

МЕТРОЛОГИЯ (ТРИ МОДУЛЯ)

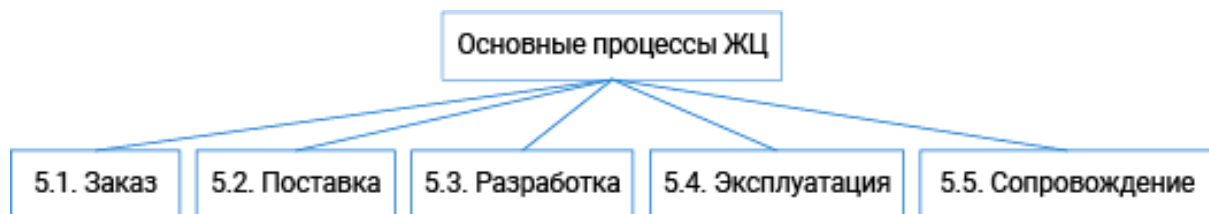
Стандарт СТБ ИСО/МЭК 12207-2003. Процессы ЖЦ ПС

Основные процессы ЖЦ ПС

Государственный стандарт СТБ ИСО/МЭК 12207 определяет жизненный цикл программных средств и систем в виде трехуровневой иерархической структуры. Основу ЖЦ составляет набор **процессов**. Каждый процесс разделен на набор **работ**. Каждая работа в свою очередь разделена на набор **задач**.

Стандарт регламентирует общее число процессов ЖЦ ПС, равное 17, которые разбиваются на три группы:

- **Основные процессы**, которые реализуются под управлением основных сторон, участвующих в ЖЦ ПС.
- **Вспомогательные процессы**, являющиеся целенаправленными составными частями других процессов. Их основное назначение – обеспечить успешную реализацию и качество выполнения программного проекта.
- **Организационные процессы**, предназначенные для создания в некоторой организации и совершенствования организационных структур, охватывающих процессы ЖЦ и соответствующий персонал.



На данном и на последующих рисунках сохранена нумерация процессов, принятая в СТБ ИСО/МЭК 12207-2003

Процесс заказа (The Acquisition Process)

Заинтересованная сторона – Заказчик

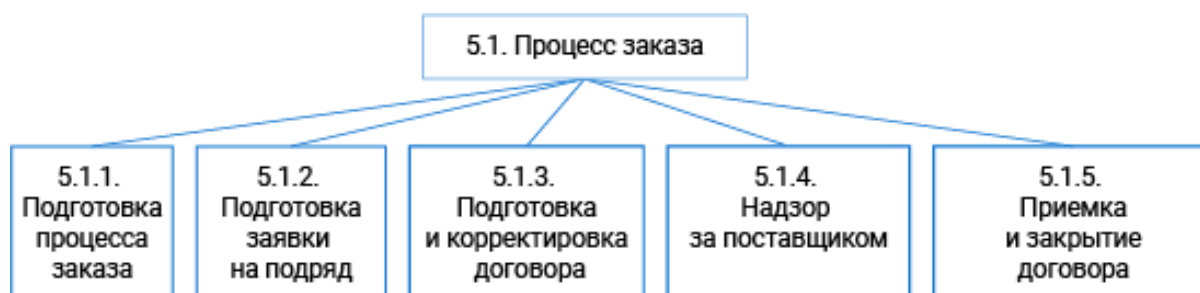


Таблица 1 содержит задачи, реализуемые при выполнении соответствующих работ процесса заказа, и типы выходных результатов данных задач, которые должны быть документально оформлены.

Таблица 1 – Задачи работ процесса заказа

№ задачи в ИСО/МЭК 12207	Описание задачи	Тип выходного результата задачи
5.1.1. Задачи работы <i>Подготовка процесса заказа</i>		
5.1.1.1	Описание концепции или потребности в заказе	Описание (в составе результата задачи 5.1.1.8)
5.1.1.2	Согласование проанализированных требований (выполняется, если анализ требований осуществляет не заказчик, а поставщик)	Описание (в составе результата 5.1.1.8)
5.1.1.3	Патент на селекционное достижение	
5.1.1.4	Анализ требований к программным средствам (выполняется заказчиком или, по его поручению, поставщиком)	Описание (в составе результата 5.1.1.8)
5.1.1.5	Использование процессов разработки при выполнении задач 5.1.1.2 и 5.1.1.4	

