимости разработки программы | Постановка задачи  Сбор исходных материалов  Выбор и обоснование критериев эффективности и качества разрабатываемой программы  Обоснование необходимости проведения научно-исследова­тельских работ |
| Научно-исследовательские работы | Определение структуры вход­ных и выходных данных  Предварительный выбор методов решения задач  Обоснование целесообразности применения ранее разрабо­танных программ  Определение требований к тех­ническим средствам  Обоснование принципиальной возможности решения постав­ленной задачи |
| Разработка утверждение технического задания | Определение требований к про­грамме  Разработка технико-экономического обоснования разработки программы  Определение стадий, этапов и сроков разработки программы и документации на нее  Выбор языков программиро­вания  Определение необходимости проведения научно-исследова­тельских работ на последующих стадиях  Согласование и утверждение технического задания |
|  |  | Предварительная разработка структуры входных и выходных данных |
|  |  |  |
|  |  | *Продолжение* |
| *Стадии разработки* | *Этапы работ* | *Содержание работ* |
| Эскизный проект | Разработка эскизного  проекта | Уточнение методов решения задачи  Разработка общего описания алгоритма решения задачи  Разработка технико-экономи­ческого обоснования |
| Утверждение эскизного проекта | Разработка пояснительной записки  Согласование и утверждение эскизного проекта |
| Технический проект | Разработка технического проекта | Уточнение структуры входных и выходных данных  Разработка алгоритма решения задачи  Определение формы представления входных и выходных данных  Определение семантики и синтаксиса языка  Разработка структуры про­граммы  Окончательное определение конфигурации технических средств |
| Утверждение технического проекта | Разработка плана мероприятий по разработке и внедрению программ  Разработка пояснительной за­писки  Согласование и утверждение технического проекта |
| Рабочий проект | Разработка программы | Программирование и отладка программы |
| Разработка программной документации | Разработка программных документов в соответствии с требованиями ГОСТ 19.101-77 |
| Испытания программы | Разработка, согласование и утверждение программы и методи­ки испытаний  Проведение предварительных государственных, межведом­ственных, приемосдаточных и других видов испытаний  Корректировка программы и программной документации по результатам испытаний |
|  |  |  |
|  |  | *Окончание* |
| Внедрение | Подготовка и передача программы | Подготовка и передача про­граммы и программной доку­ментации для сопровождения и (или) изготовления  Оформление и утверждение акта о передаче программы на сопровождение и (или) изго­товление  Передача в фонд алгоритмов и программ |

*Примечания:*

1) Допускается исключать вторую стадию разработки, а в технически обо­снованных случаях — вторую и третью стадии. Необходимость проведения этих стадий указывается в техническом задании.

2) Допускается объединять, исключать этапы работ и (или) их содержание, а также вводить другие этапы работ по согласованию с заказчиком.

**ГОСТ 19.103-77ЕСПД. Обозначение программ и программных документов**

Программа и ее документ «Спецификация» имеют следующую струк­туру обозначения:

А.В.ХХХХХ-ХХ

Номер издания (для программы)

Номер редакции (для документа)

Регистрационный номер

Код организации-разработчика

Код страны

Структура обозначения других программных документов:

А.В.ХХХХХ-ХХ XX ХХ-Х

Номер части документа

Номер документа данного вида

Код вида документа

Номер редакции документа

Общая часть обозначения программы

и программных документов на нее

• Код страны-разработчика и код организации-разработчика присва­ивают в установленном порядке.

• Регистрационный номер присваивается в порядке возрастания, на­чиная с 00001 до 99999, для каждой организации-разработчика.

• Номер издания программы или номер редакции. Номер документа данного вида, номер части документа присваиваются в порядке возрастания с 01 до 99. (Если документ состоит из одной части, то дефис и порядковый номер части не указывают.)

• Номер редакции спецификации и ведомости эксплуатационных до­кументов на программу должны совпадать с номером издания этой же программы.

**ГОСТ 19.105-78 ЕСПД. Общие требования к программным документам**

Настоящий стандарт устанавливает общие требования к оформлению программных документов для вычислительных машин, комплексов и сис­тем независимо от их назначения и области применения и предусмотренных стандартами Единой системы программной документации (ЕСПД) для любого способа выполнения документов на различных носителях данных.

Программный документ может быть представлен на различных типах носителей данных и состоит из следующих условных частей:

• титульной;

• информационной;

• основной.

Правила оформления документа и его частей на каждом носителе данных устанавливаются стандартами ЕСПД на правила оформления до­кументов на соответствующих носителях данных.

**ГОСТ 19.106-78 ЕСПД. Требования к программным документам, вы­полненным печатным способом**

Программные документы оформляют:

• на листах формата А4 (ГОСТ 2.301-68) при изготовлении докумен­та машинописным или рукописным способом;

• допускается оформление на листах формата АЗ;

• при машинном способе выполнения документа допускаются от­клонения размеров листов, соответствующих форматам А4 и АЗ, определяемые возможностями применяемых технических средств; на листах форматов А4 и А3, предусматриваемых выходными ха­рактеристиками устройств вывода данных, при изготовлении до­кумента машинным способом;

• на листах типографических форматов при изготовлении документа

типографским способом.

Расположение материалов программного документа осуществляется в следующей последовательности:

*титульная часть:*.

• лист утверждения (не входит в общее количество листов документа);

• титульный лист (первый лист документа);

*информационная часть*:

• аннотация;

• лист содержания;

*основная часть*:

• текст документа (с рисунками, таблицами и т.п.);

• перечень терминов и их определений;

• перечень сокращений;

• приложения;

• предметный указатель;

• перечень ссылочных документов;

*часть регистрации изменений*:

• лист регистрации изменений.

Перечень терминов и их определений, перечень сокращений, прило­жения, предметный указатель, перечень ссылочных документов выпол­няются при необходимости.

Следующий стандарт ориентирован на документирование результи­рующего продукта разработки.

**ГОСТ 19.402-78ЕСПД. Описание программы**

Состав документа «Описание программы» в своей содержательной части может дополняться разделами и пунктами, почерпнутыми из стан­дартов для других описательных документов и руководств:

• ГОСТ 19.404-79 ЕСПД. *Пояснительная записка;*

• ГОСТ 19.502-78 ЕСПД. *Описание применения;*

• ГОСТ 19.503-79 ЕСПД. *Руководство системного программиста*;

• ГОСТ 19.504-79 ЕСПД. *Руководство программиста*;

• ГОСТ 19.505-79 ЕСПД. *Руководство оператора*.

Есть также группа стандартов, определяющая требования к фикса­ции всего набора программ и ПД, которые оформляются для передачи ПС. Они порождают лаконичные документы учетного характера и могут быть полезны для упорядочения всего хозяйства программ и ПД (ведь очень часто требуется просто навести элементарный порядок!). Есть и стандарты, определяющие правила ведения документов в «хозяйстве» ПС.

