

Министерство образования и науки Российской Федерации
НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Л.А. КОРШИКОВА

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И СТАНДАРТИЗАЦИЯ

Утверждено Редакционно-издательским советом университета
в качестве учебного пособия

НОВОСИБИРСК
2018

УДК 004.057.2(075.8)
К 704

Рецензенты:

канд. техн. наук, доц. *Е.Л. Романов*
канд. физ.-мат. наук, доц. *Ю.А. Котов*

Работа подготовлена на кафедре вычислительной техники
для студентов АВТФ всех форм обучения по направлениям 09.04.01 –
Информатика и вычислительная техника
и 09.04.04 – Программная инженерия

Коршикова Л.А.

К 704 Информационные технологии и стандартизация: учебное пособие / Л.А. Коршикова. – Новосибирск: Изд-во НГТУ, 2018. – 76 с.

ISBN 978-5-7782-3545-8

В пособии содержатся основные сведения о стандартах в области информационных технологий. В настоящее время информационные технологии находят широкое применение во всех сферах жизнедеятельности человека. От правильности работы программных средств во многом зависят результаты труда и безопасность тысяч людей. Уделено большое внимание стандартизации программных средств и процессов их разработки, прогнозированию, оценке и управлению качеством программных средств на протяжении всего их жизненного цикла.

Предназначено для специалистов в области программного обеспечения, использования в дисциплинах, изучающих принципы разработки и построения программных продуктов.

УДК 004.057.2(075.8)

ISBN 978-5-7782-3545-8

© Коршикова Л.А., 2018
© Новосибирский государственный
технический университет, 2018

ПРЕДИСЛОВИЕ

В последние годы во всем мире особое внимание уделяется стандартизации программных средств и процессов их разработки, прогнозированию, оценке и управлению качеством программных средств на протяжении всего их жизненного цикла.

Имеется множество разнообразных стандартов, процедур, методов, инструментальных средств и типов операционной среды для разработки и управления программным обеспечением. Это разнообразие создает трудности при проектировании и управлении программным обеспечением, особенно при объединении программных продуктов и сервисных программ. Стратегия разработки программного обеспечения требует перехода от этого множества к общему порядку, который позволит специалистам, практикующимся в программном обеспечении, «говорить на одном языке» при разработке и управлении программным обеспечением.

Контролирующие материалы, представленные в учебном пособии, – контрольные вопросы – позволяют оперативно провести промежуточный и итоговый контроль знаний.

1. ВВЕДЕНИЕ В ПРЕДМЕТНУЮ ОБЛАСТЬ

Понятие «Стандартизация» определяется как **принятие соглашения по спецификации, производству и использованию аппаратных и программных средств вычислительной техники; установление и применение стандартов, норм, правил.**

Стандарты занимают все более значительное место в направлении развития индустрии информационных технологий. Более 250 подкомитетов в официальных организациях по стандартизации работают над стандартами в области информационных технологий. Множество стандартов или уже приняты этими организациями, или находятся в процессе разработки [3, 14].

Классификация стандартов

На рис. 1 представлена классификация стандартов, принятая официальными организациями по стандартизации, работающих в области информационных технологий.

Пояснения к рис. 1:

стандарт **«де-факто»** – термин, обозначающий продукт какого-либо поставщика, который захватил большую долю рынка и который другие поставщики стремятся эмулировать, копировать или использовать, для того чтобы захватить свою часть рынка;

стандарт **«де-юре»** создается формально признанной стандартизирующей организацией. Он разрабатывается при соблюдении правил консенсуса в процессе открытой дискуссии, в которой каждый имеет шанс принять участие.

Ни одна группа не может действовать независимо, создавая стандарты для промышленности. Если какая-либо группа поставщиков создаст стандарт, не учитывающий требований пользователей, она потерпит неудачу.

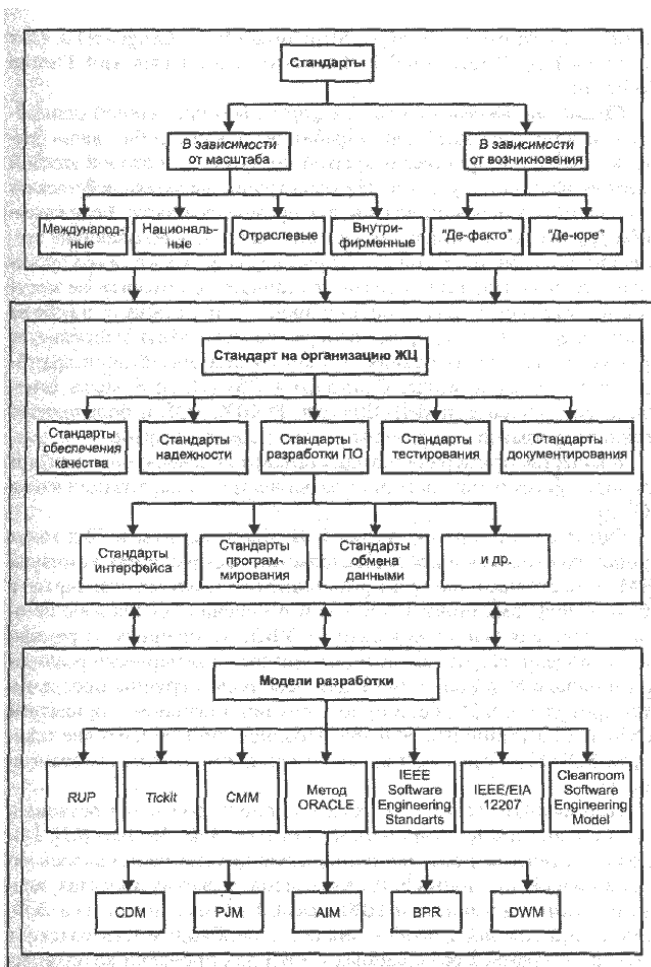


Рис. 1. Классификация стандартов

В качестве примера перехода стандарта «де-факто» в стандарт «де-юре», как правило, рассматривается история развития и стандартизации языка SQL.

Например, даты разработки стандартов языка SQL следующие:

- 1975–1980 – прототипы, основанные на SQL,
- 1980 – появление первых коммерческих SQL-продуктов,

- 1985 – работа комитета по стандартам над SQL,
- 1986 – SQL –86,
- 1989 – SQL – 89,
- 1990 – работа над SQL – 2, SQL – 3,
- 200.. – принятие следующих версий SQL.

В области информационных технологий существуют два основных исторически сложившихся подхода к разработке стандартов:

первый, когда назревает проблема в необходимости стандарта;

во втором случае – компании-разработчики программного обеспечения (ПО) разрабатывают каждая свое решение, и самое популярное, массовое, с точки зрения частоты использования, решение обретает статус стандарта.

Требования к стандартам

С точки зрения пользователя, все многообразие ПО должно управляться **единообразно**: должна быть единообразная **навигация** – перемещение по программе, единообразные **органы управления ПО и единая реакция программного обеспечения на действия пользователя**. Для этого разработаны стандарты на пользовательский интерфейс – GUI (Graphical User Interface). Все это регламентируется стандартами, действующими в сфере информационных технологий.

Необходимость стандартизации разработки программного обеспечения наиболее удачно описана во введении в стандарт ISO/ IEC 12207:

«**Программное обеспечение** является неотъемлемой частью информационных технологий и традиционных систем, таких как транспортные, военные, медицинские и финансовые. Имеется множество разнообразных стандартов, процедур, методов, инструментальных средств и типов операционной среды для разработки и управления программным обеспечением. Это разнообразие создает трудности при проектировании и управлении программным обеспечением, особенно при объединении программных продуктов и сервисных программ. Стратегия разработки программного обеспечения требует перехода от этого множества к общему порядку, который позволит специалистам, практикующимся в программном обеспечении, “говорить на одном языке” при разработке и управлении программным обеспечением. Этот международный стандарт обеспечивает такой общий порядок».

Виды нормативных документов

Нормативный документ – документ, устанавливающий правила, общие принципы или характеристики, касающиеся различных видов деятельности или их результатов.

Правило (ПР) – документ, устанавливающий обязательные для применения организационно-технические или общетехнические положения, порядки, методы выполнения работ.

Рекомендации (Р) – документ, содержащий добровольные для применения организационно-технические или общетехнические положения, порядки, методы выполнения работ.

Норма – положение, устанавливающее количественные или качественные критерии, которые должны быть удовлетворены.

Регламент – документ, содержащий правовые нормы и принятый органом власти.

Общероссийский классификатор технико-экономической и социальной информации (ОКТЕСИ) – официальный документ, представляющий собой систематизированный свод наименований кодов классификационных группировок и (или) объектов классификации в области технико-экономической и социальной информации.

Существуют стандарты, зависящие от сферы действия, например:

- международный стандарт;
- региональный стандарт;
- государственный стандарт РФ (ГОСТ-Р);
- межгосударственный стандарт (ГОСТ);
- стандарт отрасли (ОСТ);
- стандарт научно-технического или инженерного общества (СТО);
- стандарт предприятия (СТП).

К техническим регламентам относятся:

- законодательные акты;
- постановления правительства РФ, содержащие требования, нормы и технические характеристики;
- государственные стандарты;
- нормы и правила федеральных органов исполнительной власти.

Основные правила стандартизации следующие:

- сбалансированность интересов сторон, разрабатывающих, изготавливающих, потребляющих определенный продукт;

- системность и комплексность стандартизации – рассмотрение каждого объекта как части более сложной системы;
- динамичность и опережающее развитие стандартизации – необходимо учитывать возможность появления новых изделий и новых технологических процессов;
- эффективность стандартизации – экономический или социальный эффект за счет экономии ресурсов, повышения надежности, повышения технической и информационной совместимости;
- приоритетность разработки стандартов, способствующих безопасности, совместимости и взаимозаменяемости.

Приоритетные для РФ направления стандартизации информационных технологий

Направления 1-го приоритета:

- языки программирования и программный интерфейс;
- языки описания документов;
- программная инженерия;
- сервисы управления данными;
- работа в сетях и соответствующие соединения;
- безопасность информационных технологий;
- терминология.

Направления 2-го приоритета:

- сбор данных и системы идентификации;
- мультимедиа и представление информации;
- пользовательский интерфейс;
- офисное оборудование;
- кодирование набора символов.

Направления 3-го приоритета:

- среды для информационного обмена;
- геоинформационные технологии – повышение качества электронных карт и соответствие их требованиям международных стандартов.

Приоритетными для РФ являются и стандарты на организацию жизненного цикла (ЖЦ) программного продукта, например:

- стандарты обеспечения качества,
- стандарты надежности,
- стандарты разработки ПО,
- стандарты тестирования,
- стандарты документирования,
- стандарты интерфейса,
- стандарты программирования,
- стандарта обмена данными.

Контрольные вопросы

1. Что такое стандартизация?
2. В чем заключается необходимость стандартизации?
3. Назовите виды нормативных документов.
4. Назовите основные направления 1-го приоритета.

2. ПОНЯТИЕ ОТКРЫТЫХ СИСТЕМ

Развитие систем и средств вычислительной техники привели к необходимости объединения конкретных вычислительных устройств и реализованных на их основе информационных систем в единые информационно-вычислительные системы и среды, к формированию единого информационного пространства.

Единое информационное пространство – это:

- информационные ресурсы,
- организационные структуры,
- средства информационного взаимодействия.

Информационные ресурсы:

- данные,
- сведения,
- информация и знания.

Структуры организации систем и сред должны выполнять функции поиска, сбора, обработки, хранения, защиты и передачи информации.

Средства информационного взаимодействия

- программно-аппаратные средства и пользовательские интерфейсы,
- правовые и организационно-нормативные документы, обеспечивающие доступ к информационным ресурсам на основе соответствующих информационно-коммуникационных технологий.

Концепция открытых систем

Одним из первых шагов в концепции открытых систем, оказавших влияние на развитие отечественной вычислительной техники, явилось создание компьютеров серии IBM 360.

Следующие этапы связаны с областью интерактивной обработки и увеличением объема продуктов, для которых требуется переносимость, например, управление распределенными базами данных.

Конец же 1970-х гг. характеризуется массовым применением сетевых технологий. Организация ISO в 1977–1978 гг. развернула интенсивные работы по созданию стандартов взаимосвязи в сетях открытых систем. Тогда же впервые было введено определение открытой информационной системы.

Таким образом, решение проблем совместимости и мобильности привело к разработке большого числа международных стандартов и соглашений в сфере применения информационных технологий и разработки информационных систем. Основопологающим, базовым, понятием при использовании стандартов стало понятие «открытая система».

Определение открытой системы Национальным институтом стандартов и технологий США (NIST) следующее: «Открытая система – это система, которая способна взаимодействовать с другой системой посредством реализации международных стандартных протоколов».

Открытую систему сегодня определяют как исчерпывающий и согласованный набор международных стандартов на информационные технологии и профили функциональных стандартов, которые реализуют **открытые спецификации** на интерфейсы, службы и поддерживающие их форматы, чтобы обеспечить взаимодействие (интероперабельность) и мобильность программных приложений, данных и персонaла [4].

Принцип формирования открытых систем состоит в создании среды, включающей в себя программные и аппаратные средства, службы связи, интерфейсы, форматы данных и протоколы. Такая среда в основе имеет развивающиеся доступные и общепризнанные стандарты и обеспечивает значительную степень взаимодействия (Interoperability), переносимости (Portability) и масштабирования (Scalability) приложений и данных.