№ задачи в ИСО/МЭК 12207	Описание задачи	Тип выходного результата задачи
5.1.1.6	Рассмотрение вариантов реализации заказа (покупка готового ПП, разработка собственными силами, разработка на договорной основе, модернизация существующего ПП, комбинация вариантов)	Описание (в составе результата 5.1.1.8)
5.1.1.7	Оценка условий приобретения готового программного продукта	
5.1.1.8	Документальное оформление и выполнение плана заказа	План
5.1.1.9	Определение и документальное оформление правил и условий реализации договора	Описание
5.1.2. Задачи работы <i>Подготовка заявки на подряд</i>		
5.1.2.1	Документальное оформление требований к заказу (заявки на подряд)	Описание
5.1.2.2	Адаптация СТБ ИСО/МЭК 12207-2003 к условиям проекта	Описание (в составе результата 5.1.2.1)
5.1.2.3	Определение контрольных точек договора	Описание (в составе результата 5.1.2.1)
5.1.2.4	Предоставление требований к заказу поставщикам для тендера (конкурса)	
5.1.3. Задачи работы <i>Подготовка и корректировка договора</i>		
5.1.3.1	Определение процедуры выбора поставщика на тендерной (конкурсной) основе	Процедура
5.1.3.2	Выбор поставщика	

№ задачи в ИСО/МЭК 12207	Описание задачи	Тип выходного результата задачи
5.1.3.3	Окончательное решение по адаптации СТБ ИСО/МЭК 12207-2003 к условиям проекта	Описание (в составе результата 5.1.3.4)
5.1.3.4	Подготовка и заключение договора с поставщиком	Договор
5.1.3.5	Контроль изменений, вносимых в договор, в ходе его реализации	
5.1.4. Задачи работы <i>Надзор за поставщиком</i>		
5.1.4.1	Надзор за работами поставщика	
5.1.4.2	Взаимодействие с поставщиком для решения проблем	
5.1.5. Задачи работы <i>Приемка и закрытие договора</i>		
5.1.5.1	Подготовка к приемке	Процедура
5.1.5.2	Проведение приемочных испытаний в соответствии с условиями задачи 5.1.1.9	Протокол (общий с результатом 5.3.13.1)
5.1.5.3	Управление конфигурацией поставленного ПП	

Процесс поставки (The Supply Process)

Заинтересованная сторона – Поставщик

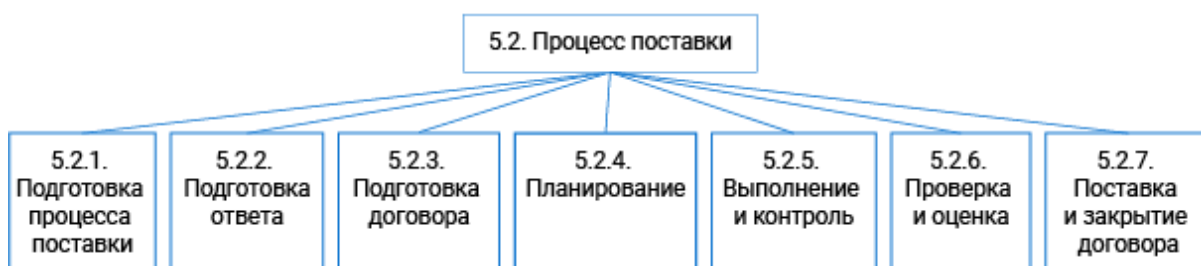


Таблица 2 содержит задачи, реализуемые при выполнении соответствующих работ процесса поставки, и типы выходных результатов данных задач, которые должны быть документально оформлены.

Таблица 2 – Задачи работ процесса поставки

№ задачи в ИСО/МЭК 12207	Описание задачи	Тип выходного результата задачи
5.2.1. Задачи работы <i>Подготовка процесса поставки</i>		
5.2.1.1	Анализ требований в заявке на подряд	
5.2.1.2	Принятие решения об участии в тендере или о подписании договора	
5.2.2. Задачи работы <i>Подготовка ответа</i>		
5.2.2.1	Подготовка предложения в ответ на заявку о подряде, включая предложения по адаптации СТБ ИСО/МЭК 12207-2003	Предложение
5.2.3. Задачи работы <i>Подготовка договора</i>		
5.2.3.1	Проведение переговоров с заказчиком	
5.2.3.2	Внесение изменений в текст договора по согласованию с заказчиком	