Надо также выделить **ГОСТ 19.301-79 ЕСПД. Программа и методика испытаний**, который (в адаптированном виде) может использоваться для разработки документов планирования и проведения испытательных ра­бот по оценке готовности и качества ПС.

Наконец, последний по году принятия стандарт.

**ГОСТ 19.701-90 ЕСПД. Схемы алгоритмов, программ, данных и сис­тем. Обозначения условные графические и правила выполнения**

Он устанавливает правила выполнения схем, используемых для ото­бражения различных видов задач обработки данных и средств их реше­ния и полностью соответствует стандарту ИСО 5807:1985.

Наряду с ЕСПД на межгосударственном уровне действуют еще два стандарта, также относящиеся к документированию ПС и принятые не так давно, как большая часть ГОСТ ЕСПД.

**ГОСТ 19.781-90. Обеспечение систем обработки информации програм­мное. Термины и определения**

Разработан взамен ГОСТ 19.781-83 и ГОСТ 19.004-80 и устанавлива­ет термины и определения понятий в области программного обеспечения систем обработки данных (СОД), применяемые во всех видах документа­ции и литературы, входящих в сферу работ по стандартизации или ис­пользующих результаты этих работ.

#### 4. Стандарты комплекса ГОСТ 34

ГОСТ 34 задумывался в конце 1980-х годов как всеобъемлющий ком­плекс взаимоувязанных межотраслевых документов. Мотивы и получив­шиеся результаты описаны ниже в «Особенностях» ГОСТ 34. Объектами стандартизации являются автоматизированные системы (АС) различных (любых!) видов и все виды их компонентов, а не только ПО и БД.

Комплекс рассчитан на взаимодействие заказчика и разработчика. Аналогично ISO12207 предусмотрено, что заказчик может разрабатывать АС для себя сам (если создаст для этого специализированное подразделе­ние). Однако формулировки ГОСТ 34 не ориентированы на столь явное и в известном смысле симметричное отражение действий обеих сторон, как ISO 12207. Поскольку ГОСТ 34 в основном уделяет внимание содер­жанию проектных документов, распределение действий между сторона­ми обычно делается, отталкиваясь от этого содержания.

Из всех существующих и нереализованных групп документов будем основываться только на *Группе 0 «Общие положения»* и *Группе 6 «Созда­ние, функционирование и развитие АС»*. Наиболее популярными можно считать стандарты *ГОСТ34.601-90 (Стадии создания АС), ГОСТ34.602-89 (ТЗ на создание АС) и методические указания РД 50-34.698-90 (Требования к содержанию документов)*. Стандарты предусматривают стадии и этапы выполнения работ по созданию АС, но не предусматривают сквозных процессов в явном виде.

Для общего случая разработки АС стадии и этапы ГОСТ 34 приведе­ны в таблице.

|  |  |
| --- | --- |
| 1. ФТ - Формирование требований к АС | * 1. Обследование объекта и обоснование необходимости создания АС   2. Формирование требований пользователя к АС   3. Оформление отчета о выполненной работе и заявки на разработку АС (тактико-технического задания) |
| 2. РК - Разработка концепции АС | 2.1. Изучение объекта  2.2. Проведение необходимых научно- исследовательских работ  2.3. Разработка вариантов концепции АС, удовлетворяющей требованиям пользо­вателя  2.4. Оформление отчета о выполненной работе |
| 3. ТЗ - Техническое зада­ние на создание АС | 3.1. Разработка и утверждение технического задания на создание АС |
| 4. ЭП - Эскизный проект | 4.1. Разработка предварительных проект-ных решении по системе и ее частям  4.2. Разработка документации на АС и ее части |
| 5. ТП - Технический проект | 5.1. Разработка проектных решений по системе и ее частям  5.2. Разработка документации на АС и ее части  5.3. Разработка и оформление документа-ции на поставку изделии для комплекто-вания АС и (или) технических требований (технических заданий) на их разработку  5.4. Разработка заданий на проектирование в смежных частях проекта объекта автоматизации |
| 6. РД - Рабочая документация | 6.1. Разработка рабочей документации на систему и ее части  6.2. Разработка или адаптация программ |
| 7. ВД - Ввод в действие | 7.1. Подготовка объекта автоматизации к вводу АС в действие  7.2. Подготовка персонала  7.3. Комплектация АС поставляемыми изделиями (программными и техни­ческими средствами, программно-техническими комплексами, инфор­мационными изделиями) |
|  |  |
|  | *Окончание* |
|  | 7.4. Строительно-монтажные работы  7.5. Пуско-наладочные работы  7.6. Проведение предварительных испытаний  7.7. Проведение опытной эксплуатации  7.8. Проведение приемочных испытаний |
| 8. Сп - Сопровождение АС | 8.1. Выполнение работ в соответствии с гарантийными обязательствами  8.2. Послегарантийное обслуживание |

Описано содержание документов, разрабатываемых на каждом этапе. Это определяет потенциальные возможности выделения на содержательном уровне сквозных работ, выполняемых параллельно или последовательно (то есть по сути — процессов), и составляющих их задач. Такой прием может использоваться при построении профиля стандартов ЖЦ проекта, вклю­чающего согласованные подмножества стандартов ГОСТ 34 и ISO12207.

Главный мотив: разрешить проблему «вавилонской башни».

В 1980-х годах сложилось положение, при котором в различных от­раслях и областях деятельности использовалась плохо согласованная или несогласованная НТД — нормативно-техническая документация. Это за­трудняло интеграцию систем, обеспечение их эффективного совместного функционирования. Действовали различные комплексы и системы стан­дартов, устанавливающие требования к различным видам АС.

Практика применения стандартов показала, что в них применяется по существу (но не по строгим определениям) единая система понятий, есть много общих объектов стандартизации, однако требования стандар­тов не согласованы между собой, имеются различия по составу и содер­жанию работ, различия по обозначению, составу, содержанию и оформ­лению документов и пр.

Конечно, эта ситуация отчасти отражала и естественное многообра­зие условий разработки АС, целей разработчиков, применяемых подхо­дов и методик.

В этих условиях можно было провести анализ такого многообразия и далее поступить, например, одним из двух во многом противоположных способов:

• выработать одну обобщенную понятийную и терминологическую систему, общую схему разработки, общий набор документов с их содержанием и определить их как обязательные для всех АС;

• определить также одну общую понятийную и терминологическую систему, обобщенный комплекс системных требований, набор кри­териев качества, но предоставить максимальную свободу в выборе схемы разработки, состава документов и других аспектов, наложив только минимум обязательных требований, которые позволяли бы:

— определять уровень качества результата;

— выбирать те конкретные методики (с их моделями ЖЦ, набором документов и др.), которые наиболее подходят к условиям разработ­ки и соответствуют используемым информационным технологиям;

— работать, таким образом, с минимальными ограничениями эф­фективных действий проектировщика АС.

Разработчики комплекса стандартов 34 выбрали способ, близкий к первому из указанных выше, т.е. пошли по пути, более близкому к схе­мам конкретных методик, чем к стандартам типа ISO12207. Тем не менее благодаря общности понятийной базы стандарты остаются применимы­ми в весьма широком диапазоне случаев.