Термин «открытая спецификация» – «это общедоступная спецификация, которая поддерживается открытым, гласным, согласитель-

ным процессом, направленным на постоянную адаптацию новой технологии, и которая соответствует стандартам».

Примеры использования технологии открытых систем:

- технологии фирм «Intel» Plug&Play и USB,
- операционная система UNIX,
- операционная система Windows NT,
- язык программирования Java фирмы «Sun Microsystems».

Общие свойства открытых информационных систем

Общие свойства открытых информационных систем – следующие:

1) взаимодействие/интероперабельность – способность к взаимодействию с другими прикладными системами;

2) стандартизуемость – ИС проектируются и разрабатываются на основе согласованных международных стандартов и предложений;

3) расширяемость/масштабируемость – возможность перемещения прикладных программ и передачи данных в системах и средах;

4) мобильность/переносимость – обеспечение возможности переноса прикладных программ и данных при модернизации или замене аппаратных платформ ИС;

5) дружелюбность к пользователю – развитые унифицированные интерфейсы в процессах взаимодействия в системе «пользователь – компьютерное устройство – программное обеспечение», позволяющие работать пользователю, не имеющему специальной системной подготовки.

Открытым системам присущи следующие черты:

- реализация открытости осуществляется на основе функциональных стандартов (профилей) в области информационных технологий;

- информационные системы, обладающие свойством открытости, могут выполняться на любых технических средствах, которые входят в единую среду открытых систем;

- открытые системы предполагают использование унифицированных интерфейсов в процессах взаимодействия в системе «человек – компьютер»;

- применение положений открытости предполагает некоторую избыточность при разработке программно-аппаратных комплексов.

Стандарты в области информационных технологий (ИТ) определяют:

- глобальные концепции развития области ИТ;
- концептуальный базис и эталонные модели построения основных разделов ИТ;
- функции, протоколы взаимодействия, интерфейсы и другие аспекты ИТ;
- языки программирования, языки спецификации информационных ресурсов, языки управления базами данных;
- модели технологических процессов создания и использования систем ИТ, а также языки описания таких моделей;
- методы тестирования соответствия (конформности) систем ИТ исходным стандартам и профилям;
- методы и процедуры функционирования собственно системы стандартов ИТ;
- метаязыки и нотации для описания стандартов ИТ;
- общесистемные функции ИТ, например, безопасность, администрирование, интернационализация, качество сервисов и пр.

Особенности состояния и развития стандартизации

Основные положения состояния стандартизации и стандартов:

- существует несколько сотен международных и национальных стандартов создания ИС;
- длительные сроки разработки, согласования и утверждения стандартов;
- стандарты современных ИС должны учитывать необходимость построения ИС как открытых систем;
- в области ИС функциональными стандартами поддержаны только массовые процессы (телекоммуникация, программирование, документирование программ и данных);
- не поддерживаются требованиями стандартов сложные и творческие процессы создания крупных распределенных ИС (системный анализ и проектирование, интеграция компонентов и систем, испытания и сертификация ИС и т. д.);
- пробелы и задержки в подготовке и издании стандартов высокого ранга;

- последующие селекция, совершенствование и согласование нормативных и методических документов в ряде случаев позволяют создать на их основе национальные и международные стандарты.

На рис. 2 приводится информация о разработчиках стандартов в области ИТ.

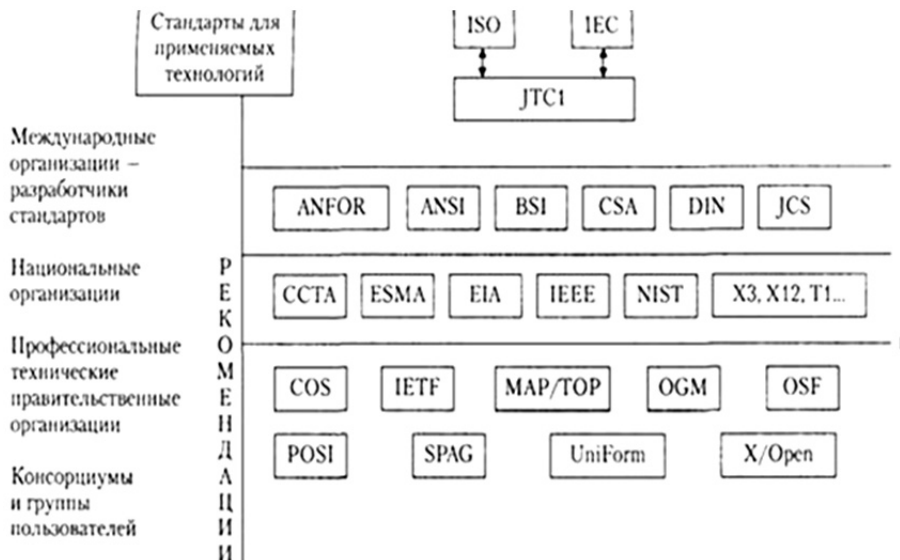


Рис. 2. Разработчики стандартов

Существуют и специализированные профессиональные организации:

- IEEE (Институт инженеров по электротехнике и электронике – международная организация – разработчик ряда важных международных стандартов в области ИТ);

- CEN (Европейский комитет стандартизации широкого спектра товаров, услуг и технологий, в том числе связанных с областью разработки ИТ, аналог ISO);

- CENELEC (Европейский комитет стандартизации решений в электротехнике, в частности, стандартизации коммуникационных кабелей, волоконной оптики и электронных приборов – аналог IEC);

- ETSI (Европейский институт стандартизации в области сетевой инфраструктуры – аналог ITU-T);

- OMG (Группа объектно-ориентированного управления – крупнейший международный консорциум, осуществляющий разработку стандартов для создания унифицированного распределенного объектного программного обеспечения, включающий в себя свыше 600 компаний – производителей программного продукта, разработчиков прикладных систем и конечных пользователей);

- ECMA (Европейская ассоциация производителей вычислительных машин – международная ассоциация, целью которой служит промышленная стандартизация информационных и коммуникационных систем) и др.

Отдельно выделенный Комитет JTC1 выполняет следующие функции:

- проектирование и разработка систем и средств ИТ;
- решение проблем производительности и качества продуктов и систем ИТ:
 - безопасности систем ИТ и информации;
 - переносимости прикладных программ;
 - интероперабельности продуктов и систем ИТ;
 - унифицированным средствам и окружениям;
 - гармонизированному словарю понятий области ИТ;
 - дружеским и эргономичным пользовательским интерфейсам.

Существуют Подкомитеты и группы JTC1, связанные с разработкой стандартов ИТ:

- C2 – символьные наборы и кодирование информации;
- SC6 – телекоммуникация и информационный обмен между системами;
- SC7 – разработка программного обеспечения и системная документация;
- SC18 – текстовые и офисные системы;
- SC21 – открытая распределенная обработка (Open Distributed Processing – ODP), управление данными (Data Management – DM) и взаимосвязь открытых систем (Open System Interconnection – OSI);
- SC22 – языки программирования, их окружения и интерфейсы системного программного обеспечения;
- SC24 – компьютерная графика;
- SC27 – общие методы безопасности для ИТ-приложений;
- SGFS – специальная группа по функциональным стандартам.

Вывод: характерная особенность стандартов ИТ состоит в том, что они содержат определения основных понятий и терминов области ИТ, описания моделей, сценариев, функций, правил поведения и представления информации. По существу, в стандартах ИТ свойства систем ИТ представляются в виде концептуальных, функциональных, информационных моделей объектов стандартизации.

Контрольные вопросы

1. Дайте определение понятия открытых систем.
2. Назовите концепцию открытых систем.
3. Дайте определение термина «открытая спецификация».
4. Назовите несколько черт, присущих открытым системам.

3. МЕТОДОЛОГИЧЕСКИЙ БАЗИС ОТКРЫТЫХ СИСТЕМ

Определение методологического базиса открытых систем:

«Информационные технологии включают в себя спецификацию, проектирование и разработку систем и средств, имеющих дело со сбором, представлением, обработкой, безопасностью, передачей, организацией, хранением и поиском информации, а также обменом и управлением информацией – база реализована в виде методологического базиса открытых систем».

При становлении базиса открытых систем наблюдалось:

- создание международных организаций по разработке и стандартизации открытых систем;
- разработка эталонных моделей и соответствующих им базовых спецификаций;
- разработка и широкое использование концепции профиля;
- разработка таксономии профилей, представляющей собой классификационную систему ИТ/ИС и обеспечивающую систематическую идентификацию профилей в пространстве ИТ/ИС;
- разработка концепции и методологии соответствия реализаций ИТ/ИС тем спецификациям, которые ими реализуются.

Состав многоуровневой модели пространства спецификаций

В состав многоуровневой модели пространства спецификаций включены следующие понятия:

- профили приложений;
- OSE – профили платформ (систем);
- OSE-профили;

- локальные профили (OSE-профили, API-профили);
- базовые спецификации;
- архитектурные спецификации (эталонные модели);
- предметные, или локальные, например: OSI-профили, API-профили, профили, разрабатываемые на основе базовых спецификаций;
- OSE-профили спецификации поведения открытых систем на их границах (интерфейсах);
- полные OSE-профили открытых платформ и систем – спецификации, предназначенные для описания поведения ИТ-систем;
- OSE-профили прикладных технологий – полная спецификация окружений прикладных технологий обработки данных (например, банковских систем, распределенных офисных приложений и т.п.);
- стратегические профили (например, International Standardized Profiles – IPS, Government Open System Interconnection Profile – GOSIP) – наборы стандартов.

Итак, архитектурные спецификации (эталонные модели) применяются для формирования концептуального базиса и определения семантической структуры важнейших разделов ИТ.

Эталонные модели определяют архитектуру наиболее важных и достаточно независимых разделов ИТ.

Ниже перечислены примеры эталонных моделей:

- базовая эталонная модель взаимосвязи открытых систем (Basic Reference Model for Open Systems Interconnection – RM-OSI);
- руководство по окружению открытых систем POSIX (Portable Operating System Interface for Computer Environments – RM API);
- эталонная модель для открытой распределенной обработки (Reference Model for Open Distributed Processing – RM-ODP);
- эталонная модель управления данными (Reference Model for Data Management – RM DF);
- эталонная модель компьютерной графики (Reference Model of Computer Graphics – RM CG);
- эталонная модель текстовых и офисных систем (Text and Office Systems Reference Model);
- модель конформности (Coformality – соответствия, подобия) и методы тестирования конформности, называемые также методами аттестационного тестирования;
- модель основ общей безопасности (Generic Security Frameworks);
- модель качества OSI-сервиса (Quality of Service for OSI).

Пример модели Open Systems Environment – OSE

Состав прикладных программ OSE следующий:

- системы реального времени (Real Time System – **RTS**) и **встроенные системы (Embedded System – ES)**;
- системы обработки транзакций (Transaction Processing System – **TPS**);
- системы управления базами данных (DataBase Management System – **DBM**);
- разнообразные системы поддержки принятия решения (Decision Support System – **DSS**);
- управленческие информационные системы административного (Executive Information System – **EIS**) и **производственного (Enterprise Resource Planning – ERP) назначения**;
- географические информационные системы (Geographic Information System – **GIS**);
- другие системы, в которых могут применяться рекомендуемые международными организациями спецификации.

Модель Open Systems Environment — OSE используется для систем, которые:

- выполняются на любой используемой платформе поставщика или пользователя;
- используют любую ОС;
- обеспечивают доступ к базе данных и управление данными;
- обмениваются данными и взаимодействуют через сети любых поставщиков и в локальных сетях потребителей.

Итак, OSE поддерживает переносимые, масштабируемые и взаимодействующие прикладные компьютерные программы через стандартные функциональности, интерфейсы, форматы данных, протоколы обмена и доступа.

При этом положения следующие:

- стандартами могут быть международные, национальные и другие общедоступные спецификации и соглашения;
- прикладные программы и средства OSE **переносимы, если они реализованы на стандартных платформах и написаны на стандартизованных языках программирования**;
- прикладные программы и средства OSE **масштабируемы в среде различных платформ и сетевых конфигураций**;

- **прикладные программы и средства OSE взаимодействуют друг с другом, если они предоставляют услуги пользователю, используя стандартные протоколы, форматы обмена данными и интерфейсы систем совместной или распределенной обработки данных.**

В описании Open Systems Environment – OSE OSE/RM используется два типа элементов:

- логические объекты, включающие в себя ППО, прикладные платформы и внешнюю функциональную среду;
- интерфейсы, содержащие интерфейс прикладной системы и интерфейс обмена с внешней средой.

Логические объекты представлены тремя классами, интерфейсы – двумя.

Прикладная платформа состоит из совокупности программно-аппаратных компонентов, реализующих системные услуги, которые используются прикладным программным обеспечением (ППО).

Внешняя среда платформ состоит из элементов, внешних по отношению к ППО и прикладной платформе (например, внешние периферийные устройства, услуги других платформ, операционных систем или сетевых устройств).

Интерфейс прикладной программы (Application Program Interface – API) является интерфейсом между ППО и прикладной платформой. Основная функция API – поддержка переносимости ППО.

Интерфейс обмена с внешней средой (External Environment Interface – EEI) обеспечивает передачу информации между прикладной платформой и внешней средой.