№ задачи в ИСО/МЭК 12207	Описание задачи	Тип выходного результата задачи
5.2.4. Задачи работы <i>Планирование</i>		
5.2.4.1	Анализ требований к заказу	
5.2.4.2	Определение модели жизненного цикла программных средств со структурированием в нее процессов, работ и задач СТБ ИСО/МЭК 12207-2003	Описание (в составе результата задачи 5.2.4.5)
5.2.4.3	Определение требований к планированию управления и обеспечения проекта, обеспечения качества ПП или программной услуги, ресурсов	Описание (в составе результата задачи 5.2.4.5)
5.2.4.4	Анализ вариантов разработки программного продукта или предоставления программной услуги	Описание (в составе результата задачи 5.2.4.5)
5.2.4.5	Разработка и документальное оформление плана (планов) управления проектом	План (ссылка)
5.2.5. Задачи работы <i>Выполнение и контроль</i>		
5.2.5.1	Реализация планов управления проектом	

№ задачи в ИСО/МЭК 12207	Описание задачи	Тип выходного результата задачи
5.2.5.2	Разработка, проведение опытной эксплуатации и сопровождение ПП в соответствии с процессами разработки, эксплуатации и сопровождения	
5.2.5.3	Надзор за реализацией проекта, выявление и решение проблем	
5.2.5.4	Управление и контроль деятельности субподрядчиков в соответствии с процессом заказа	Процедура
5.2.5.5	Взаимодействие с верифицирующей, аттестующей или проверяющей организацией	
5.2.5.6	Взаимодействие с другими исполнителями договора	
5.2.6. Задачи работы <i>Проверка и оценка</i>		
5.2.6.1	Координация работы по проверке выполнения договора	
5.2.6.2	Участие в совещаниях, подготовке приемки ПП, приемочных испытаниях,	

№ задачи в ИСО/МЭК 12207	Описание задачи	Тип выходного результата задачи
	совместных анализах и аудиторских проверках	
5.2.6.3	Выполнение верификации и аттестации ПП, программной услуги и процессов	
5.2.6.4	Предоставление заказчику отчетов об оценках, анализах, аудиторских проверках, испытаниях, решениях проблем	
5.2.6.5	Обеспечение заказчику доступа к ресурсам для проверки ПП или программной услуги	
5.2.6.6	Выполнение работ по обеспечению качества	
5.2.7. Задачи работы <i>Поставка и закрытие договора</i>		
5.2.7.1	Поставка ПП или программной услуги заказчику	
5.2.7.2	Помощь заказчику в поддержке поставленного ПП или программной услуги	

Процесс разработки (The Development Process)