Степень адаптивности формально определяется возможностями:

• опускать стадию эскизного проектирования и объединять стадии «Технический проект» и «Рабочая документация»;

• опускать этапы, объединять и опускать большинство документов и их разделов;

• вводить дополнительные документы, разделы документов и работы;

• динамически создавая так называемые ЧТЗ — частные технические задания — достаточно гибко формировать ЖЦ АС; как правило, этот прием используется на уровне крупных единиц (подсистем, комплексов), ради которых считается оправданным создавать ЧТЗ, однако нет никаких существенных оснований сильно ограничи­вать этот способ управления ЖЦ.

Стадии и этапы, выполняемые организациями — участниками работ по созданию АС, устанавливаются в договорах и техническом задании, что близко к подходу ISO.

Введение единой, достаточно качественно определенной терминологии, наличие достаточно разумной классификации работ, документов, видов обес­печения и т.д., безусловно, полезно. ГОСТ 34 способствует более полной и качественной стыковке действительно разных систем, что особенно важно в условиях, когда разрабатывается все больше сложных комплексных АС, например, типа САD-САМ, включающих в свой состав АСУТП, АСУП, САПР-конструктора, САПР-технолога, АСНИ и другие системы.

Определено несколько важных положений, отражающих особеннос­ти АС как объекта стандартизации, например: «в общем случае АС состо­ит из программно-технических (ПТК), программно-методических (ПМК) комплексов и отдельных компонентов организационного, технического, программного и информационного обеспечения».

Разделение понятий ПТК и АС закрепляло принцип, по которому АС есть не «ИС с БД», но:

• «организационно-техническая система, обеспечивающая выработку решений на основе автоматизации информационных процессов в различных сферах деятельности (управление, проектирование, про­изводство и т.д.) или их сочетаниях» (по РД 50-680-88), что особен­но актуально в аспектах бизнес-реинжиниринга;

• «система, состоящая из персонала и комплекса средств автомати­зации его деятельности, реализующая информационную техноло­гию выполнения установленных функций» (по ГОСТ 34.003-90).

Эти определения указывают на то, что АС — это в первую очередь персонал, принимающий решения и выполняющий другие управляющие действия, поддержанный организационно-техническими средствами.

Степень обязательности:

• прежняя полная обязательность отсутствует, материмы ГОСТ 34 по сути стали методической поддержкой, причем чаще для заказчи­ков, имеющих в стандарте набор требований к содержанию ТЗ и проведению испытаний АС. При этом польза ГОСТ 34 может мно­гократно возрасти в случае их более гибкого использования при формировании профиля ЖЦ АС.

Ключевым документом взаимодействия сторон является ТЗ — техни­ческое задание на создание АС. ТЗ — основной исходный документ для создания АС и его приемки, ТЗ определяет важнейшие точки взаимодей­ствия заказчика и разработчика. При этом ТЗ разрабатывает организа­ция-разработчик (по ГОСТ 34.602-89), но формально выдает ТЗ разра­ботчику заказчик (по РД 50-680-88).

**5. Государственные стандарты РФ (ГОСТ Р)**

В РФ действует ряд стандартов в части документирования ПС, разра­ботанных на основе прямого применения международных стандартов ИСО. Это — самые «свежие» по времени принятия стандарты. Некоторые из них прямо адресованы руководителям проекта или директорам информа­ционных служб. Вместе с тем они неоправданно малоизвестны в среде профессионалов. Вот их представление.

**ГОСТ Р ИСО/МЭК 9294-93. Информационная технология. Руководство по управлению документированием программного обеспечения**

Стандарт полностью соответствует международному стандарту ИСО/ МЭК ТО 9294:1990 и устанавливает рекомендации по эффективному уп­равлению документированием ПС для руководителей, отвечающих за их создание. Цель стандарта — оказание помощи в определении стратегии документирования ПС; выборе стандартов по документированию; выбо­ре процедур документирования; определении необходимых ресурсов; со­ставлении планов документирования.

**ГОСТ Р ИСО/МЭК 9126-93. Информационная технология. Оценка програм­мной продукции. Характеристики качества и руководства по их применению**

Стандарт полностью соответствует международному стандарту ИСО/ МЭК 9126:1991. В его контексте под характеристикой качества понима­ется «набор свойств (атрибутов) программной продукции, по которым ее качество описывается и оценивается». Стандарт определяет шесть комп­лексных характеристик, которые с минимальным дублированием описы­вают качество ПС (ПО, программной продукции):

* функциональные воз­можности;
* надежность;
* практичность;
* эффективность;
* сопровождаемость;
* мобильность.

Эти характеристики образуют основу для дальнейшего уточ­нения и описания качества ПС.

***ГОСТ Р ИСО 9127-94. Системы обработки информации. Документа­ция пользователя и информация на упаковке для потребительских програм­мных пакетов***

Стандарт полностью соответствует международному стандарту ИСО 9127:1989. В контексте настоящего стандарта под потребительским программным пакетом понимается «программная продукция, спроектиро­ванная и продаваемая для выполнения определенных функций; програм­ма и соответствующая ей документация, упакованные для продажи как единое целое». Под документацией пользователя понимается документа­ция, которая обеспечивает конечного пользователя информацией по ус­тановке и эксплуатации ПП. Под информацией на упаковке понимают информацию, воспроизводимую на внешней упаковке ПП. Ее цель — предоставление потенциальным покупателям первичных сведений о ПП.

***ГОСТ Р ИСО/МЭК 8631-94. Информационная технология. Програм­мные конструктивы и условные обозначения для их представления***

Описывает программные конструктивы, состоящие из набора одной или более процедурных частей и управляющей части, которая может быть задана неявно представленным процедурным алгоритмом.

#### 6. Процессы жизненного цикла программных средств

**(ГОСТ Р ИСО/МЭК 12207-99)**

Первая редакция ISO/IЕС 12207: 1995-08-01 подготовлена в 1995г. объединенным техническим комитетом ISO/IЕС JТС1 «Информационные технологии, подкомитет SC7, проектирование программного обеспечения».

В 1999 г. он введен в России и странах СНГ - **ГОСТ Р ИСО/МЭК 12207:99**.

По определению, ISO 12207 — базовый стандарт процессов ЖЦ ПО, ориентированный на различные (любые!) виды ПО и типы проектов АС, куда ПО входит как часть. Стандарт определяет стратегию и общий поря­док в создании и эксплуатации ПО, он охватывает ЖЦ ПО от концепту­ализации идей до завершения ЖЦ.