Эталонная модель OSE/RM реализует и регулирует взаимоотношения «поставщик – пользователь». Логические объекты прикладной платформы и внешней среды являются поставщиком услуг, ППО – пользователем.

Классификация базовых спецификаций

Ниже перечисляются функции базовых спецификаций OSE.

Функции управления базами данных включают в себя:

- язык баз данных SQL (Structured Query Language);
- информационно-справочная система IRDS (Information Resource Dictionary System);

- протокол распределенных операций RDA (Remote Database Access);
- PAS Microsoft на открытый прикладной интерфейс доступа к базам данных ODBC API.

Функции пользовательского интерфейса, которые включают в себя следующие стандарты:

- MOTIF из OSF для графического пользовательского интерфейса;
- стандарт OPEN LOOK;
- X Window вместе с GUI и телекоммуникациями;
- стандарты для виртуального терминала (Virtual Terminal – VT);
- стандарты машинной графики GKS (Graphical Kernel System);
- GKS-3D (Graphical Kernel System - 3 Dimentional);
- PHIGS (Programmers Hierarchical Interactive Graphics System);
- CGI (Computer Graphics Interface).

Функции взаимосвязи открытых систем, включающие в себя:

- спецификации сервиса и протоколов, разработанные в соответствии с моделью OSI (рекомендации серии X.200);
- стандарты для локальных сетей (IEEE 802) [IEEE Std 802-1990];
- спецификации сети Интернет [Transmission Control Protocol (TCP) – RFC 793.

Функции распределенной обработки, включая следующие базовые спецификации OSI:

- вызов удаленной процедуры RPC (Remote Procedure Call);
- фиксация, параллельность и восстановление CCR (Commitment, Concurrency and Recovery);
- протокол надежной передачи (RT);
- обработка распределенной транзакции DTP (Distributed Transaction Processing);
- управление файлами, доступ к файлам и передача файлов FTAM (File Transfer, Access and Management);
- управление открытыми системами (OSI Management);
- API для доступа к сервису Object Request Broker (ORB) в архитектуре CORBA и API, определяющий базовые возможности такого сервиса (Common Object Services – COS);
- язык спецификации интерфейсов объектов IDL (Interface Definition Language) и его проекции на объектно-ориентированные языки.

Существуют спецификации специальных сервисных элементов прикладного уровня модели OSI, стандартов Internet OMG, X/Open, например:

- служба справочника (The Directory – X.500);
- спецификации распределенных приложений с архитектурой «клиент-сервер» и распределенных объектных приложений.

В OSE используются следующие структуры данных и документов, форматы данных:

- спецификация сообщений и электронных данных для электронного обмена в управлении, коммерции и транспорте EDIFACT (Electronic Data Interchange for Administration, Commence and Trade);
- средства языка ASN. 1 (Abstract Syntax Notation One), предназначенного для спецификации прикладных структур данных – абстрактного синтаксиса прикладных объектов;
- форматы метафайла для представления и передачи графической информации CGM (Computer Graphics Metafile);
- спецификации документов – спецификации структур учреждений документов ODA (Open Document Architecture);
- языки описания документов гипермедиа и мультимедиа, например: HTML (Hypertext Markup Language); HyTime, SMDI»(Standard Music Description Language), SMSL (Standard Multimedia/Hypermedia Scripting Language), SPDS (Standard Page Description Language), DSSSL (Document Style Semantics and Specification Language);
- спецификации структур документов для производства, например SGML (Standard Generalized Markup Language);
- спецификация форматов графических данных, например форматов JPEG, JBIG и MPEG

В OSE, кроме того, рассматриваются спецификации инструментальных окружений (в частности, языков реализации и их библиотек) и CASE-окружений (например, ISO/IEC DIS 13719, ECMA Portable Common Tool Environment).

Идентификация спецификаций производится по следующим элементам: имя (наименование) спецификации, дата публикации (дата, когда спецификация стала доступной для общего использования), организация-спонсор (организация, ответственная за разработку и/или поддержание, и/или существование данной спецификации), примени-

мость, степень согласованности, доступность изделия, полнота, зрелость, стабильность, проблемы/ограничения, аттестационное тестирование, привязки, дальнейшие возможности развития, альтернативные спецификации.

Контрольные вопросы

1. Дайте определение понятия методологический базис открытых систем.
2. Назовите примеры эталонных моделей.
3. Для каких систем Open Systems Environment – OSE?
4. Приведите одну базовую спецификацию из классификации (перечислить состав).

4. ПОНЯТИЕ ПРОФИЛЯ ОТКРЫТОЙ СИСТЕМЫ

При создании и развитии сложных, распределенных, тиражируемых программных и информационных систем требуется:

- гибкое формирование и применение согласованных (гармонизированных) совокупностей базовых стандартов и нормативных документов разного уровня;
- выделение в них требований и рекомендаций, необходимых для реализации заданных функций ИС.

Профиль – это совокупность нескольких базовых стандартов с определенными подмножествами обязательных и рекомендуемых возможностей.

Эта совокупность предназначена для реализации заданной функции или группы функций ИТ/ИС в конкретной функциональной среде.

Примерами такой среды могут быть:

- среда рабочей станции;
- управления встроенными вычислительными устройствами;
- распределенная среда передачи и обработки данных;
- среда офисного документооборота и т.д.

Профиль не может противоречить использованным в нем базовым стандартам и нормативным документам.

Базовые стандарты и профили могут использоваться в качестве:

- непосредственных директивных,
- руководящих,
- рекомендательных документов,
- нормативной базы, необходимой при разработке средств автоматизации технологических этапов.

Основными целями применения профилей при создании и использовании ИС являются:

- снижение трудоемкости и повышение связности проектов ИС;
- обеспечение переносимости ППО;
- обеспечение расширяемости ИС по набору прикладных функций и масштабируемости;
- предоставление возможности функциональной интеграции в ИС задач, которые раньше решались отдельно и менее эффективно;
- повышение качества компонентов ИС.

Классификация профилей

На верхнем классификационном уровне находятся **международные стандартизованные профили (International Standardized Profiles – ISP)**, признанные соответствующим комитетом ИСО. В области международной стандартизации ИТ профили ISP имеют такой же статус, что и международные базовые стандарты, и предназначены для широкой области применения.

Определение профиля включает в себя следующие элементы:

- область действия функции, для которой определяется профиль;
- иллюстрация сценария, показывающего пример применения профиля, при этом желательно использование диаграммного представления ИТ-системы, самого приложения и имеющих место интерфейсов;
- нормативные ссылки на набор базовых стандартов или ISP, содержащие точную идентификацию актуальных текстов базовых спецификаций, а также охватывающие принятые дополнения и исправления;
- спецификации применения каждого цитируемого базового стандарта или ISP, устанавливающие выбор классов, подмножеств, опций, диапазонов значений параметров, а также ссылки на регистрируемые объекты;
- раздел, определяющий требования на соответствие данному профилю реализующих его ИТ-систем;
- ссылка на спецификацию аттестационных тестов для реализации данного профиля, если таковые имеются;
- информативные ссылки на любые полезные, желательно актуализированные документы.

Требования к содержанию и формату профиля ISR:

- профили должны быть непосредственно связаны с базовыми стандартами, и аттестация на соответствие профилю подразумевает аттестацию на соответствие этим базовым стандартам;
- ISP должен удовлетворять правилам ISO/IEC для представления проектов и самих международных стандартов;
- ISP должен быть компактным документом, не повторяющим текста документов, на которые он ссылается;
- определение одного профиля может включать в себя ссылки на определение других;
- многие профили документируются и публикуются в виде отдельных ISP;
- для каждого профиля должна обеспечиваться спецификация тестирования профиля (Profile Test Specification), которая определяется или как часть ISP, или как отдельный самостоятельный ISP.

В дополнение к ГОСТ Р ISO/IEC TR-10000-1 приводятся правила составления каждого из элементов ISP, соответствующие правилам ISO/IEC (в случае разбиения ISP на части каждая часть должна удовлетворять этой структуре).

Требования к содержанию и формату профиля APP

Ступенькой ниже в иерархии следуют **национальные профили, в соответствии с которыми должна строиться национальная система ИТ-стандартизации.**

Примерами наиболее «влиятельных» может служить профиль переносимости приложений APP (Application Portability Profile – APP-США), а также входящий в него государственный профиль взаимосвязи открытых систем GOSIP (Government Open System Interconnection Profile).

Все виды функционального обслуживания в рамках APP могут быть представлены следующими семью функциональными областями:

- функции, реализуемые операционной системой;
- функции, реализующие человекомашинные интерфейсы;
- поддержка разработки программного обеспечения;
- управление данными;
- обмен данными;
- компьютерная графика;
- сетевые функции.

Основные типы интерфейсов OSE

Далее рассмотрим основные типы интерфейсов OSE и их характеристики.

- API (Application Program Interface – интерфейс прикладной программы);
- CSI (Communication Services Interface – интерфейс коммуникационных услуг);
- HCI (Human/Computer Interface – человекомашинный интерфейс);
- ISI (Information Services Interface – интерфейс информационных услуг).

Под API понимается интерфейс между ППО и поставщиком необходимого для функционирования этого программного обеспечения сервиса, т.е. прикладной платформой.

Интерфейс CSI обеспечивает реализацию взаимодействия с внешними системами, которая осуществляется с помощью протоколов (процедур обмена).

Через интерфейс HCI осуществляется физическое взаимодействие пользователя и системы ИТ.

Интерфейс ISI рассматривается как граница взаимодействия с внешней памятью долговременного хранения данных.

Область функций операционной системы

Функции, реализуемые ОС, обеспечивают управление прикладной платформой, а также обслуживают интерфейсы для взаимодействия прикладных программ и платформы.

Функции ядра ОС – функции нижнего уровня, применяются для создания и управления процессами исполнения программ, генерации и передачи сигналов операционной системы, генерации и обработки сигналов системного времени, управления файловой системой и каталогами, управления и обработкой запросов ввода/вывода и обслуживании внешних устройств.

Команды и утилиты – механизмы для исполнения функций уровня оператора, такие как сравнение, печать и отображение содержимого файлов, редактирование файлов; поиск образцов; регистрация сообщений; перемещение файлов из каталога в каталог; сортировка данных;

исполнение командных строк и доступ к служебной информации системы.

Расширение реального времени – функции, реализующие прикладные и системные интерфейсы, которые используются в прикладных областях, требующих детерминированного исполнения, обработки и реакции. Расширения этого типа определяют прикладные интерфейсы к базовым функциям ОС: ввода/вывода, доступа к файловой системе и управления процессами.

Управление системой – функции, позволяющие создавать и управлять ресурсами, предоставляемыми пользователю, правами доступа к подсистемам и устройствам, управление конфигурацией и производительностью устройств, файловой системой, административными процессами, авторизацией доступа, поддержкой живучести системы.

Человекомашинные интерфейсы. Функции

Операции типа «клиент-сервер» – определяют взаимоотношения между процессом «клиент» и процессом «сервер» в сети, в частности, между процессами, имеющими место при отображении с помощью графического пользовательского интерфейса. В этом случае программа, которая управляет каждым дисплейным устройством, реализует процесс-сервер, в то время как пользовательская программа представляет процесс-клиент, который запрашивает обслуживание сервером.

Определение объектов и управление устройством включает в себя спецификации, с помощью которых задаются характеристики отображаемых элементов: цвет, форма, размеры, движение, графические характеристики, взаимодействие между отдельными элементами и т.д.

Параметры окон – спецификации, которые позволяют определить, как создаются окна, передвигаются, сохраняются, восстанавливаются, удаляются и взаимодействуют друг с другом.

Поддержка диалога – спецификации, с помощью которых устанавливаются взаимоотношения между тем, что отображено на экране (в том числе движение курсора, данные, введенные с клавиатуры и дополнительных устройств), и тем, как меняется изображение в зависимости от вводимых данных.

Спецификации мультимедиа, включая API, – спецификации, определения функций и форматов данных, которые поддерживают манипу-

ляции различными формами цифровой и аналоговой аудиовизуальной информации в рамках единой прикладной системы.

Программная инженерия. Функциональная область поддержки разработки программного обеспечения

Программная инженерия – это создание и применение мобильных, гибких, способных настраиваться на различные конфигурации аппаратных платформ, интероперабельных программных средств.

Функциональная область программной инженерии обеспечивает для этого необходимую инфраструктуру, в которую входят как языки программирования, так и интегрированные инструментальные системы для поддержки разработки программного обеспечения. В этой функциональной области можно выделить следующие средства.

- Стандартные языки программирования вместе со средой поддержки, в которую входят отладчики, средства настройки и оптимизации кода, редакторы [12].
- Интегрированные среды и инструментальные системы для разработки программного обеспечения, включающие в себя системы и программы его автоматизированного создания и поддержки.

Функциональная область управления данными

Центральной задачей большинства систем является управление данными. Системы управления данными реализуют следующие функции.

1. Обслуживание доступа к словарям и каталогам данных – обеспечивает управление доступом к структурированным данным и позволяют обращаться к комбинациям данных, расположенным в различных схемах.

2. Документы – спецификации для кодирования данных (текст, рисунки, числа, специальные символы и т.д.) как логические, так и визуальные структуры электронных документов.

3. Графические данные – независимые от устройств определения элементов рисунков.

4. Производственные данные – спецификации, описывающие технические рисунки, документацию, данные другого типа, включая геометрические или негеометрические данные, характеристики форм, допуски.