Заинтересованная сторона – Разработчик

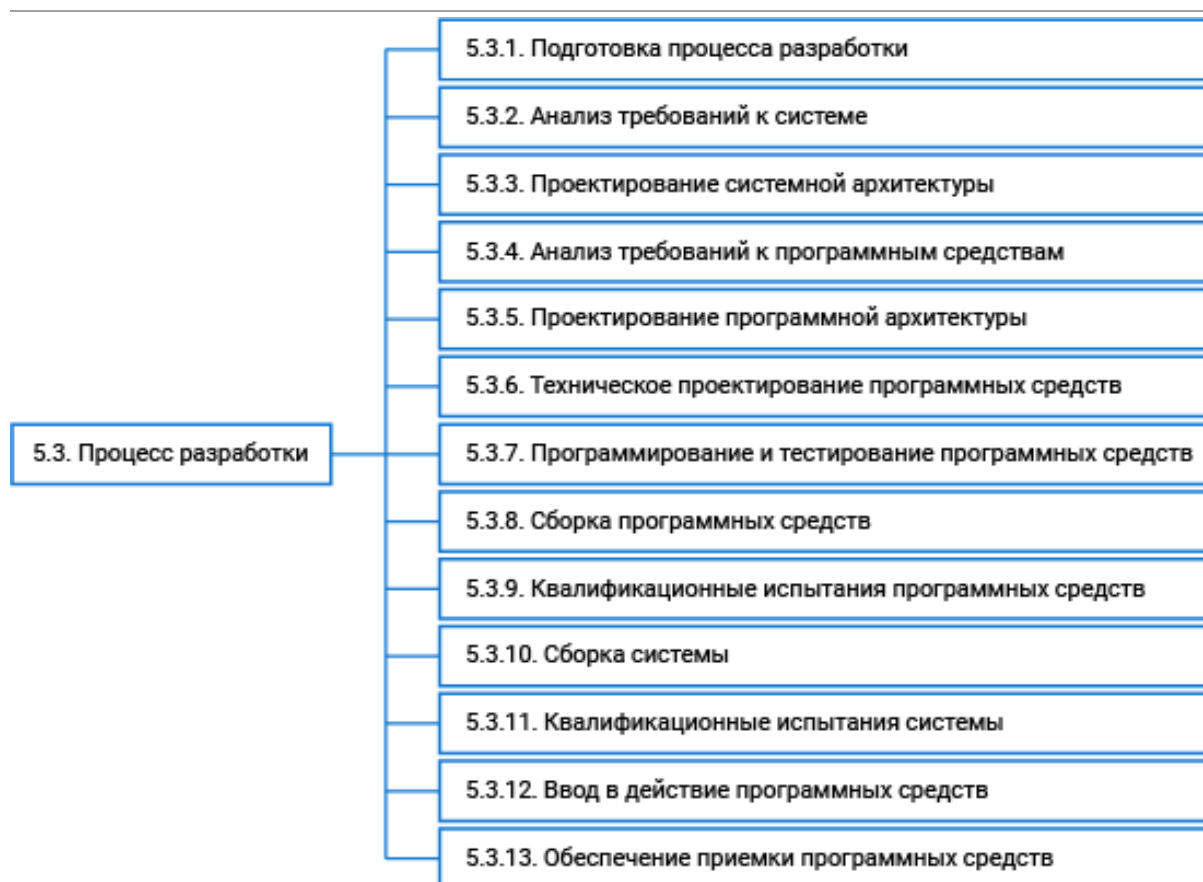


Таблица 3 содержит задачи, реализуемые при выполнении соответствующих работ процесса разработки, и типы выходных результатов данных задач, которые должны быть документально оформлены.

Таблица 3 – Задачи работы Подготовка процесса разработки

№ задачи в ИСО/МЭК 12207	Описание задачи	Тип выходного результата задачи
5.3.1. Задачи работы Подготовка процесса разработки		

№ задачи в ИСО/МЭК 12207	Описание задачи	Тип выходного результата задачи
5.3.1.1	Определение модели жизненного цикла программных средств со структурированием в нее процессов, работ и задач СТБ ИСО/МЭК 12207-2003 (если модель не определена в договоре)	Описание (в составе результата задачи 5.1.3.4)
5.3.1.2	Документальное оформление выходных результатов, управление конфигурацией выходных результатов, решение возникающих проблем, выполнение вспомогательных процессов в соответствии с процессами вспомогательной группы	Протоколы и отчеты в составе результатов соответствующих работ
5.3.1.3	Выбор и адаптация стандартов, методов, инструментариев, языков программирования	Описание (в составе результата 5.1.3.4)
5.3.1.4	Разработка и выполнение планов проведения работ процесса разработки	План
5.3.1.5	Поставка всех комплектующих изделий	
5.3.2. Задачи работы Анализ требований к системе		
5.3.2.1	Анализ области применения системы и определение требований к ней	Описание

№ задачи в ИСО/МЭК 12207	Описание задачи	Тип выходного результата задачи
5.3.2.2	Оценка требований к системе по критериям	Отчет
5.3.3. Задачи работы Проектирование системной архитектуры		
5.3.3.1	Определение общей архитектуры системы, распределение требований к ней между объектами технических и программных средств архитектуры и ручными операциями и дальнейшее уточнение требований	Описание
5.3.3.2	Оценка архитектуры системы и требований к объектам архитектуры по критериям	Отчет
5.3.4. Задачи работы Анализ требований к программным средствам		
5.3.4.1	Определение требований к ПС	Описание
5.3.4.2	Оценка требований к ПС по критериям	Отчет
5.3.4.3	Проведение совместных анализов в соответствии с процессом совместного анализа	Протокол