Очень важные ***ЗАМЕЧАНИЯ СТАНДАРТА***:

1) процессы, используемые во время ЖЦ ПО, должны быть со­вместимы с процессами, используемыми во время ЖЦ АС. (Отсюда понятна целесообразность совместного использования стандартов на АС и на ПО);

2) добавление уникальных или специфических процессов, действий и задач должно быть оговорено в контракте между сторонами. Контракт понимается в широком смысле: от юридически оформленного контракта до неформального соглашения, соглашение может быть определено и единственной стороной как задача, поставленная самому себе;

3) стандарт принципиально не содержит конкретных методов дей­ствий, тем более — заготовок решений или документации. Он описывает архитектуру процессов ЖЦ ПО, но не конкретизирует в деталях, как реа­лизовать или выполнить услуги и задачи, включенные в процессы, не предназначен для предписывания имени, формата или точного содержи­мого получаемой документации. Решения такого типа принимаются ис­пользующим стандарт.

**ОПРЕДЕЛЕНИЯ СТАНДАРТА**:

1. *Система* — комплекс, состоящий из процессов, технических и про­граммных средств, устройств и персонала, обладающий возможностью удов­летворять установленным потребностям или целям.

2. *Модель жизненного цикла* — структура, состоящая из процессов, работ и задач, включающих в себя разработку, эксплуатацию и сопро­вождение программного продукта, охватывающая жизнь системы от ус­тановления требований к ней до прекращения ее использования. Мно­жество процессов и задач сконструировано так, что возможна их адапта­ция в соответствии с проектами ПО. Процесс адаптации — процесс ис­ключения процессов, видов деятельности и задач, не применимых в кон­кретном проекте. Степень адаптивности — максимальная.

3. *Квалификационное требование* — набор критериев или условий, которые должны быть удовлетворены для того, чтобы квалифицировать про­граммный продукт на соответствие установленным требованиям и готов­ность к использованию в заданных условиях эксплуатации.

Стандарт не предписывает конкретную модель ЖЦ или метод разра­ботки ПО, но определяет, что стороны — участники использования стан­дарта ответственны за выбор модели ЖЦ для проекта ПО, за адаптацию процессов и задач стандарта к этой модели, за выбор и применение мето­дов разработки ПО, за выполнение действий и задач, подходящих для проекта ПО.

Стандарт ISO 12207 равносильно ориентирован на организацию дей­ствий каждой из двух сторон: поставщик (разработчик) и покупатель (пользователь); может быть в равной степени применен, когда обе сторо­ны — из одной организации.

**Каждый процесс ЖЦ разделен на набор действий, каждое действие — на набор задач.** Очень важное отличие ISO: каждый процесс, действие или задача инициируются и выполняются другим процессом по мере не­обходимости, причем нет заранее определенных последовательностей (ес­тественно, при сохранении логики связей по исходным сведениям задач и т.п.). Все процессы жизненного цикла изображены на рис. 1.

В стандарте ISO 12207 описаны:

1) пять основных процессов ЖЦ ПО:

• ***процесс приобретения***. Определяет действия предприятия-покупателя, которое приобретает АС, программный продукт или сервис ПО;

• ***процесс поставки***. Определяет действия предприятия-поставщика, снабжающего покупателя системой, программным продуктом или сервисом ПО;

• ***процесс разработки***. Определяет действия предприятия-разработ­чика, которое разрабатывает принцип построения программного изделия и программный продукт;

• ***процесс функционирования***. Определяет действия предприятия-опе­ратора, обеспечивающего обслуживание системы (а не только ПО) в процессе ее функционирования в интересах пользователей. В от­личие от действий, определяемых разработчиком в инструкциях по эксплуатации (эта деятельность разработчика предусмотрена во всех трех рассматриваемых стандартах), также определяются действия опе­ратора по консультированию пользователей, получению обратной связи и т.д., которые он планирует сам и берет на себя соответству­ющие обязанности;

• ***процесс сопровождения***. Определяет действия персонала сопровож­дения, который обеспечивает сопровождение программного про­дукта, что представляет собой управление модификациями про­граммного продукта, поддержку его текущего состояния и функ­циональной пригодности, включает в себя инсталляцию и удале­ние программного изделия из состава вычислительной системы;

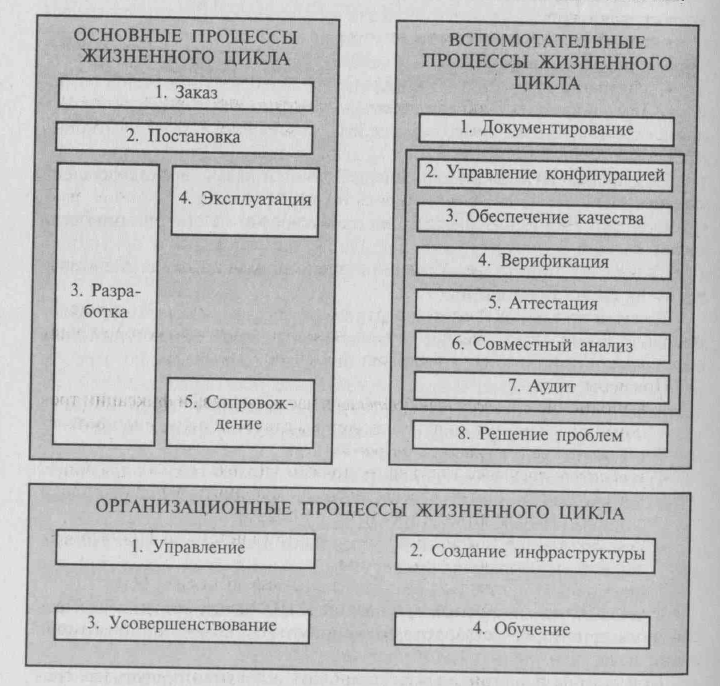


Рис. 1. **Процессы жизненного цикла программных средств**

2) восемь вспомогательных процессов, которые поддерживают реа­лизацию другого процесса, будучи неотъемлемой частью всего ЖЦ про­граммного изделия, и обеспечивают должное качество проекта ПО:

• **решения проблем;**

**• документирования;**

**• управления конфигурацией;**

**• гарантирования качества**, использующий результаты остальных про­цессов группы обеспечения качества, в которую входят:

**- верификация;**

**- аттестация;**

**- совместная оценка;**

**- аудит;**

3) четыре организационных процесса:

• **управления;**

**• создания инфраструктуры;**

**• усовершенствования;**

**• обучения.**

К ним примыкает особый процесс адаптации, определяющий основ­ные действия, необходимые для адаптации стандарта к условиям конк­ретного проекта.

Под процессом усовершенствования здесь понимается не усовершен­ствование АС или ПО, а улучшение самих процессов приобретения, раз­работки, гарантирования качества и т.п., реально осуществляемых в орга­низации.

Каких-либо этапов, фаз, стадий не предусмотрено, что дает описыва­емую ниже степень адаптивности.

Динамический характер стандарта определяется способом определе­ния последовательности выполнения процессов и задач, при котором один процесс при необходимости вызывает другой или его часть.

Примеры:

• выполнение процесса приобретения в части анализа и фиксации тре­бований к системе или ПО может вызывать исполнение соответ­ствующих задач процесса разработки;

• в процессе поставки поставщик должен управлять субподрядчика­ми согласно процессу приобретения и выполнять верификацию и аттестацию по соответствующим процессам;

• сопровождение может требовать развития системы и ПО, что вы­полняется по процессу разработки.