Ниже перечислены уровни сложности представления данных.

- Уровень 1 (формат данных) обеспечивает возможность представить данные, участвующие в обмене явным образом, путем указания формата либо ссылки на язык программирования.
- Уровень 2 (объект единого контекста) отображает содержание одиночного объекта. Примерами спецификаций такого типа могут быть тексты, растровые изображения или аудиоинформация.
- Уровень 3.
- Уровень 4 (семантика и синтаксис языков) – это уровень языка представления данных.
- Уровень 5 (прикладной) – уровень приложений, который может использовать любые из нижних уровней для обмена с другими прикладными программами.

Область графических функций

Эта область используется для создания и манипуляций с отображаемыми изображениями. Функции такого рода обеспечивают определение и поддержку отображаемого элемента и его атрибутов.

Функции этой области содержатся в спецификациях многомерных графических объектов и изображений в форме, независимой от конкретных устройств.

Доступность и целостность обеспечивается за счет средств, осуществляющих безопасность для данной области.

Функциональная область сетевой поддержки

Сетевые функции предоставляют механизмы для поддержки прикладных систем, которым требуется интероперабельный доступ к данным и программам в неоднородной сетевой среде. В этой области имеются следующие механизмы.

1. Собственно коммуникации, которые включают в себя API и спецификации протоколов для прозрачной и надежной передачи данных через коммуникационные сети.
2. Прозрачный доступ к файлам, расположенным в любом месте неоднородной сети.
3. Поддержка персональных ЭВМ и микро-ЭВМ для обеспечения интероперабельности с системами, базирующимися на различных ОС,

в частности, микрокомпьютерных, которые могут оказаться не соответствующими международным и национальным стандартам.

4. Дистанционное обращение к процедурам – спецификации для обращения к процедурам, расположенным во внешней распределенной среде.

Интегрально поддерживаемые функциональные области

Национальные профили GOSIP имеют Великобритания, Франция, Швеция, Япония, Австралия, Гонконг (Сянган). В январе 2000 г. государственный профиль взаимосвязи открытых систем России был утвержден Госстандартом (ГОСТ Р 50.1.22–2000).

Государственный профиль России имеет некоторые заметные отличия от GOSIP других стран.

Версии GOSIP объединяет функциональная идентичность по следующим обстоятельствам:

- проведение единой политики федеральных органов и правительственных департаментов по приобретению, внедрению и эксплуатации вычислительного и коммуникационного оборудования для реализации максимального информационного взаимодействия;
- уменьшение зависимости от поставщиков;
- обеспечение разработчиков, поставщиков и пользователей на федеральном уровне однозначно понимаемыми спецификациями, на основе которых должна формироваться стратегия разработки вычислительных и информационных систем, сетей и систем связи.

Характеристики GOSIP следующие:

- широкая применимость (активное использование не только соответствующими службами отдельных стран, но и международными организациями);
- доступность (реализации уже существуют либо имеются пилотные выпуски);
- стабильность ;
- эффективность (протоколы удовлетворяют общим потребностям федеральных органов и правительственных учреждений).

Эталонная модель среды открытых систем (OSE/RM) определяет разделение любой информационной системы на приложения (при-

кладные программы и программные комплексы) и среду, в которой эти приложения функционируют.

Общие свойства профилей

Полный OSE-профиль – это профиль, который специфицирует все поведение ИТ-системы или часть ее поведения на одном или большем числе интерфейсов OSE. Он состоит из выбранного набора открытых, общедоступных, согласованных стандартов и спецификаций, определяющих различные услуги в среде эталонной модели OSE/RM.

Профиль OSI – конкретный (локальный) профиль, составленный из базовых стандартов, соответствующих модели OSI (Open System Interconnection), и (или) базовых стандартов представления форматов и данных, т.е. F-профилей.

Замечания

- Профиль только ограничивает функциональность базовой спецификации благодаря выбору его опций и значений параметров. Таким образом, функциональность профилей вытекает из функциональности выбранных в них базовых стандартов.
- Профили не могут содержать никаких требований, противоречащих базовому стандарту, они лишь осуществляют выбор соответствующих опций и диапазонов значений параметров.

Для целей интеграции и совместного использования разнородных платформ в рамках распределенных систем ИТ необходима унификация и концептуальная целостность средств административного управления и управления информационной безопасностью систем ИТ.

Принято, что OSE-профиль должен обеспечивать:

- переносимость ППО и многократную используемость ПО на уровне исходного кода и стандартных библиотек (Application Software Portability and Software Reuse at the Source Code Level);
- интероперабельность прикладного программного обеспечения (Application Software Interoperability). Здесь имеется в виду возможность обмена данными между сущностями ПО, в том числе между сущностями, реализуемыми на разнородных прикладных платформах, а также возможность совместного использования ими обмениваемых данных;

- интероперабельность управления и безопасности (Management and Security Interoperability);
- использование существующих стандартов и аккомодацию к стандартам перспективных технологий (Accommodation of Standards). Профили OSE являются эффективным средством продвижения существующих стандартов в практику. Таким образом, на основе понятия OSE-профиля поддерживается такое свойство открытых систем, как адаптируемость к изменению стандартов;
- легкую настраиваемость на новые технологии создания информационных систем (Accommodation of New Information System Technology);
- гибкость аппарата OSE-профилей позволяет учитывать тенденции перехода к новым стандартам и соответственно к новым технологиям;
- масштабируемость прикладных платформ и распределенных систем (Application Platform Scalability);
- возможность разных типов реализаций некоторого OSE-профиля, отличающихся техническими и ресурсными характеристиками (например, суперкомпьютеры и рабочие станции), поддерживать одну и ту же функциональность, т.е. один и тот же набор сервисов;
- прозрачность реализаций процессов (Implementation Transparency). Данное свойство поддерживается благодаря систематическому использованию через аппарат OSE-профилей стандартизованных спецификаций (стандартов и ISPs), одним из принципов разработки которых является независимость от конкретных реализаций.

Для эффективного использования конкретного профиля необходимо:

- выделить объединенные логической связью проблемно-ориентированные области функционирования, где могут применяться стандарты, общие для одной организации или их группы;
- идентифицировать стандарты и нормативные документы, варианты их использования и параметры, которые необходимо включить в профиль;
- документально зафиксировать участки конкретного профиля, где требуется создание новых стандартов или нормативных документов, и идентифицировать характеристики, которые могут оказаться важными для разработки недостающих стандартов и нормативных документов этого профиля;

- формализовать профиль в соответствии с его категорией, включая стандарты, различные варианты нормативных документов и дополнительные параметры, которые непосредственно связаны с профилем.

Функциональный профиль

Пример компоновки функционального профиля

Пример ориентируется на открытые распределенные ИС с архитектурой «клиент-сервер».

Профиль среды ИС должен определять ее **архитектуру в соответствии с выбранной моделью распределенной обработки данных, например DCE (Distributed Computing Environment) или CORBA (Common Object Request Broker Architecture).**

В первом случае модель определяется стандартами Консорциума OSF, в частности, механизма удаленного вызова процедур **RPC (Remote Procedure Call)** с учетом стандартов де-факто, которые специфицируют применяемые мониторы транзакций (например, монитор транзакций Tuxedo).

Во втором случае модель определяется стандартами консорциума OMG, в частности, спецификацией брокера объектных запросов **ORB (Object Request Broker).**

Детализация профиля среды ИС производится на различных уровнях:

- графического пользовательского интерфейса (Motif консорциума «OSF» или стандарт X Window IEEE);
- реляционных или объектно-ориентированных СУБД (например, стандарт языка SQL-92 и спецификации доступа к разным базам данных);
- операционных систем с учетом сетевых функций, выполняемых на уровне ОС (например, набора стандартов POSIX – ISO и IEEE);
- телекоммуникационной среды в части услуг и сервисов прикладного уровня: электронной почты (по рекомендациям ITU-T X.400, X.500);
- доступа к удаленным базам данных RDA (по стандарту ISO; 9594-1.2), передачи файлов, доступа к файлам и управления файлами (по стандарту ISO 10607 - 1, 2, 3, 4, 5, 6);

- профиль среды распределенной ИС должен включать в себя стандарты протоколов транспортного уровня (по ISO OSI или стандарт де-факто протокола TCP/IP), стандарты локальных сетей (например, стандарт Ethernet IEEE 802.3 или стандарт Fast Ethernet IEEE 802.3 u), а также стандарты средств сопряжения проектируемой ИС с сетями передачи данных общего назначения (например, по рекомендациям ITU-T X.25,X.3,X.29).

Профиль защиты информации в распределенных системах

Профиль защиты информации в распределенных системах используется:

- для защиты от несанкционированного доступа, реализуемого на уровне ЦР промежуточного слоя;
- управления данными, реализуемыми СУБД;
- защиты программных средств, включая средства защиты от вирусов;
- защиты информации при обмене данными в распределенных системах, включая криптографические функции;
- администрирования средств безопасности.

Профиль инструментальных средств

Профиль инструментальных средств в распределенных системах используется:

- для контроля производительности и корректности функционирования системы в целом;
- преобразования конфигурации ППО, тиражирования версий;
- управления доступом пользователей к ресурсам системы и конфигурации ресурсов;
- перенастройки приложений в связи с изменениями прикладных функций ИС;
- настройки пользовательских интерфейсов (генерация экранных форм и отчетов); ведение баз данных системы;
- восстановления работоспособности системы после сбоев и аварий.

На основе профиля должно проводиться тестирование и сертификация приложений на соответствие требованиям открытости. Основу большинства применяемых в настоящее время профилей составляют стандарты серии POSIX. Общий их перечень состоит из более 45 наименований. Перечень российских стандартов в области реализации открытых систем составляет 92 стандарта.

Контрольные вопросы

1. Дайте определение понятия «профиль открытой системы».
2. Назовите все виды функционального обслуживания в рамках APP.
3. Назовите основные свойства и назначение профилей.
4. Назовите общие свойства профилей.

5. МЕТОДЫ И ТЕХНОЛОГИИ ПРОЕКТИРОВАНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Метод проектирования ПО представляет собой организованную совокупность процессов создания ряда моделей, которые описывают различные аспекты разрабатываемой системы с использованием четко определенной нотации [1].

Основные понятия и требования

Существует определенная совокупность составляющих метода проектирования:

- концепций и теоретических основ – в качестве таких основ могут выступать структурный или объектно-ориентированный подход;
- нотаций, используемых для построения моделей статической структуры и динамики поведения проектируемой системы – диаграммы потоков данных, диаграммы вариантов использования и др.

Технология проектирования определяется как совокупность технологических операций проектирования в их последовательности и взаимосвязи, приводящая к разработке проекта ПО.

Требования к технологии

Современная технология проектирования должна обеспечивать:

- соответствие стандарту ISO/IEC 12207: 1995 (поддержка всех процессов ЖЦ ПО);
- гарантированное достижение целей разработки ЭИС в рамках установленного бюджета, с заданным качеством и в установленное время;
- возможность декомпозиции проекта на составные части, разрабатываемые группами исполнителей ограниченной численности (3-7 чел.), с последующей интеграцией составных частей;

- минимальное время получения работоспособного ПО ЭИС. Речь идет не о сроках готовности всей ЭИС, а о сроках реализации отдельных подсистем. Практика показывает, что даже при наличии полностью завершенного проекта внедрение ЭИС идет последовательно по отдельным подсистемам;

- независимость получаемых проектных решений от средств реализации ЭИС;

- поддержку комплексом согласованных CASE – средств, обеспечивающих автоматизацию процессов, выполняемых на всех стадиях ЖЦ ПО.

Существуют обязательные стандарты для всех участников проекта, а именно:

- стандарт проектирования;
- стандарт оформления проектной документации;
- стандарт интерфейса конечного пользователя с системой.

Далее рассмотрим каждый из них.

Стандарт проектирования – это:

- набор необходимых моделей (диаграмм) на каждой стадии проектирования и степень их детализации;

- правила **именования объектов, набор атрибутов, правила оформления диаграмм;**

- требования конфигурации рабочих мест разработчиков, включая настройки ОС, настройки CASE-средств.

Стандарт оформления проектной документации включает в себя:

- комплектность, состав и структуру документации на каждой стадии проектирования;

- требования к оформлению документации;

- правила утверждения документации с указанием предельных сроков на каждой стадии; требования к настройке CASE – средств для обеспечения подготовки документации.

Стандарт интерфейса конечного пользователя с системой включает:

- правила оформления экранов (шрифты и цветовая палитра), состав и расположение окон и элементов управления;

- правила использования клавиатуры и мыши;
- правила оформления текстов помощи;
- перечень стандартных сообщений;
- правила обработки реакций пользователя.

При проектировании необходима сущность структурного подхода, которая подразумевает следующее.

Единственно эффективный подход к решению этой проблемы заключается в построении сложной системы из небольшого количества крупных частей, каждая из которых, в свою очередь, строится из частей меньшего размера и так до тех пор, пока самые небольшие части можно будет строить из имеющегося материала.

По отношению к проектированию сложной программной системы это означает, что ее необходимо разделять (декомпозировать) на небольшие подсистемы, каждую из которых можно разрабатывать независимо от других. Это позволяет при разработке подсистемы любого уровня держать в уме информацию только о ней, а не обо всех остальных частях системы.

При этом учитывается правильная декомпозиция, которая означает следующее.