№ задачи в ИСО/МЭК 12207	Описание задачи	Тип выходного результата задачи
5.3.5. Задачи работы Проектирование программной архитектуры		
5.3.5.1	Преобразование требований к программному объекту в его архитектуру, распределение требований к программному объекту между его компонентами и дальнейшее уточнение требований (разработка эскизного проекта)	Описание
5.3.5.2	Разработка эскизного проекта интерфейсов программного объекта и его компонентов	Описание
5.3.5.3	Разработка эскизного проекта базы данных	Описание
5.3.5.4	Разработка предварительных версий документации пользователя	Руководство
5.3.5.5	Разработка предварительных требований к испытаниям (тестированию) программного объекта и графика сборки ПП	Описание, план

№ задачи в ИСО/МЭК 12207	Описание задачи	Тип выходного результата задачи
5.3.5.6	Оценка архитектуры программного объекта и эскизных проектов интерфейсов и базы данных по критериям	Отчет
5.3.5.7	Проведение совместных анализов в соответствии с процессом совместного анализа	
5.3.6. Задачи работы Техническое проектирование программных средств		
5.3.6.1	Разработка технического проекта для компонентов программного объекта (представление их в виде набора программных модулей), распределение технических требований к компонентам между программными модулями и дальнейшее уточнение требований	Описание
5.3.6.2	Разработка технического проекта интерфейсов программного объекта, его компонентов и модулей	Описание
5.3.6.3	Разработка технического проекта базы данных	Описание
5.3.6.4	Уточнение документации пользователя	Руководство

№ задачи в ИСО/МЭК 12207	Описание задачи	Тип выходного результата задачи
5.3.6.5	Разработка требований к испытаниям и программе испытаний программных модулей	Описание, план
5.3.6.6	Уточнение общих требований к испытанию (тестированию) и программе сборки программных средств	Описание, план
5.3.6.7	Оценка технического проекта и требований к тестированию по критериям	Отчет
5.3.6.8	Проведение совместных анализов в соответствии с процессом совместного анализа	
5.3.7. Задачи работы Программирование и тестирование программных средств		
5.3.7.1	Программирование программных модулей, базы данных, разработка процедур и данных для их испытаний (тестирования)	Программный продукт, описание, процедура
5.3.7.2	Тестирование программных модулей и базы данных	Протокол

№ задачи в ИСО/МЭК 12207	Описание задачи	Тип выходного результата задачи
5.3.7.3	Уточнение документации пользователя	Руководство
5.3.7.4	Уточнение общих требований к тестированию и программы сборки программного средства	Описание, план
5.3.7.5	Оценка результатов программирования и тестирования по критериям	Отчет
5.3.8. Задачи работы Сборка программных средств		
5.3.8.1	Разработка плана сборки для объединения программных модулей и компонентов в программный объект	План
5.3.8.2	Сборка программных модулей и компонентов и тестирование результатов сборки	Описание, протокол
5.3.8.3	Уточнение документации пользователя	Руководство
5.3.8.4	Разработка плана и процедуры квалификационных испытаний программных средств	План, процедура

№ задачи в ИСО/МЭК 12207	Описание задачи	Тип выходного результата задачи
5.3.8.5	Оценка планов и результатов сборки и тестирования, а также документации пользователя по критериям	Отчет
5.3.8.6	Проведение совместного анализа в соответствии с процессом совместного анализа	
5.3.9. Задачи работы Квалификационные испытания программных средств		
5.3.9.1	Проведение квалификационных испытаний (тестирования) ПС	Протокол
5.3.9.2	Уточнение документации пользователя	Руководство
5.3.9.3	Оценка результатов испытаний и документации пользователя по критериям	Отчет
5.3.9.4	Обеспечение проведения аудиторской проверки в соответствии с процессом аудита	Протокол
5.3.9.5	Доработка и подготовка ПП к последующим работам, определение	Протокол

№ задачи в ИСО/МЭК 12207	Описание задачи	Тип выходного результата задачи
	состояния конфигурации (базовой линии) проекта	
5.3.10. Задачи работы Сборка системы		
5.3.10.1	Сборка объектов программной и технической конфигурации, ручных операций, других систем в единую систему, испытания собранной системы	Описание, протокол
5.3.10.2	Разработка плана и процедуры квалификационных испытаний системы	План, процедура
5.3.10.3	Оценка собранной системы по критериям	Отчет
5.3.11. Задачи работы Квалификационные испытания системы		
5.3.11.1	Проведение квалификационных испытаний системы	Описание, протокол
5.3.11.2	Оценка системы по критериям	Отчет