Такой характер позволяет реализовывать любую модель ЖЦ.

При выполнении анализа требований к ПО предусмотрены 11 клас­сов характеристик качества, которые используются позже при гарантиро­вании качества.

При этом разработчик должен установить и документировать как тре­бования к программному обеспечению:

1) функциональные и возможные спецификации, включая исполне­ние, физические характеристики и условия среды эксплуатации, при которых единица программного обеспечения должна быть выполнена;

2) внешние связи (интерфейсы) с единицей программного обеспечения;

3) требования квалификации;

4) спецификации надежности, включая спецификации, связанные с методами функционирования и сопровождения, воздействия ок­ружающей среды и вероятностью травмы персонала;

5) спецификации защищенности;

6) человеческие факторы спецификаций по инженерной психологии (эргономике), включая связанные с ручным управлением, взаимо­действием человека и оборудования, ограничениями на персонал и областями, нуждающимися в концентрированном человеческом внимании, которые являются чувствительными к ошибкам чело­века и обучению;

7) определение данных и требований базы данных;

8) установочные и приемочные требования поставляемого программ­ного продукта в местах функционирования и сопровождения (экс­плуатации);

9) документация пользователя;

10) работа пользователя и требования выполнения;

11) требования сервиса пользователя.

(Интересно и важно, что эти и аналогичные характеристики хорошо корреспондируются с характеристиками АС, предусматриваемыми в ГОСТ 34 по видам обеспечения системы.)

Стандарт содержит предельно мало описаний, направленных на про­ектирование БД. Это можно считать оправданным, так как разные систе­мы и разные прикладные комплексы ПО могут не только использовать весьма специфические типы БД, но и не использовать.

Итак, ISO 12207 имеет набор процессов, действий и задач, охватыва­ющий наиболее широкий спектр возможных ситуаций при максималь­ной адаптируемости.

Он показывает пример того, как должен строиться хорошо организо­ванный стандарт, содержащий минимум ограничений (принцип «нет оди­наковых проектов»). При этом детальные определения процессов, форм документов и т.п. целесообразно выносить в различные функциональные стандарты, ведомственные нормативные документы или фирменные ме­тодики, которые могут быть использованы или не использованы в конк­ретном проекте.

По этой причине центральным стандартом, положения которого бе­рутся за начальный «стержневой» набор положений в процессе построе­ния профиля стандартов ЖЦ для конкретного проекта, полезно рассмат­ривать именно ISO 12207. Этот «стержень» может задавать модель ЖЦ ПО и АС, принципиальную схему гарантирования качества, модель уп­равления проектом.

Практики используют еще один путь: сами переводят и используют в своих проектах современные стандарты на организацию ЖЦ ПС и их документирование. Но этот путь страдает как минимум тем недостатком, что разные переводы и адаптации стандартов, сделанные разными разра­ботчиками и заказчиками, будут отличаться массой деталей. Эти отличия неизбежно касаются не только наименований, но и их содержательных определений, вводимых и используемых в стандартах. Таким образом, на этом пути неизбежно постоянное возникновение путаницы, а это прямо противоположно целям не только стандартов, но и любых грамотных методических документов.

**7. Создание и сопровождение программных средств и информационных систем**

Констатация того, что программные разработки не укладываются в сроки и бюджет, а качество производимого продукта оставляет желать лучшего, стала общим фактом. То, что возможный путь решения проблем лежит в упорядочении процессов разработки на основе впитавших в себя мировой опыт стандартов, менее известно, а то, что эти стандарты мож­но освоить и с пользой применять на небольших отечественных пред­приятиях, верят единицы.

Рассмотрим возможность применения схематической интер­претации базовых элементов стандарта SPICE (Спайс).

Большинство стандартов, даже будучи международными, имеют аме­риканское происхождение. Отечественные программисты, работающие на американских заказчиков, уже обратили внимание на национальные особенности стилей мышления. Американец большей частью мыслит индуктивно, собирая факты и складывая их один за другим, мало забо­тясь об их упорядоченности (корова, белая корова, коза, теленок...). Нам же, по-видимому, вследствие математического тренинга, пройденного еще в средней школе, необходима некая общая схема (вид — пол — возраст­ная группа — цвет). К сожалению, многие перечни (на сотни пунктов) в стандартах выглядят подобным образом. Для того, чтобы стандарт при­жился на отечественной почве, необходимо его не просто перевести, но и представить в некоторой структурно упорядоченной форме.

Схематически интерпретируем базовые элементы стандарта SPICE (Software Process Improvement Capability dEtermination), официально именуе­мого ISO/IEC TR 15504 — «Оценка и аттестация зрелости процессов создания и сопровождения программных средств и информационных систем». В отли­чие от других стандартов ISO, он открыт (см. *http://www.sqi.gu.edu.au/spice*).

Стандартов, регламентирующих программные процессы и качество программ, а также способы их усовершенствования, довольно много.

SPICE — попытка объединить наиболее значимые из них. Это прежде всего:

* **ISO 12207, регламентирующий процессы жизненного цикла про­граммного обеспечения;**
* **СММ, определяющий модель зрелости процесса разработки ПО;**
* **стандарты серии ISO 9000, рассматривающие проблемы управления качеством,**

а также ряд национальных стандартов и нормати­вов крупных компаний. В результате, несмотря на отсылки к смежным стандартам ISO, при уточнении деталей объем SPICE превысил 500 стра­ниц. Если же учесть, что являющаяся сердцевиной стандарта модель про­граммных процессов содержит перечни и таблицы в сотни пунктов, то становится понятно, почему его сторонятся менеджеры проектов.

Основные цели SPICE — помощь потребителям (заказчикам) про­граммной продукции в выборе надежного поставщика и поддержка по­ставщика (разработчика) в его стремлении усовершенствовать процессы разработки.

Для достижения поставленной цели предлагается оценить, как ведется работа. Оценка, в свою очередь, производится путем сравне­ния с эталонной моделью (фактически той же, но несколько менее де­тальной, что и в ISO 12207). Рассмотрим модель и оценочные показатели, на основе которых производится сравнение: это, с одной стороны, наи­более сложные, а с другой — ключевые для понимания стандарта в целом компоненты. Остальное в основном связано с установкой рейтингов, под­бором команды оценщиков и т.п., все это наглядно и толково сопровожда­ется иллюстративными примерами из жизни и более связано с общими проблемами управления, чем со спецификой программирования.

Предваряя рассмотрение, необходимо сделать замечание относитель­но терминологии. Впервые встречающиеся русскоязычные термины из стандарта выделены курсивом, а вводимые классификационные термины при первом упоминании подчеркнуты.

**Эталонная модель SPICE**. Деятельность по созданию и приобретению программного продукта или услуги в соответствии с эталонной моделью SPICE представляет собой взаимодействующие процессы без каких-либо ограничений на последовательность. Каждый процесс должен включать в себя некоторые базовые операции (base practice). Процессы оцениваются в зависимости от степени организации (управления):

• 0 — не выполняется;

• 1 — выполняется неформально;

• 2 — планируется и контролируется;

• 3 — четко определяется;

• 4 — количественно регулируется;

• 5 — постоянно совершенствуется.