- Количество связей между отдельными подсистемами должно быть минимальным.
- Связность отдельных частей внутри каждой подсистемы должна быть максимальной.
- Каждая подсистема должна инкапсулировать свое содержимое (скрывать его от других подсистем).

Очень важно обратить внимание на подходы в программной инженерии.

• Первый подход называется функционально-модульным, или структурным. В его основу положен принцип функциональной декомпозиции, при которой структура системы описывается в терминах иерархии ее функций и передачи информации между отдельными функциональными элементами.

• Второй, объектно-ориентированный, подход использует объектную декомпозицию. При этом структура системы описывается в терминах объектов и связей между ними, а поведение системы описывается в терминах обмена сообщениями между объектами. Итак, сущ-

ность структурного подхода к разработке ПО ЭИС заключается в ее декомпозиции (разбиении) на автоматизируемые функции.

При разработке учитываются следующие принципы метода структурного подхода.

- Принцип иерархического упорядочения – принцип организации составных частей системы в иерархические древовидные структуры с добавлением новых деталей на каждом уровне.
- Принцип абстрагирования – выделение существенных аспектов системы и отвлечение от несущественных.
- Принцип непротиворечивости, обоснованности и согласованности элементов системы.
- Принцип структурирования данных – данные должны быть структурированы и иерархически организованы.

Существуют различные виды моделей, среди которых наиболее распространенными видами моделей (диаграмм) являются следующие:

- **DFD (Data Flow Diagrams)** – диаграммы потоков данных;
- **SADT (Structured Analysis and Design Technique)** – модели и соответствующие функциональные диаграммы;
- **ERD (Entity – Relationship Diagrams)** – диаграммы «сущность-связь». Конкретный вид перечисленных диаграмм и интерпретация их конструкций зависят от стадии ЖЦ ПО.

Пример концепции SADT

Метод SADT представляет собой совокупность правил и процедур, предназначенных для построения функциональной модели объекта какой-либо предметной области. Функциональная модель SADT отображает функциональную структуру объекта, т.е. производимые им действия и связи между этими действиями. В модели учитываются:

- графическое представление блочного моделирования. Графика блоков и дуг SADT – диаграммы отображает функцию в виде блока;
- строгость и точность. Правила SADT включают: ограничение количества блоков на каждом уровне декомпозиции;

- отделение организации от функции, т.е. исключение влияния административной структуры на модель.

Состав функциональной модели

Результатом применения метода SADT является модель, состоящая из **диаграмм, фрагментов текстов и глоссария, имеющих ссылки друг на друга.**

Диаграммы – главные компоненты модели, все функции организации и интерфейсы на них представлены как блоки и дуги соответственно. **Управляющая информация** входит в блок сверху. **Механизм** (человек или автоматизированная система), который осуществляет операцию, представляется дугой, входящей снизу.

На рис. 3 представлен отражающий функциональный блок и интерфейсные дуги.

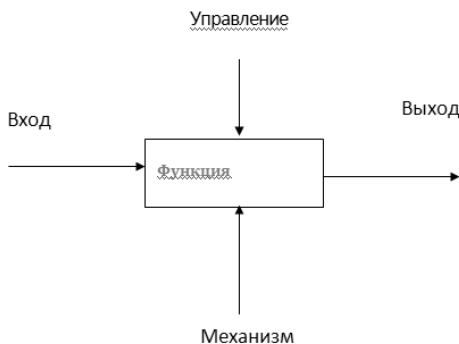


Рис. 3. Функциональный блок и интерфейсные дуги

При построение иерархии диаграмм учитываются следующие правила.

- Построение SADT-модели начинается с представления всей системы в виде простейшего компонента – **одного блока и дуг**, изображающих интерфейсы с функциями вне системы. **Затем блок детализируется** на другой диаграмме с **помощью нескольких блоков**. Эти блоки определяют основные подфункции исходной функции.
- На каждом шаге декомпозиции диаграмма предыдущего уровня называется **родительской** для более детальной диаграммы.

На рис. 4 -10 представлены по шагам действия, связанные с построением SADT, от общего представления до итоговой иерархии.

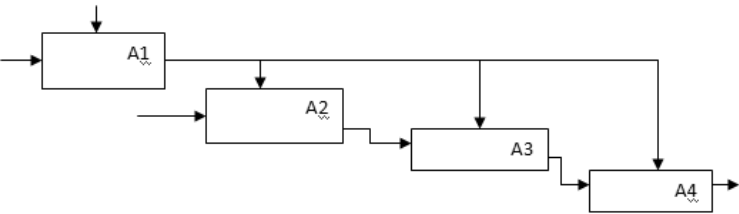


Рис. 4. Общее представление

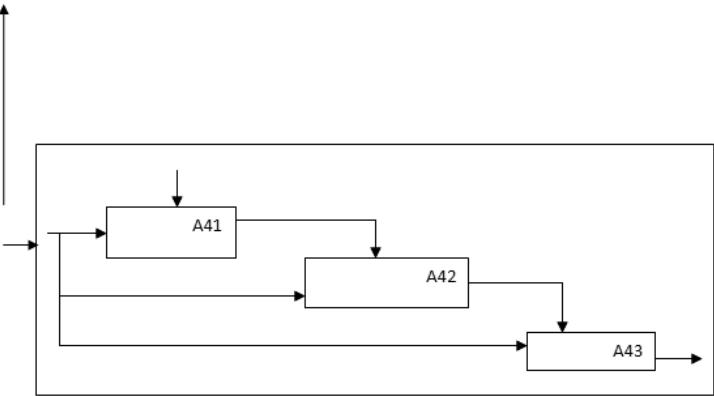


Рис. 5. Варианты выполнения функций и соединения дуг с блоками

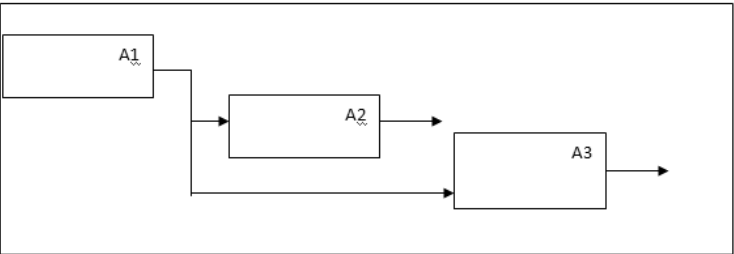


Рис. 6. Верхняя диаграмма является родительской для нижней диаграммы

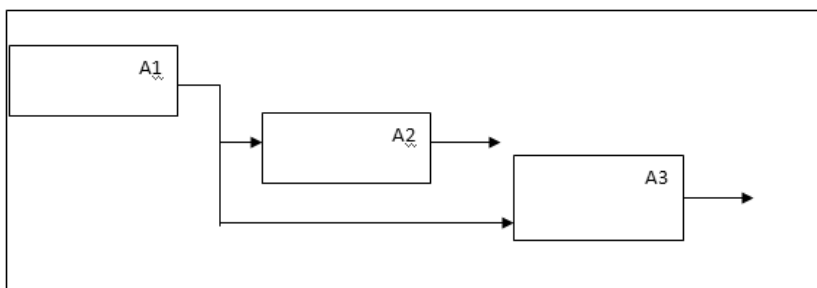
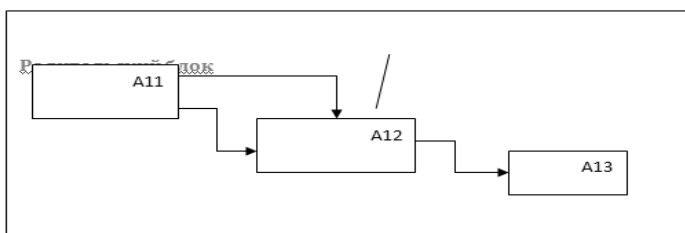
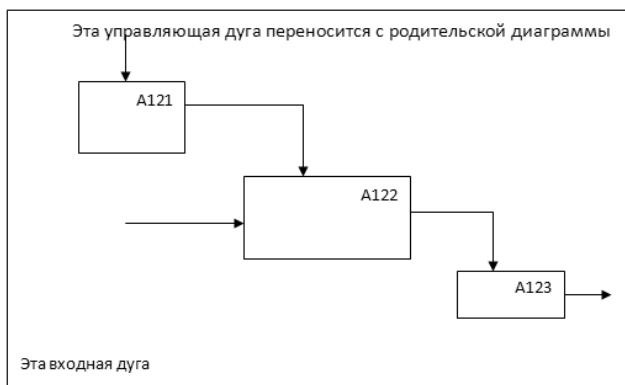


Рис. 7. Функции блоков A2 и A3 могут выполняться параллельно



а



б

Рис. 8. Соответствие интерфейсных дуг родительской (а) и детальной (б) диаграмм



Рис. 9. Пример обратной связи

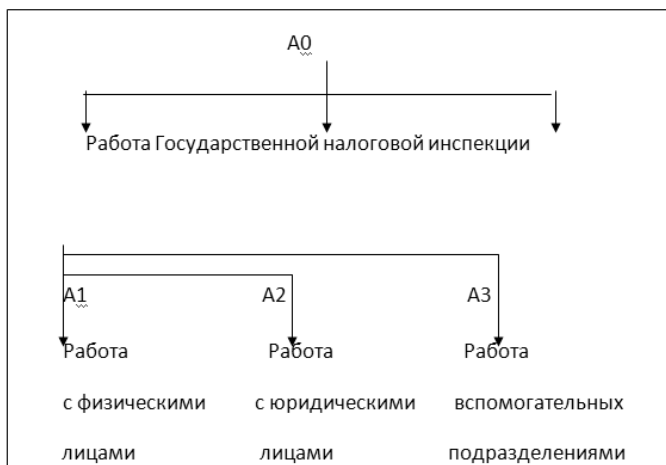


Рис. 10. Пример иерархии диаграмм механизмов

Типы связей между функциями

Различают связи семи типов (в порядке возрастания их относительной значимости):

- случайная;
- логическая;

- временная;
- процедурная;
- коммуникационная;
- последовательная;
- функциональная.

В табл. 1 представлены все типы связей. Важно отметить, что уровни 4–6 устанавливают типы связей, которые разработчики считают важнейшими для получения диаграмм хорошего качества.

Т а б л и ц а 1

Типы связей

Уровень значимости	Тип связи	Характеристика типа связи	
		Для функций	Для данных
0	Случайная	Случайная	Случайная
1	Логическая	Функции одного и того же множества или типа (например, «редактировать все входы»)	Данные одного и того же множества или типа
2	Временная	Функции одного и того же периода времени (например, «операции инициализации»)	Данные, используемые в каком-либо временном интервале
3	Процедурная	Функции, работающие в одной и той же фазе или итерации, например, «первый проход компилятора»	Данные, используемые во время одной и той же фазы или итерации
4	Коммуникационная	Функции, использующие одни и те же данные	Данные, на которые воздействует одна и та же деятельность
5	Последовательная	Функции, выполняющие последовательное преобразование одних и тех же данных	Данные, преобразуемые последовательными функциями
6	Функциональная	Функции, объединяемые для выполнения одной функции	Данные, связанные с одной функцией

Комментарии к табл. 1.

1. Случайная связь – показывает, что конкретная связь между функциями незначительна или полностью отсутствует, это относится к ситуации, когда имена данных на SADT-дугах в одной диаграмме имеют слабую связь друг с другом.

2. Логическая связь – данные и функции собираются вместе благодаря тому, что они попадают в общий класс или набор элементов, но необходимых функциональных отношений между ними не обнаруживается.

3. Временная связь – представляет функции, связанные во времени, когда данные используются одновременно или функции включаются параллельно, а не последовательно.

4. Процедурная связь – функции сгруппированы вместе благодаря тому, что они выполняются в течение одной и той же части цикла или процесса.

5. Коммуникационная связь – функции группируются благодаря тому, что они используют одни и те же исходные данные и/или производят одни и те же выходные данные.

6. Последовательная связь – выход одной функции служит входными данными для следующей функции. Связь между элементами на диаграмме является более тесной, чем в рассмотренных выше случаях, поскольку моделируются причинно-следственные зависимости.

7. Функциональная связь – все элементы функции влияют на выполнение одной и той же функции. Диаграмма, являющаяся чисто функциональной, не содержит чужеродных элементов, относящихся к последовательному или более слабому типу связи. Одним из способов определения функционально связанных диаграмм является рассмотрение двух блоков, связанных через управляющие дуги.

Контрольные вопросы

1. Дайте определение понятия «метод и технология».
2. Назовите принципы метода структурного подхода.
3. Назовите семь типов связей между функциями и дайте краткую характеристику.
4. Дайте определение понятия «случайная связь».

6. КАЧЕСТВО ПРОГРАММНЫХ СРЕДСТВ

В соответствии со стандартами **обеспечение качества** – это «совокупность планируемых и систематически проводимых мероприятий, необходимых для уверенности в том, что продукция или процессы удовлетворяют определенным требованиям потребителей к качеству».

Основные понятия

Система качества включает в себя: совокупность организационной структуры, ответственности, процедур, процессов и ресурсов, обеспечивающих осуществление руководства качеством продукции или процессов. Изучением и реализацией методов и средств количественного оценивания качества продукции занимается научная дисциплина – **квалиметрия**.