№ задачи в ИСО/МЭК 12207	Описание задачи	Тип выходного результата задачи
5.3.11.3	Обеспечение проведения аудиторской проверки в соответствии с процессом аудита	Протокол
5.3.11.4	Доработка и подготовка ПП к последующим работам, определение состояния конфигурации (базовой линии) проекта	Протокол
5.3.12. Задачи работы Ввод в действие программных средств		
5.3.12.1	Разработка плана ввода в действие ПП в среде эксплуатации	План
5.3.12.2	Ввод в действие ПП в соответствии с планом	Протокол
5.3.13. Задачи работы Обеспечение приемки программных средств		
5.3.13.1	Обеспечение проведения заказчиком приемочных испытаний	Процедура, протокол
5.3.13.2	Укомплектование и поставка ПП заказчику на условиях договора	Программный продукт

№ задачи в ИСО/МЭК 12207	Описание задачи	Тип выходного результата задачи
5.3.13.3	Обеспечение обучения и поддержка персонала заказчика на условиях договора	

Процесс эксплуатации (The Operation Process)

Заинтересованная сторона – Оператор

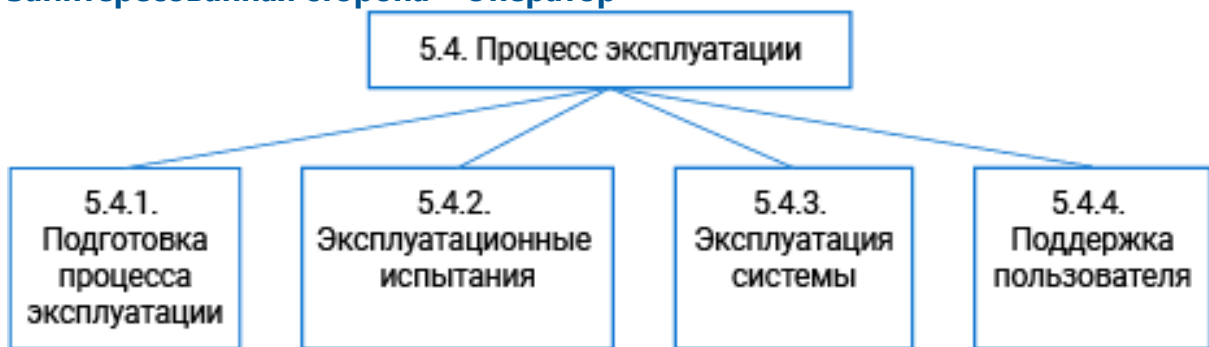


Таблица 4 содержит задачи, реализуемые при выполнении соответствующих работ процесса эксплуатации, и типы выходных результатов данных задач, которые должны быть документально оформлены.

Таблица 4 Задачи работ процесса эксплуатации

№ задачи в ИСО/МЭК 12207	Описание задачи	Тип выходного результата задачи
5.4.1. Задачи работы <i>Подготовка процесса эксплуатации</i>		
5.4.1.1	Разработка плана эксплуатации и определение набора стандартов по эксплуатации	План

№ задачи в ИСО/МЭК 12207	Описание задачи	Тип выходного результата задачи
5.4.1.2	Установка процедур документирования и решения проблем	Процедура
5.4.1.3	Установка процедур для тестирования ПП в эксплуатационной среде, ввода сообщений о проблеме в процесс сопровождения и ввода ПП в эксплуатацию	Процедура
5.4.2. Задачи работы <i>Эксплуатационные испытания</i>		
5.4.2.1	Проведение эксплуатационных испытаний и ввод программных продуктов в промышленную эксплуатацию	Протокол
5.4.2.2	Обеспечение инициализации и эксплуатации программ и базы данных в соответствии с планом эксплуатации	
5.4.3. Задачи работы <i>Эксплуатация системы</i>		
5.4.3.1	Эксплуатация в установленной среде в соответствии с документацией пользователя	
5.4.4. Задачи работы <i>Поддержка пользователя</i>		

№ задачи в ИСО/МЭК 12207	Описание задачи	Тип выходного результата задачи
5.4.4.1	Помощь и консультации пользователям в установленном порядке	Протокол
5.4.4.2	Передача запросов пользователя в процесс сопровождения и контроль их решения	Протокол
5.4.4.3	Обеспечение временного решения проблем	

Процесс сопровождения (The Maintenance Process)