Для того чтобы оценить уровень процесса, проверяется наличие у него некоторых общих свойств, формулируемых в терминах обобщенных операций (generic practice). Такая операция представляет собой действия, уместные для любого процесса: выполнение, планирование, фиксация состояния, подготовка методики, использование количественных оценок, обучение персонала, распределение ответственности и т.п.

В SPICE заявлена наглядная форма, иллюстрирующая отношения про­цессов и их базовых операций к обобщенным операциям. Форма пред­ставляет собой таблицу, в столбцах которой размещаются категория про­цесса, потребитель, а в строках — уровень, общие свойства, обобщенная операция. Тот факт, что некоторый процесс X использует обобщенную операцию Y (крестик на пересечении соответствующего столбца и строки), означает наличие определенной базовой операции в этом процессе. К сожа­лению, в стандарте не дается заполненной таблицы. Вместо этого базовые операции приведены списками для соответствующих процессов. Ясное со­поставление базовых операций обобщенным отсутствует. Более того, неко­торые из обобщенных операций вынесены в базовые операции служебных (supporting) процессов (документирование, управление конфигурациями и др.). По-видимому, исторически сложившийся и унаследованный из ран­них стандартов список базовых операций не был упорядочен в соответствии с позднее выдвинутой наглядной и компактной двумерной формой.

В более наглядной форме уровни возможностей процессов можно представить охватывающими операцию слоями операций, способствую­щих ее успеху (рис. 2).

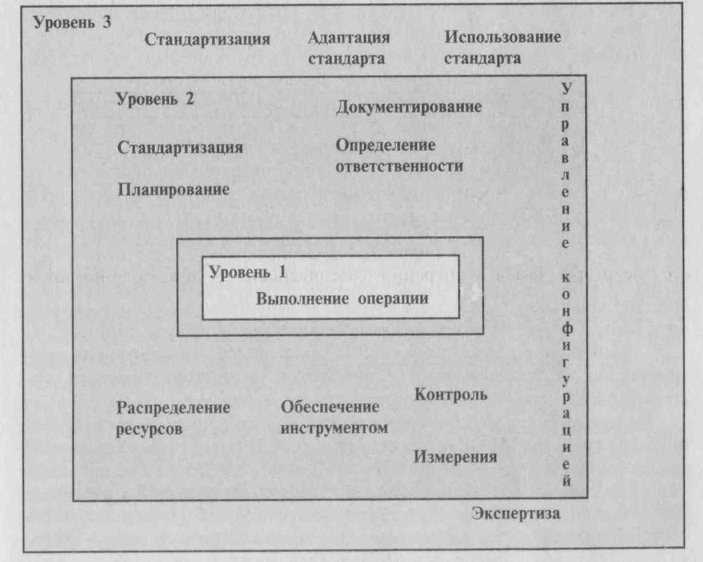


Рис. 2. **Уровни процессов**

В качестве центральной может быть рассмотрена любая операция, как непосредственно связанная с разработкой программы, так и из окружаю­щих ее слоев. Например, операция контроля требует документирования, обеспечения инструментом, ресурсами и т.д. Таким образом, предложен­ная модель определяет значительно более широкий спектр операций, чем список базовых операций SPICE; в частности, возможны стандарты на разработку стандартов или контроль операций контроля. На практике (осо­бенно в начале упорядочения) вполне хватает и одной итерации.

Модель была бы более наглядной, если бы в качестве обобщенной операции была выделена операция «Обеспечение связи/коммуникации». Чрезвычайно важным свойством SPICE является то, что вслед за ISO 12207 в явной форме рассмотрены проблемы взаимоотношения различных сто­рон. Обсуждение контракта, приемка-сдача готового продукта, определе­ние взаимоотношений между разработчиками, информирование об из­менениях, а также многие другие операции — конкретные формы реали­зации этой операции. Такого рода операции уместно отнести ко второму уровню возможностей.

**Рабочие продукты SPICE**. Основным инструментом оценки процес­сов в соответствии со SPICE являются их показатели (process indicators). Для оценки адекватности процесса или операции предлагается исследо­вать наличие и содержание рабочих продуктов (work product), составляю­щих его вход/выход. К сожалению, как список рабочих продуктов (состо­ит из 109 неупорядоченных пунктов), так и двадцатистраничная таблица соответствия «базовая операция — входные/выходные рабочие продук­ты» в SPICE представлены в неудобоваримом сложном виде.

Усугубляет ситуацию то, что часть из перечисляемых в общем списке рабочих продуктов носит конкретный характер, а другие представляют обобщенные свойства. В частности, имеется рабочий продукт под назва­нием «Рабочий продукт», а есть «План вообще» и конкретные планы. Для того чтобы понять, какие рабочие продукты используются в SPICE, ра­зумно их классифицировать, причем классифицировать в двух смыслах: общеметодическом (разбить на группы) и в программистском (сопоста­вить общую структуру и единообразные механизмы работы с ними).

1. *Инженерные рабочие продукты*. Первую группу, очевидную для про­граммистов, не сталкивавшихся с проблемами управления проектами, со­ставляют инженерные рабочие продукты.

Цель разработки в соответствии со SPICE — создание Системы. Очень полезное свойство SPICE — явное разделение Системы и Программного продукта (Software). Собственно Программный продукт составляет лишь часть Системы, в которую, кроме того, входят оборудование, персонал, средства инфраструктуры. Разработчики с советским стажем вспомнят соответствующие этому разделению стандарты на автоматические систе­мы и программное обеспечение. SPICE выделяет в целевые рабочие про­дукты некоторые части Системы (Компонент системы, Интегрированный программный продукт, Клиентская документация, Тестовый план клиентской документации).

Чтобы достичь поставленной цели, разработка (инженерные процес­сы) детализирует и формализует исходную информацию, закрепляя про­межуточные результаты в рабочих продуктах, озаглавленных Требования, Проект (Design) [Общий/Детальный (High/Low Level)].

В SPICE перечислены не все возможные инженерные рабочие про­дукты, причем это касается не только иерархии целевой Системы, пред­ставленной избранными компонентами, но и промежуточных докумен­тов. Например, выделяется Проект Базы данных (Database Design), но ничего не говорится относительно целевой Базы данных и относительно проектов, выделяемых компонентов Системы. Полную картину всех воз­можных инженерных продуктов может представить таблица, в строках ко­торой перечисляются элементы иерархии целевой Системы, а в графах — уровни разработки (Требования, Проекты, Реализация). Клетки таблицы будут соответствовать возможным рабочим продуктам.

2. *Управленческие рабочие продукты*. Создание (приобретение) програм­много продукта проводится группой людей в некоторые сроки с исполь­зованием определенного оборудования. Выделение работ, сроков, распре­деление между персоналом, обеспечение ресурсами — все это фиксируют управленческие рабочие продукты: Планы, Графики, Обязательства и пр. Рабочие продукты этого класса являются результирующими для опера­ций, соотнесенных с обобщенными операциями Планирование, Опреде­ление ответственности, Распределение ресурсов.