В основе квалиметрии лежат три базовых положения:

- практическая необходимость методов количественной оценки характеристик качества продукции для решения задач их планирования и контроля;
- подход к качеству как к единому динамическому сочетанию ряда отдельных свойств;
- наличие принципиальной возможности измерения в количественной форме, как отдельных свойств, так и их сочетаний, в том числе интегрального качества.

Теоретическая квалиметрия абстрагируется от конкретных объектов и изучает общие закономерности и математические модели, связанные с оцениванием качества. Практической задачей квалиметрии является разработка и развитие всех комплексных и дифференциальных методов оценки качества.

Разнообразие областей применения компьютеров становится все шире и их корректная работа часто является определяющей для каче-

ственного управления объектами, успеха предприятий или безопасности человека. Поэтому тщательное специфицирование и оценивание характеристик качества программного продукта – ключевой фактор обеспечения их адекватного применения.

По отношению к программным средствам система обеспечения качества – это совокупность методов и средств организации управляющих и исполнительных подразделений предприятия, участвующих в проектировании, разработке и сопровождении комплексов программ с целью придания им свойств, обеспечивающих удовлетворение потребностей заказчиков и потребителей при минимальном или допустимом расходовании ресурсов.

Требуемое качество при разработке проектов программных систем (ПС), как и любой продукции, можно обеспечить двумя способами:

- путем использования только заключительного контроля и испытаний готовых объектов и исключением из поставки или направлением на доработку продуктов, не соответствующих требуемому качеству;
- посредством применения регламентированных технологий и систем обеспечения качества процессов проектирования и разработки, предотвращающих дефекты и гарантирующих высокое качество продукции во время ее создания и модификации.

Для создания современных прикладных высококачественных информационных систем необходимы оба способа с акцентом на применение регламентированных технологий. Таким образом, обеспечение и удостоверение качества сложных ПС должно базироваться на проверках и испытаниях:

- технологий обеспечения жизненного цикла программных средств, поддержанных регламентированными системами качества;
- готового программного продукта с полным комплектом адекватной эксплуатационной документации.

Для конкретных ПС доминирующие критерии качества выделяются и определяются при проектировании его функциональным назначением и требованиями технического задания.

Программы для вычислительных систем (ВС) как объекты проектирования, разработки, испытаний и оценки качества характеризуются следующими обобщенными показателями:

- проблемно-ориентированной областью применения, техническим и социальным назначением программного комплекса;

- конкретным типом решаемых функциональных задач с достаточно определенной областью применения соответствующими пользователями;
- объемом и сложностью совокупности программ и базы данных, решающей единую целевую задачу данного типа;
- необходимым составом и требуемыми значениями характеристик качества функционирования программ и величиной допустимого риска (ущерба) из-за недостаточного их качества;
- степенью связи решаемых задач с реальным масштабом времени или допустимой длительностью ожидания результатов решения задачи;
- прогнозируемыми значениями длительности эксплуатации и перспективой создания, множества версий комплекса программ;
- предполагаемым тиражом производства и применения комплекса программ;
- степенью необходимой документированности программ.

Качество в использовании – это основное качество системы, содержащей ПС, которое воспринимается пользователями. Оно измеряется, скорее, в терминах результата функционирования и применения программ, чем внутренних свойств самого ПС. Цель такого оценивания – определение, имеет ли продукт требуемый эффект в специфическом контексте использования.

Качество изменяется в течение жизненного цикла ПС, т.е. его требуемое и реальное значение в начале ЖЦ почти всегда отличается от фактически достигнутого при завершении проекта и качества поставляемой пользователям версии продукта. На практике важно оценивать качество программ не только в завершенном виде, но и в процессе их проектирования, разработки и сопровождения.

При системном анализе и проектировании программных средств необходимо определять и учитывать связи, влияние и взаимодействие следующих основных факторов, которые отражаются на их качестве:

- назначение, содержание и описание функциональных характеристик, субхарактеристик и атрибутов, определяющих специфические особенности целей, задач, свойств и сферы применения конкретного программного средства – его функциональную пригодность;
- конструктивные характеристики качества, способствующие улучшению и совершенствованию назначения, функций и возможностей применения ПС;

- метрики, меры и шкалы выбранных и пригодных для измерения и оценивания конкретных характеристик и атрибутов качества ПС с учетом определенной достоверности;
- уровни возможной детализации при описании и оценивании определенных характеристик и атрибутов качества ПС;
- цели и особенности потребителей результатов оценивания характеристик качества ПС;
- внешние и внутренние негативные факторы, влияющие на достигаемое качество создания и применения ПС;
- доступные ресурсы, ограничивающие возможные величины реальных характеристик качества ПС;
- конкурентоспособность, выраженная отношением эффективности применения к стоимости приобретения и эксплуатации ПС.

Множество характеристик качества программных средств можно разделить на две принципиально различающихся группы.

- **Функциональные характеристики** (функциональность) – определяющие назначение, свойства и задачи, решаемые комплексом программ для основных пользователей, отличающиеся очень широким спектром и разнообразием, состав и специфику которых трудно унифицировать и можно категоризировать только по большому количеству классов и свойств ПС.
- **Конструктивные характеристики** качества, номенклатура которых может быть унифицирована, адаптирована и использована для описания остальных – внутренних и внешних – стандартизируемых характеристик качества, поддерживающих и улучшающих реализацию основных функциональных требований к качеству объектов и процессов ЖЦ программных средств.

Определение и сравнение **функционального качества** программ целесообразно рассматривать в пределах ограниченных классов ПС, выполняющих подобные функции. Функциональная пригодность (см. стандарт ISO-9126) непосредственно определяет основное назначение и функции ПС для пользователей. В контракте и техническом задании для каждого проекта она должна быть выделена и формализована для однозначного понимания.

Группа **конструктивных характеристик** играет подчиненную роль и должна в первую очередь поддерживать и обеспечивать высокое качество реализации функций ПС и его применения по основному назначению.

Для каждого конкретного проекта ПС из них может быть выделена представительная группа наиболее важных и оказывающих наибольшее влияние на решение определенных функциональных задач.

Таким образом, при экономическом анализе проектов ПС возможны два сценария:

1) создание и весь жизненный цикл комплекса программ и/или базы данных ориентируется разработчиком на **массовое тиражирование** и распространение на рынке для заранее неизвестных покупателей;

2) разработка проекта ПС и/или БД предполагается поставщиком-разработчиком **для конкретного потребителя-заказчика**.

Оценка рентабельности затрат: потенциальные покупатели-пользователи перед приобретением ПС обычно оценивают конкурентоспособность новой продукции на рынке **по величине отношения** возможной экономической **эффективности** (ценности) применения ПС и способности удовлетворения пользователями своих потребностей с необходимым качеством при его использовании **к стоимости** (цене), которую требуется заплатить пользователям при приобретении и эксплуатации данного комплекса программ или базы данных.

Отбраковка вариантов реализации ПС ведется по показателю эффективность/стоимость для пользователей с учетом прогнозов конкурентоспособности и возможностей распространения на рынке.

Второй сценарий предполагает наличие **определенного заказчика-потребителя** проекта ПС, который определяет основные технические и экономические требования и характеристики. Однако для заказчика и разработчика перед заключением контракта необходим достаточно достоверный системный анализ путем прогнозирования технико-экономического обоснования (ТЭО) требуемых ресурсов по трудоемкости, стоимости, срокам и другим характеристикам.

Жизненный цикл ПС можно разделить на **две части**, существенно различающиеся экономическими особенностями процессов, характеристиками и влияющими на них факторами. **В первой части ЖЦ** производятся системный анализ, проектирование, разработка, тестирование и испытания базовой версии ПС.

Исследования последних десятилетий показали, что при разработке крупномасштабных комплексов программ **реальная средняя производительность** разработчиков почти на два порядка ниже таких оценок (тысяча строк в неделю) и измеряется только несколькими десятками строк в неделю. При этом средняя стоимость разработки одной

строки полностью новой программы высокого качества составляет свыше 10 \$ и достигает 100 \$.

При таком анализе, естественно, должен учитываться весь цикл разработки, начиная от **подготовки технического задания до завершения успешных квалификационных испытаний**, а также весь коллектив участников проекта, включая **руководителей, системных аналитиков и вспомогательный персонал**. Непосредственным «написанием» текстов программ в коллективе занимается только треть или четверть разработчиков и почти столько же их автономным и квалификационным тестированием.

В табл. 2 представлен вариант относительного распределения затрат ресурсов по этапам работ для сложных встроенных ПС реального времени, который можно использовать как базу для приближенных оценок.

Т а б л и ц а 2

Вариант относительного распределения затрат ресурсов по этапам работ для сложных встроенных ПС реального времени

Этапы жизненного цикла	Относительная трудоемкость этапа работ, %	Относительная длительность этапа работ, %	Относительное число специалистов на этапе, %
1. Анализ требований к программам и планирование	8	22	25
2. Архитектурное проектирование программного средства	16	22	40
3. Детальное проектирование программного средства	26	18	60
4. Кодирование и тестирование программных компонентов	28	18	100
5. Интеграция, квалификационное тестирование и испытания программного средства	22	20	70

Вторая часть ЖЦ, отражающая эксплуатацию, сопровождение, модификацию и перенос ПС на иные платформы, в меньшей степени связана с функциональными характеристиками объекта и среды разработки.

Пример стандарта, регламентирующего качество программных средств

Стандарт ISO 9126:1991 – используется при оценке программного продукта.

Проект стандарта ISO 9126 состоит из нескольких частей под общим заголовком. Информационная технология. Качество программных средств:

Часть 1. Модель качества.

Часть 2. Внешние метрики качества.

Часть 3. Внутренние метрики качества.

Часть 4. Метрики качества в использовании.

Ниже представлена информация по отдельным частям.

Модель качества

Модель характеристик качества ПС состоит из шести групп базовых показателей, каждая из которых детализирована несколькими нормативными субхарактеристиками:

- функциональная пригодность;
- надежность;
- эффективность;
- применимость и переносимость;
- сопровождаемость.

Дополнительно каждая характеристика сопровождается субхарактеристикой **согласованность**, которая должна отражать отсутствие противоречий с иными стандартами и нормативными документами, а также с другими показателями в данном стандарте.

Функциональная пригодность определяется:

- пригодностью для применения;
- корректностью (правильностью, точностью);
- способностью к взаимодействию;
- защищенностью.

Надежность характеризуется:

- уровнем завершенности (отсутствие ошибок);
- устойчивостью к дефектам;
- восстанавливаемостью;
- доступностью-готовностью.

Эффективность рекомендуется отражать временной эффективностью и используемостью ресурсов.

Применимость (практичность) отражена:

- понятностью;
- простотой использования;
- изучаемостью;
- привлекательностью.

Сопровождаемость представляется:

- удобством для анализа;
- изменяемостью;
- стабильностью;
- тестируемостью.

Переносимость (мобильность) предлагается отражать:

- адаптируемостью;
- простотой установки – инсталляцией;
- сосуществованием – соответствием;
- замещаемостью.

Описания показателей качества ориентированы на высококвалифицированных системных аналитиков и заказчиков ПС, которым предоставляется возможность выбирать необходимую номенклатуру и способ оценивая характеристик в соответствии с назначением, областью применения и конкретными особенностями создаваемых объектов.

Общее представление стандарта ISO 9126 рекомендуется отражать тремя взаимодействующими и взаимозависимыми метриками характеристик качества.

Внутренние метрики в соответствии со стандартами могут применяться в ходе проектирования и программирования к неисполняемым компонентам ПС таким, как спецификация или исходный про-

граммный текст. Основная цель применения внутренних метрик – обеспечение требуемого внешнего качества.

Измерения внутренних метрик используют: **свойства, категории, числа или характеристики элементов из состава ПС**, которые, например, **имеются в процедурах исходного программного текста, в графе потока управления, в потоке данных и в описаниях изменения состояний памяти**. Качество документации также может оцениваться с использованием внутренних метрик.

Внешние метрики используют меры ПС, выведенные из поведения системы, частью которых они являются, путем испытаний, эксплуатации и наблюдения исполняемых программ или функционирования информационной системы.

Внешние метрики обеспечивают заказчикам, пользователям и разработчикам возможность прослеживать и анализировать качество ПС в ходе испытаний или опытной эксплуатации.

Метрики качества в использовании отражают, в какой степени продукт удовлетворяет потребности конкретных пользователей в достижении заданных целей. Эта метрика не отражена в числе шести базовых характеристик ПС, регламентируемых стандартом ISO 9126-1 вследствие ее общности. Она рекомендуется для интегральной оценки результатов функционирования и применения комплексов программ в стандарте ISO 9126-4.

Качество в использовании. Связь качества в использовании с другими характеристиками ПС зависит **от задач и функций их потребителей:**

- для заказчика требуется полное соответствие характеристик программного продукта условиям контракта, технического задания и спецификациям требований;
- для конечного оперативного пользователя ПС по основному назначению качество в использовании обуславливают в основном характеристики функциональных возможностей, надежности, практичности и эффективности;
- для персонала сопровождения ПС качество в использовании определяется преимущественно сопровождаемостью;
- для персонала, выполняющего перенос ПС на иные платформы, а также установку и адаптацию к среде применения, качество в использовании определяется прежде всего мобильностью.