3. *Оперативные рабочие продукты*. Оперативные рабочие продукты — документы, фиксирующие некоторые факты, в частности, как соотносится текущее состояние разрабатываемых продуктов и их окружения с ожидае­мым, наличие дефектов, проблем, запросов на изменение, предложений, ответов и пр. В терминах SPICE — это прежде всего всевозможные Записи (Records), а также Отчеты (Reports), Протоколы собраний и т.д., которые можно рассматривать как агрегаты Записей. Запись содержит определен­ное число полей, большинство из которых имеет четко очерченный набор значений. Часть из них определяет контекст: дату, автора, ссылочные до­кументы (продукты). Другие представляют оценочные значения, причем, чтобы обеспечить однозначность в интерпретации (избежать сравнения «неплохой» с «нормальный»), необходима нормативная регламентация ис­пользуемых значений. С точки зрения обобщенных операций оператив­ные рабочие продукты прежде всего представляют результаты Контроля. Суть оперативных рабочих продуктов — в передаче информации для све­дения (принятия решения, исправления, ответа) от одного лица к другому (другим), а также ее фиксации (для памяти). Таким образом, они составля­ют предмет опеки для выделенных нами операций коммуникации.

4. *Нормативно-методические рабочие продукты*. Создание не только отдельных полей оперативных документов, но и любых рабочих продук­тов будет эффективным, если предоставить для него соответствующую нормативно-методическую поддержку. Нормативно-методические рабо­чие продукты определяют стандарты содержания и оформления осталь­ных рабочих продуктов, а также стратегию и регламент выполнения ра­бот по их созданию. К ним можно отнести все документы, озаглавлен­ные: Стандарт, Методология, Политика, Стратегия, Измерение.

5. *Конфигурационные рабочие продукты*. Состав сложных рабочих продуктов, история их изменения, а также информационные и причин­но-следственные связи между ними задаются при помощи конфигура­ционных рабочих продуктов. Они озаглавливаются: Список, Отображе­ние (Mapping). С конфигурационными рабочими продуктами имеют дело не только операции Управления конфигурацией систем в том смысле, в котором они упоминаются в SPICE, но и все операции, для которых существен сложный состав рабочих продуктов.

6. *Инструментальные рабочие продукты*. В качестве входных/выходных продуктов для базовых операций в SPICE перечислен ряд инструменталь­ных рабочих продуктов (инструментов): Коммуникационный механизм, Хранилище повторно используемых объектов, Средства управления кон­фигурацией систем. Представленный список, конечно же, не исчерпывает всех используемых в разработке инструментов, но выделяет наиболее важ­ные с точки зрения оценки организованности проведения работ.

Перечислим отношения между рабочими продуктами: обобщение — конкретизация; целое — часть; исходный — результирующий документ опе­рации; предыдущая — следующая версия/вариант; задание — результат вы­полнения задания; объект — результат контроля; методика — результат при­менения методики; инструмент — формируемый инструментом продукт.

Первые два неявно наблюдались уже в исходном списке SPICE (на­пример, План — План Проекта, Система — Компонент). Все остальные представляют различные варианты явно определяемого SPICE отноше­ния вход-выход базовой операции. Такое разделение позволяет существен­но упростить описание соответствующих операций и в то же время созда­ет условия для более полного отражения функционирования программ­ных процессов. В частности, появляется возможность отследить для каж­дой операции, как формулируется задание, контролируется результат, какие методики и инструменты используются.

Большинство программных процессов организуются как действия над рабочими продуктами:

• разработка (проектирование, реализация или изменение);

• контроль (рассмотрение, оценка, верификация, тестирование, аудит);

• коммуникация (распространение, согласование, инсталляция, замена);

• отслеживание (мониторинг);

• хранение (формирование версий и историй, обеспечение доступа).

Оставшиеся действия — это действия с персоналом (подбор, обуче­ние, управление) и оборудованием (приобретение/подготовка и обеспе­чение работоспособности).

В операциях (в том числе в базовых операциях SPICE) возможно при­сутствие нескольких действий при превалировании одного. Например, при создании некоторого продукта может выявиться наличие дефектов в исходных данных, что соответствует их контролю. Действия по разработ­ке характерны для инженерных процессов, планированию, подготовке методик, инструментов. Их результат подвергается изменениям.

Результат действий по контролю, под которыми понимается не толь­ко аудит или обсуждение (review), но и тестирование, представляет фак­ты, фиксируемые здесь и сейчас. Он сохраняется, рассылается, по изме­нению уже не подлежит. В случае тестирования тестовые примеры и сце­нарии рассматриваются как компоненты нормативно-методического обес­печения, подлежащие разработке специальной операцией.

На рис. 3 представлены операции разработки и контроля, пунк­тиром отмечены отношения между рабочими продуктами.

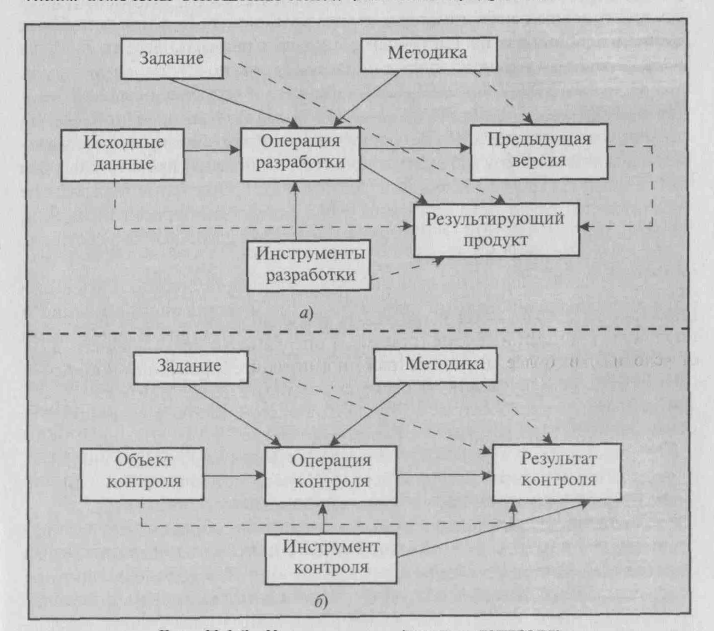


Рис. 3. **Операции разработки и контроля**:

а) — отношения: операция разработки — рабочий продукт

б) — отношения: операция контроля — рабочий продукт

Классификация рабочих продуктов, определение их отношений, а также дифференциация входов-выходов базовых операций имеют своей целью не только создание базы для структурированного изложения SPICE, но и наглядного пособия, призванного помочь практикам-программис­там анализировать свою работу.