Характеристики, субхарактеристики и атрибуты качества ПС следующие:

категорийные-описательные, отражающие набор свойств и общие характеристики объекта – его функции, категории ответственности, защищенности и важности, которые могут быть представлены номинальной шкалой категорий-свойств;

количественные, представляемые множеством упорядоченных числовых точек, отражающих непрерывные или дискретные закономерности и описываемые интервальной или относительной шкалой, которые можно объективно измерить и численно сопоставить с требованиями;

качественные – содержащие несколько упорядоченных или отдельных свойств-категорий, которые характеризуются порядковой или точечной шкалой набора категорий (есть – нет, хорошо – плохо), устанавливаются, выбираются и оцениваются в значительной степени субъективно и экспертно.

Существуют следующие уровни показателей качества.

К первому уровню относятся показатели качества, которые характеризуются наибольшим разнообразием значений - свойств программ и наборов данных и охватывают весь спектр классов, назначений и функций современных ПС.

Ко второму уровню показателей качества относятся достаточно достоверно и объективно измеряемые численные характеристики ПС.

Третий уровень – их атрибуты имеют описательный, качественный вид.

Понятие **практичность** тесно связана с функциональной пригодностью. Обобщенно этот показатель можно отразить трудоемкостью и длительностью, которые необходимы для изучения и полного освоения функций и технологии применения соответствующего ПС.

Другая характеристика – сопровождаемость может иметь ограниченный характер полной замены программ на вновь разработанные версии и тем самым сливаться с процессами разработки или осуществляться как непрерывная поддержка множества пользователей консультациями, адаптациями и корректировками программ.

Введение строгих количественных метрик в программирование должно было способствовать решению ряда практических задач:

- **предсказывать вероятное число ошибок** в системе с самого начала проектирования на основе анализа фазы проектирования системы и сложности последующего сопровождения;
- на основе анализа исходного кода программ **прогнозировать уровень сложности процессов тестирования и процент остающихся ошибок**; по оценкам сложности фазы проектирования системы определять конечный размер кода;
- **определять корреляцию** отдельных характеристик программного кода с качеством готовой системы;
- **контролировать стадии развития проекта**;
- **анализировать явные и скрытые дефекты**;
- на основе экспериментального сравнения **выявлять лучшие методы и технологии**.

Оценивание качества готового программного продукта

На отдельных этапах жизненного цикла (ЖЦ) программные системы и ПС имеют законченный характер и могут быть квалифицированы как конечный продукт этих этапов или продукт, пригодный для поставки заказчику. Таким продуктом могут быть программные модули, версии функциональных компонентов и комплексов программ, их проекты и прототипы с высоким уровнем конкретизации и документирования.

Процессы анализа, оценки и удостоверения качества промежуточных и конечных продуктов подробно рассмотрены в стандарте **ISO 14598-1-6:1998- 2000. Оценивание программного продукта**. Данный стандарт состоит из шести частей.

- Ч. 1. Общий обзор.
- Ч. 2. Планирование и управление.
- Ч. 3. Процессы для разработчиков.
- Ч. 4. Процессы для покупателей.
- Ч. 5. Процессы для оценщиков.
- Ч. 6. Документация по модулям оценки.

В первой части стандарта изложена концепция и методология применения частей стандарта. Конкретизирована связь положений стандарта со стандартами ISO/IEC 12207:2010 и ISO/IEC 9126-1-4 (в том числе с проектами частей последнего). Рекомендуется следующая общая схема процессов оценивания качества программ:

1) установка исходных требований для оценивания – определение целей; идентификация типа ПС; выделение адекватных показателей качества;

2) селекция метрик качества; установление уровней приоритета метрик; определение критериев для экспертизы;

3) планирование и проектирование процессов оценивания качества продукта;

4) выполнение измерений для оценивания; сравнение с критериями; обобщение и оценка результатов.

В данном стандарте, как и в ISO/IEC 9126-1-4, выделяются характеристики качества для пользователей – внешние и внутренние.

Во второй части стандарта изложена концепция планирования и управления процессами оценивания, а также связь с процессом управления проектом системы или ПС. Даны рекомендации по организации, технологии, инструментальному оснащению и проведению испытаний качества ПС или системы. Дана схема плана количественной оценки качества ПС.

Третья, четвертая и пятая части стандарта имеют аналогичную структуру, но ориентированы конкретно на разработчиков, пользователей и оценщиков качества соответственно. Даны рекомендации по реализации процессов оценивания ПС, в том числе:

- концепция оценивания;
- специфика определения требований к процессам оценивания;
- идентификация показателей качества для конкретных пользователей результатов испытаний.

Рекомендуемая технология выполнения оценивания программ содержит следующие этапы:

1) специфицирование внешних и внутренних характеристик качества;

2) планирование и проектирование их оценивания;

3) реализация процессов испытаний;

4) анализ результатов и оформление выводов о качестве ПП.

Реализация процессов оценивания должна быть автономной и независимой, но связанной с работами и задачами жизненного цикла

конкретного проекта ПС в соответствии с адаптированной версией ISO/IEC 12207:2010.

В **шестой части стандарта** приводится концепция документирования модулей оценки. Модуль оценки представляет собой полностью укомплектованную информацию, необходимую для проведения процесса оценки некоторой характеристики или подхарактеристики качества, и включает в себя описание метрик, методик сбора и оценки данных, форматы отчетов по результатам оценки [10–14].

Таким образом, всю информацию по оценке качества ПС рекомендуется представлять в виде набора модулей оценки.

Характеристики качества баз данных

Для базы данных (БД) необходимо особенно высокое качество и его квалифицированное системное проектирование.

Базу данных можно рассматривать как два компонента: систему программ управления данными и совокупность данных, упорядоченных по некоторым правилам. Поэтому при анализе качества базу данных целесообразно делить на два компонента:

- программные средства системы управления базой данных (СУБД), независимые от сферы их применения, структуры и смыслового содержания накапливаемых и обрабатываемых данных;
- информацию базы данных (ИБД), доступную для накопления, упорядочения, обработки и использования в конкретной проблемно-ориентированной сфере применения.

Одна и та же система управления базой данных (СУБД) может обрабатывать различные по структуре, составу и содержанию данные, а одни и те же данные могут управляться программными средствами различных СУБД. Они могут рассматриваться в их жизненном цикле (ЖЦ) как два объекта, которые различаются:

- номенклатурой и содержанием показателей качества, определяющих их назначение, функции и потребительские свойства;
- технологией и средствами автоматизации разработки и обеспечения всего ЖЦ каждого объекта;
- категориями специалистов, обеспечивающих: создание, эксплуатацию или применение компонентов БД;
- комплектами эксплуатационной и технологической документации, поддерживающими жизненный цикл объектов.

Важнейшими характеристиками качества СУБД являются требования функциональной пригодности для процессов формирования и изменения информационного наполнения БД администраторами, а также доступа к данным и представления результатов пользователям БД.

В зависимости от конкретной проблемно-ориентированной области применения СУБД приоритет при системном анализе требований к качеству может быть отдан различным конструктивным характеристикам: **либо надежности и защищенности применения** (финансовая сфера), **либо удобству использования малоквалифицированными пользователями** (социальная сфера), **либо эффективности использования ресурсов** (сфера материально-технического снабжения).

Вторым компонентом БД является собственно накапливаемая и обрабатываемая информация. Так же как для ПС, характеристики качества ИБД можно разделить на функциональные и конструктивные. Их номенклатура, содержание и субхарактеристики базируются на описаниях, рекомендуемых стандартом ISO 9126. Они представляются достаточно универсальными и применимыми для систематизации характеристик качества информации баз данных.

Так же как для ПС, для БД в составе функциональной пригодности целесообразно использовать группу субхарактеристик, определяющих функциональные и структурные требования.

Дополнительно функциональная пригодность многих ИБД может отражаться:

- полнотой накопленных описаний объектов – относительным числом объектов или документов, имеющихся в БД, к общему числу объектов по данной тематике или по отношению к числу объектов в аналогичных БД того же назначения;
- идентичностью данных – относительным числом описаний объектов, не содержащих дефекты и ошибки, к общему числу документов об объектах в ИБД;
- актуальностью данных – относительным числом устаревших данных об объектах в ИБД к общему числу накопленных и обрабатываемых данных.

Требования к информации баз данных также должны содержать особенности **обеспечения ее надежности, эффективности использования ресурсов ЭВМ, практичности, применимости, сопровождаемости, мобильности.**

Корректность или достоверность данных – это степень соответствия информации об объектах в БД реальным объектам вне ЭВМ в данный момент времени.

К корректности БД можно отнести некоторые объемно-временные характеристики сохраняемых и обрабатываемых данных:

- **объем базы данных** – относительное число записей описаний объектов или документов в базе данных, доступных для хранения и обработки, по сравнению с полным числом реальных объектов во внешней среде;

- **оперативность** – степень соответствия динамики изменения описаний данных в процессе сбора и обработки состояниям реальных объектов или величина допустимого запаздывания между появлением или изменением характеристик реального объекта относительно его отражения в базе данных;

- **глубина ретроспективы** – максимальный интервал времени от даты выпуска и/или записи в базу данных самого раннего документа до настоящего времени;

- **динамичность** – относительное число изменяемых описаний объектов к общему числу записей в БД за некоторый интервал времени, определяемый периодичностью издания версий БД.

Защищенность информации БД реализуется в основном программными средствами СУБД в сочетании с поддерживающими их средствами организации и защиты данных. Косвенным показателем ее качества может служить относительная доля вычислительных ресурсов, используемых непосредственно средствами защиты информации БД.

Надежность информации баз данных может основываться на применении при системном проектировании понятий и методов теории надежности, позволяющей получить ряд четких измеряемых интегральных показателей их качества.

Надежность должна повышаться за счет средств обеспечения помехоустойчивости, оперативного контроля и восстановления ИБД.

Устойчивость к дефектам и ошибкам – свойство ИБД автоматически поддерживать заданный уровень качества данных в случаях проявления дефектов и ошибок. Для этого в ИБД рекомендуется вводить оперативное обнаружение дефектов и ошибок информации.

Восстанавливаемость – свойство ИБД в случае отказа возобновлять требуемый уровень качества информации, а также корректировать поврежденные данные.

Для этого необходимы вычислительные ресурсы и время на выявление неработоспособного состояния, диагностику причин отказа.

Доступность или готовность – свойство ИБД быть в состоянии выполнять требуемую функцию в данный момент времени при заданных условиях использования информации базы данных.

Эффективность использования ресурсов ЭВМ при системном анализе реального функционирования БД отражается временными характеристиками взаимодействия конечных пользователей и администраторов ИБД в процессе эксплуатации базы данных по прямому назначению.

Модели оценки характеристик качества программного обеспечения

Далее рассматриваются варианты моделей оценки характеристик качества ПО.

Размерно-ориентированные метрики

Размерно-ориентированные метрики прямо измеряют программный продукт и процесс его разработки. Основываются размерно-ориентированные метрики на LOC-оценках (Lines Of Code). LOC-оценка – это количество строк в программном продукте.

Например, для проектов A01, B02, C03 исходные данные для расчета этих метрик сведены в табл. 3.

Т а б л и ц а 3

Исходные данные

Проект	Затраты, чел.- мес.	Стои- мость тыс. \$	KLOC, тыс. LOC	Количе- ство страниц	Количе- ство ошибок	Количе- ство человек
A01	24	168	12,1	365	29	3
B02	62	440	27,2	1224	86	5
C03	43	314	20,2	1050	64	6

Запись о проекте A01 показывает: 12 100 строк программы были разработаны за 24 чел.-мес. и стоили \$168 000. Кроме того, по проекту было разработано 365 страниц документации, а в течение первого года эксплуатации было зарегистрировано 29 ошибок. Разрабатывали проект три человека.

На основе таблицы вычисляются размерно-ориентированные метрики производительности и качества проекта:

производительность = Длина / Затраты (тыс.LOC/чел.-мес.);

качество = Ошибки / Длина (Единиц/тыс. LOC);

удельная стоимость = Стоимость /Длина (тыс.\$/LOC);

документированность = Страниц_Документа / Длина (Страниц/тыс.LOC).

Недостатки размерно-ориентированных метрик:

- зависимы от языка программирования;
- требуют исходных данных, которые трудно получить на начальной стадии проекта;
- не приспособлены к непроцедурным языкам программирования.

Функционально-ориентированные метрики

Функционально-ориентированные метрики косвенно измеряют программный продукт и процесс его разработки. Вместо подсчета LOC-оценки при этом рассматривается не размер, а функциональность или полезность продукта.

Для функционально-ориентированных метрик используется несколько информационных характеристик.

Количество внутренних логических файлов – логические группы данных, которые могут быть частью базы данных или отдельным файлом.

Выводы, вводы, запросы относятся к категории **транзакция**. **Транзакция** – это элементарный процесс, различаемый пользователем и перемещающий данные между внешней средой и программным приложением.

Каждой из выявленных характеристик ставится в соответствие сложность. Для этого характеристике назначается низкий, средний или высокий ранг, а затем формируется числовая оценка ранга.

Для **транзакций ранжирование основано на количестве ссылок и количестве типов элементов данных**. Для **файлов ранжирование**

основано на количестве типов-элементов записей и типов-элементов данных, входящих в файл.

Примеры элементов данных для различных характеристик приведены в табл. 4 и содержат правила учета элементов из графического интерфейса пользователя (ГИП).