Многие программисты пользуются объектно-ориентированными языками значительно успешнее, чем языком естественным. Когда, пы­таясь навести порядок в потоке экстремальных ситуаций в программи­ровании, менеджер проекта предлагает зафиксировать задания в проект­ном списке, обычно выдается текст, состоящий в основном из англий­ских аббревиатур расширений файлов вместо схемы: «Дано — требуется осуществить». Необходимо исходный документ X с помощью версии N обработать так, чтобы получить результирующий документ Y версии М, который отвечал бы запросам на изменения Z1, Z2, ... Результирующий документ должен соответствовать стандарту S версии К, а при разработке должен использоваться инструмент I конфигурации IС версии J. Такая запись позволяет точно и полно описать ситуацию.

Стандарт SPICE может быть представлен в структурированной ком­пактной форме. Тогда накопленные в нем рекомендации и типовые ре­шения, способствующие успеху программных разработок, могут эффек­тивнее применяться участниками проектов. Основным препятствием для этого является отсутствие единой терминологии. Отдельные переводы не помогают, пока не появится некий консорциум заинтересованных лиц, организаций, в том числе государственных, который мог бы обсудить, согласовать общий глоссарий.

Следующим шагом на пути внедрения стандарта должно стать выде­ление специализированных организаций, осуществляющих обучение и консультации. Наивно полагать, что заваленные текущими проблемами менеджеры проектов самостоятельно его освоят. Нужны стимулы и ме­роприятия на уровне организации. Чем дальше в рынок, тем больше та­ких стимулов. Типичный пример: если заказчик не доволен принимае­мым продуктом, а поставщик считает, что он сделал работу наилучшим образом, значит, в контракте не был четко оговорен критерий приемки. Поставщик тратит лишние ресурсы и нервы на выправление ситуации, заказчик следующий проект делает в другом месте.

Наконец, ключевую проблему — повышение общей культуры про­граммистской работы — необходимо решать на уровне воспитания, вводя соответствующие курсы в программы обучения. К сожалению, никто не рассказывает студентам о том, как трудно взаимодействовать с заказчи­ком, сколь неоднозначны могут быть слова и как быстро они забывают­ся, сколько рисков таит в себе программная разработка и как с этим всем бороться, в том числе, опираясь на опыт мировых стандартов.

#### 8. Рекомендации по выбору базовых стандартов

Есть ряд причин, стимулировавших разработку новых поколений стандартов. **SPC** (Software Productivity Consortium) относит к ним «*труд­ность гармоничного сочетания и интеграции таких дисциплин, как наука, проектирование, менеджмент и финансы*». Не менее важны причины, ха­рактерные для новейшего времени: резко возросшая изменчивость усло­вий работы систем и требований к ним, возросшее многообразие условий их разработки и сопровождения, распределенность и глобализация систем, и даже текучесть кадров ИТ-специалистов в условиях, когда программист, отладчик или системный инженер стали массовыми профессиями [42].

SPC отмечает, что в этих условиях понадобились новые базовые стан­дарты типа framework, созданные для того, чтобы «*улучшить общение и кооперацию между разными дисциплинами и вспомогательными системами, чтобы создавать, использовать [системы] и руководить [этими процесса-ми] в интегрированном, согласованном стиле*»*.*

В то же время SPC указывает на избыточное число основополагаю­щих стандартов такого типа и уровня, характеризуя ситуацию словом тря­сина (quagmire), и делает ряд предупреждений и рекомендаций. Обраща­ется внимание на то, что в разных стандартах происходит консолидация моделей (в первую очередь модели процессов), что при этом объем моде­лей растет (не только за счет описания большего числа процессов, но и за счет приведения дополнительных рекомендаций по применению), а так­же что использование новых моделей и передового опыта по трудоемко­сти сравнимо с накоплением собственных «уроков».

SPC выделяет тот минимум стандартов на процессы проектирования, который рекомендуется взять за основу. В их число включены ISO/IEC 12207, ISO/IEC 15288 СD2, ISO 15504 (SPICE), EIA/ANSI 632, EIA/IS 731 (SECM), TickIT:

• ISO/IЕС 12207, Information technologi — Software life cycle processes. 1995;

• ISO/IЕС ТR 15271, Information technologi – Guide for ISO/IЕС 12207. 1998. (Стандарт ISO/IЕС 12207 оказал революционизирующее вли­яние на многие другие НД, в том числе на стандарты моделей си­стемного проектирования: процессы жизненного цикла систем, модель зрелости процессов);

• ЕIА/АNSI 632, Processes for Engineering а Sistem. 1999. (Этот стан­дарт не только заменил ряд популярных более старых американ­ских стандартов, но был использован как вклад американской груп­пы в создание ISO/IЕС 15288);

. ЕIА/IS-731, Sistem Engineering Capability Model (SECM). 1999. Part 1, SECM Model. Part 2, SECM Appraisal Method. (В области стандар­тов на уровни зрелости процессов аналогично тому, как модель SW CMM переросла в модель и стандарт SPICE, модель SE СММ переросла в модель и стандарт SECM);

. ISO/IЕС 15288 СD2, Life Cycle Management – System Life Cycle Processes. 2000;

• ISO/IЕС ТR 15504, SPICE – Software Process Improvement Capability dEtermination. 1998 («Оценка и аттестация зрелости процессов со­здания и сопровождения программных средств и информацион­ных систем»).

Для обеспечения преемственности полезно добавить в эту группу стан­дарты ГОСТ 34 (не гармонизированные с новыми, но применимые и полезные из-за совместимости по многим базовым понятиям, по сути многих работ, по опыту применения и др.).

Существенно, что два потока стандартов — на SE (system engineering) и на SW (software engineering), развивавшихся параллельно, четко стыко­ваны посредством указанных документов. И дело не только в том, что указанные НД хорошо согласованы друг с другом по основным понятиям и принципам. Очень важно, что такие, казалось бы, чисто технические области, как создание ПО (SW-процессы), регламентированы стандарта­ми, прямо требующими их *совместного применения* со стандартами на процессы системного проектирования (SE-процессы).

#### 9. Заключение

В качестве заключения отметим, что:

• требования к техническим частям (ПО, аппаратура) должны соот­носиться с требованиями к системе и с потребностями в ее приоб­ретении (нельзя замыкаться в требованиях к ИС);

• в организациях должен вестись новый объем управленческой и ме­тодической работы:

— соединение бизнес-слоя и ИТ-слоя систем;

— создание и совершенствование моделей ЖЦС для этой органи­зации и для каждого проекта;

— формирование комплексного стандарта уровня предприятия и уровня проекта с включением в него НД на процессы и стандар­тов на языки и интерфейсы ИТ и др.;

— сохранение в стандарте уровня предприятия достаточной гибко­сти, для того, чтобы он мог в нужных пределах адаптироваться под проекты и не становился тормозящим фактором;

— такая работа может вестись сверху вниз (от основных базовых международных стандартов к стандартам проекта), снизу вверх, но лучше - с применением обоих этих подходов;

— **использование лишь даже отдельных положений новых стандар­тов в реальных проектах приносит (и уже приносило) несомнен­ную пользу.**