Т а б л и ц а 4

Примеры элементов данных

Информационная характеристика	Элементы данных
Внешние вводы	Поля ввода данных, сообщения об ошибках, вычисляемые значения, кнопки
Внешние выводы	Поля данных в отчетах, вычисляемые значения, сообщения об ошибках, заголовки столбцов, которые читаются из внутреннего файла
Внешние запросы	Вводимые элементы: поле, используемое для списка, щелчок мыши Выводимые элементы: отображаемые на экране поля

Данные для определения ранга и оценки сложности транзакций и файлов приведены в табл. 5–8. Числовая оценка указана в круглых скобках. Использовать их очень просто. Например, внешнему вводу, который ссылается на два файла и имеет 7 элементов данных, назначается средний ранг и оценка сложности 4.

Т а б л и ц а 5

Ранг и оценка сложности внешних вводов

Ссылки на файлы	Элементы данных		
	1-4	5-15	>15
0-1	Низкий (3)	Низкий (3)	Средний (4)
2	Низкий (3)	Средний (4)	Высокий (6)
>2	Средний (4)	Высокий (6)	Высокий (6)

Т а б л и ц а 6

Ранг и оценка сложности внешних выводов

Ссылки на файлы	Элементы данных		
	1-4	5-19	>19
0-1	Низкий (4)	Низкий (4)	Средний (5)
2-3	Низкий (4)	Средний (5)	Высокий (7)
>3	Средний (5)	Высокий (7)	Высокий (7)

Т а б л и ц а 7

Ранг и оценка сложности внешних запросов

Ссылки на файлы	Элементы данных		
	1-4	5-19	>19
0-1	Низкий (3)	Низкий (3)	Средний (4)
2-3	Низкий (3)	Средний (4)	Высокий (6)
>3	Средний (4)	Высокий (6)	Высокий (6)

Т а б л и ц а 8

Ранг и оценка сложности внутренних логических файлов

Ссылки на файлы	Элементы данных		
	1-19	20-50	>50
0-1	Низкий (7)	Низкий (7)	Средний (10)
2-5	Низкий (7)	Средний (10)	Высокий (15)
>5	Средний (10)	Высокий (15)	Высокий (15)

При расчете количества функциональных указателей FP (Function Points) применяются данные из табл. 9.

Порядок действий следующий.

1. В таблицу заносится количественное значение характеристики каждого вида (по всем уровням сложности).

2. Количественные значения характеристик умножаются на числовые оценки сложности.

3. Полученные в каждой строке значения суммируются, давая полное значение для данной характеристики. Эти полные значения суммируются по вертикали и формируют общее количество.

4. После сбора всей необходимой информации приступают к расчетам метрики – количества функциональных указателей FP (Function Points).

Т а б л и ц а 9

Исходные данные для расчета FP-метрик

Имя характеристики	Ранг, сложность, количество			
	Низкий	Средний	Высокий	Итого
Внешние вводы	*3=_	*4=_	*6=_	=
Внешние выводы	*4=_	*5=_	*7=_	=
Внешние запросы	*3=_	*4=_	*6=_	=
Внутренние логические файлы	*7=_	*10=_	*15=_	=
Внешние интерфейсные файлы	*5=_	*7=_	*10=_	=
<i>Общее количество</i>				=

Количество функциональных указателей вычисляется по следующей формуле:

$$FP = \text{Общее количество} \times (0,65 + 0,01 \times \sum Fi),$$

где Fi – коэффициент регулировки сложности ($i = 1 \dots 14$).

Каждый коэффициент может принимать значения: 0 – нет влияния; 1 – случайное; 2 – небольшое; 3 – среднее; 4 – важное; 5 – основное. Значения выбираются эмпирически в результате ответа на 14 вопросов, которые характеризуют системные параметры приложения (табл. 10).

После вычисления FP на его основе формируются метрики производительности, качества и другие оценки:

производительность = ФункцияУказатель / Затраты (FP/чел.-мес.);

качество = Ошибки / ФункцияУказатель (Единиц/FP);

удельная Стоимость = Стоимость / ФункцияУказатель (Тыс.\$/FP);

документированность = СтраницДокумента / ФункцияУказатель (Страниц/FP).

Существуют показатели для определения системных параметров приложения (табл. 10).

Таблица 10

Определение системных параметров приложения

№ п/п	Системный параметр	Описание
1	Передачи данных	Сколько средств данных требуется для передачи или обмена информацией с приложением или системой?
2	Распределенная обработка данных	Как обрабатываются распределенные данные и функции обработки?
3	Производительность	Нуждается ли пользователь в фиксации времени ответа или производительности?
4	Распространенность используемой конфигурации	Насколько распространена текущая аппаратная платформа, на которой будет выполняться приложение?

Окончание табл. 10

№ п/п	Системный параметр	Описание
5	Скорость транзакций	Как часто выполняются транзакции (каждый день, каждую неделю, каждый месяц)?
6	Оперативный ввод данных	Какой процент информации нужно вводить в режиме онлайн?
7	Эффективность работы конечного пользователя	Приложение проектировалось для обеспечения эффективной работы конечного пользователя?
8	Оперативное обновление	Как много внутренних файлов обновляется в онлайн-транзакции?
9	Сложность обработки	Выполняет ли приложение интенсивную логическую или математическую обработку?
10	Повторная используемость	Приложение разрабатывалось для удовлетворения требований одного или многих пользователей?
11	Легкость инсталляции	Насколько трудны преобразования и инсталляция приложения?
12	Легкость эксплуатации	Насколько эффективны и/или автоматизированы процедуры запуска, резервирования и восстановления?
13	Разнообразные условия размещения	Была ли спроектирована, разработана и поддержана возможность инсталляции приложения в разных местах для различных организаций?
14	Простота изменений	Была ли спроектирована, разработана и поддержана в приложении простота изменения?

Основной областью применения функциональных указателей являются информационные системы.

Для продуктов с высокой алгоритмической сложностью используются **метрики свойств (Features Points)**. Они применимы к системному и инженерному ПО, ПО реального времени и встроенному ПО.

Для вычисления указателя свойств добавляется одна характеристика – **количество алгоритмов**. Алгоритм здесь определяется как ограниченная программа вычислений, которая включается в общую компьютерную программу.

Примеры алгоритмов – **обработка прерываний, инвертирование матрицы, расшифровка битовой строки**.

В табл. 11 приведены данные для расчета указателя свойств.

Таблица 11

Исходные данные для расчета указателя свойств

№ п/п	Характеристика	Количество	Сложность	Итого
1	Вводы		*4	=
2	Выводы		*5	=
3	Запросы		*4	=
4	Логические файлы		*7	=
5	Интерфейсные файлы		*7	=
6	Количество алгоритмов		*3	=
<i>Общее количество</i>				=

После заполнения табл. 11 по формуле, определяющей количество функциональных указателей, вычисляется значение указателя свойств. Для сложных систем реального времени это значение на 25...30 %

больше значения, вычисляемого по табл. 9 для количества функциональных указателей.

Недостаток функционально-ориентированных метрик: результаты основаны на субъективных данных, используются не прямые, а косвенные измерения.

FP-оценки легко пересчитать в LOC-оценки (табл. 12). Как показано в таблице, результаты пересчета зависят от языка программирования, используемого для реализации программного обеспечения (ПО).

Т а б л и ц а 12

Пересчет FP-оценок в LOC-оценки

Язык программирования	Количество операторов на 1 FP
Ассемблер	320
C	128
Паскаль	90
C++	64
Java	53
Visual Basic	32
Visual C++	34
Delphi Pascal	29
HTML 3	15
LISP	64
Prolog	64

Контрольные вопросы

1. Что такое обеспечение качества?
2. Чем характеризуется надёжность?
3. Назовите виды метрик качества.
4. Что означает понятие «доступность»?

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Арлоу Д., Нейштадт И.* UML 2 и Унифицированный процесс. Практический объектно-ориентированный анализ и проектирование. – 2-е изд.: пер. с англ. – СПб.: Символ Плюс, 2007. – 624 с., ил.
2. *Мартин Фаулер.* UML. Основы. Краткое руководство по стандартному языку объектного моделирования. – 3-е изд. – СПб.: Изд-во «Символ-Плюс» (ISBN 5-93286-060-X), 2004.
3. Официальный сайт «Основы программной инженерии» (по SWEBOK) [Электронный ресурс] / Сергей Орлик; – Электрон. дан. – Режим доступа: <http://swebok.sorlik.ru> свободный.
4. Профессиональные стандарты в области ИТ [Электронный ресурс] / Ассоциация предприятий компьютерных и информационных технологий. – Электрон. дан. – Режим доступа: <http://www.apkit.ru/committees/education/meetings/standarts.php> свободный.
5. *Кузнецов С.* Открытые системы, процессы стандартизации и профили стандартов ИТ [Электронный ресурс] / CIT-forum. http://citforum.ru/database/articles/art_19.shtml свободный.
6. Техническое регулирование: учебник / под ред. В.Г. Версана, Г.И. Элькина. – М.: ЗАО «Издательство “Экономика”», 2008 – 608 с. ISBN 978-5-282-02800-3
7. IEEE Std 1517-1999. Информационная технология. Процессы жизненного цикла программных средств. Процессы повторного применения (IEEE Std 1517-1999) (IEEE Standard for Information Technology-Software Life Cycle Processes-Reuse Processes).
8. *Липаев В.В.* Программная инженерия. Методологические основы [Текст]: учебник / В.В. Липаев // Гос. ун-т – Высшая школа экономики. – М.: ТЕИС, 2006. – 608 с. ISBN 5-7598-0424-3.
9. ИСО/МЭК ТО 9126-4:2004. Программная инженерия. Качество программного продукта. Часть 4: Показатели качества при использовании (ISO/IEC TR 9126-4:2004) (Software Engineering – Product quality – Part 4: Quality in use metrics).
10. ИСО/МЭК 15288:2007. Системная инженерия. Процессы жизненного цикла систем (ISO/IEC 15288:2007) (Systems Engineering – System life cycle processes).

11. ИСО/МЭК 15504 (все части). Информационная технология. Оценка процессов (ISO/IEC 15504 (all parts)) (Information Technology – Process Assessment).

12. ИСО/МЭК 25030:2010. Программная инженерия. Требования и оценка качества программных продуктов. Требования к качеству (ISO/IEC 25030:2010) (Software Engineering – Software product Quality Requirements and Evaluation (SQuaRE) – Quality Requirements).

13. IEEE/EIA 12207.0-1996. Промышленная реализация МС ИСО/МЭК 12207:1995 Стандарт ИТ. Процессы жизненного цикла программных средств (IEEE/EIA 12207.0-1996) (Industry Implementation of International Standard ISO/IEC 12207:1995 Standard for Information Technology – Software Life Cycle Processes).

14. *Коришкова Л.А.* Стандартизация в области ИТ: учеб.-метод. пособие [Электронный ресурс]. – Режим доступа: <https://elibrary.nstu.ru/source?id=46569>.

ОГЛАВЛЕНИЕ

Предисловие	3
1. Введение в предметную область	4
Классификация стандартов	4
Требования к стандартам	6
Виды нормативных документов	6
Приоритетные для РФ направления стандартизации информаци- онных технологий	8
Контрольные вопросы	9
2. Понятие открытых систем	10
Концепция открытых систем	11
Общие свойства открытых информационных систем	12
Особенности состояния и развития стандартизации	13
Контрольные вопросы	16
3. Методологический базис открытых систем	17
Состав многоуровневой модели пространства спецификаций	17
Классификация базовых спецификаций	20
Контрольные вопросы	23
4. Понятие профиля открытой системы	24
Классификация профилей	25
Основные типы интерфейсов OSE	27
Область функций операционной системы	27
Человекомашинные интерфейсы. Функции	28
Программная инженерия. Функциональная область поддержки разработки программного обеспечения	29
Функциональная область управления данными	29
Область графических функций	30

Функциональная область сетевой поддержки	30
Интегрально поддерживаемые функциональные области	31
Общие свойства профилей	32
Функциональный профиль	34
Профиль защиты информации в распределенных системах	35
Профиль инструментальных средств	35
Контрольные вопросы	36
5. Методы и технологии проектирования программного обеспечения	37
Основные понятия и требования	37
Пример концепции SADT	40
Состав функциональной модели	41
Типы связей между функциями	44
Контрольные вопросы	46
6. Качество программных средств	47
Основные понятия	47
Пример стандарта, регламентирующего качество программных средств	53
Оценивание качества готового программного продукта	57
Характеристики качества баз данных	59
Модели оценки характеристик качества программного обеспечения	62
Размерно-ориентированные метрики	62
Функционально-ориентированные метрики	63
Контрольные вопросы	71
Библиографический список	72

Коршикова Лариса Александровна

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И СТАНДАРТИЗАЦИЯ

Учебное пособие

Редактор *Л.Н. Ветчакова*
Выпускающий редактор *И.П. Брованова*
Дизайн обложки *А.В. Ладыжская*
Компьютерная верстка *С.И. Ткачева*

Налоговая льгота – Общероссийский классификатор продукции
Издание соответствует коду 95 3000 ОК 005-93 (ОКП)

Подписано в печать 07.05.2018. Формат 60 × 84 1/16. Бумага офсетная. Тираж 100 экз.
Уч.-изд. л. 4,41. Печ. л. 4,75. Изд. № 58. Заказ № 725. Цена договорная

Отпечатано в типографии
Новосибирского государственного технического университета
630073, г. Новосибирск, пр. К. Маркса, 20