Заинтересованная сторона – Персонал сопровождения

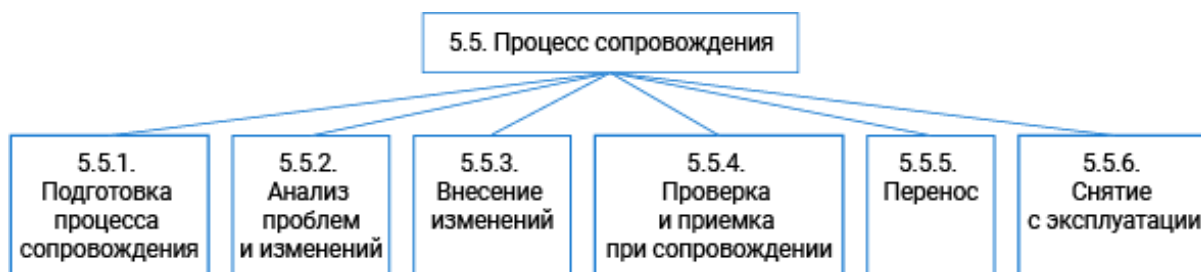


Таблица 5 содержит задачи, реализуемые при выполнении соответствующих работ процесса сопровождения, и типы выходных результатов данных задач, которые должны быть документально оформлены.

Таблица 5 – Задачи работ процесса сопровождения

№ задачи в ИСО/МЭК 12207	Описание задачи	Тип выходного результата задачи
5.5.1. Задачи работы <i>Подготовка процесса сопровождения</i>		
5.5.1.1	Разработка планов и процедур для проведения работ по сопровождению	План, процедура
5.5.1.2	Определение процедур для документирования возникающих проблем и организации связи с процессом решения проблем	Процедура
5.5.1.3	Реализация процесса управления конфигурацией для управления изменениями существующей системы	
5.5.2. Задачи работы <i>Анализ проблем и изменений</i>		
5.5.2.1	Анализ сообщений о проблеме	
5.5.2.2	Дублирование или верификация проблемы	
5.5.2.3	Разработка вариантов реализации изменения	
5.5.2.4	Документальное оформление сообщения о проблеме и вариантов ее решения	Протокол, отчет

№ задачи в ИСО/МЭК 12207	Описание задачи	Тип выходного результата задачи
5.5.2.5	Согласование выбранного варианта изменения в соответствии с договором	
5.5.3. Задачи работы <i>Внесение изменений</i>		
5.5.3.1	Определение документов и программных модулей, требующих изменения	Протокол
5.5.3.2	Реализация изменений с использованием процесса разработки, разработка процедур испытаний и оценки результатов испытаний	Процедура, протокол, отчет
5.5.4. Задачи работы <i>Проверка и приемка при сопровождении</i>		
5.5.4.1	Совместная проверка работоспособности измененной системы	
5.5.4.2	Приемка внесенного изменения	Протокол
5.5.5. Задачи работы <i>Перенос</i>		
5.5.5.1	Обеспечение соответствия измененного при переносе в новую эксплуатационную среду	

№ задачи в ИСО/МЭК 12207	Описание задачи	Тип выходного результата задачи
	программного продукта или данных стандарту СТБ ИСО/МЭК 12207-2003	
5.5.5.2	Разработка и выполнение плана переноса программного продукта	План
5.5.5.3	Уведомление пользователей о планах и работах по переносу программного продукта	Отчет
5.5.5.4	Обеспечение параллельной эксплуатации в прежней и новой среде (при необходимости)	
5.5.5.5	Уведомление заинтересованных сторон о выполненном переносе, передача в архив документации по прежней среде	
5.5.5.6	Итоговый анализ влияния переноса на эксплуатацию системы и рассылка результатов анализа заинтересованным сторонам	Отчет
5.5.5.7	Обеспечение доступности информации и документации, связанных с прежней средой	
5.5.6. Задачи работы <i>Снятие с эксплуатации</i>		

№ задачи в ИСО/МЭК 12207	Описание задачи	Тип выходного результата задачи
5.5.6.1	Разработка плана снятия с эксплуатации	План
5.5.6.2	Уведомление пользователей о планах и работах по снятию с эксплуатации	Отчет
5.5.6.3	Проведение параллельной эксплуатации прежнего и нового программных продуктов	
5.5.6.4	Уведомление заинтересованных сторон о снятии ПП с эксплуатации и передача документации о нем в архив	
5.5.6.5	Обеспечение доступности информации по снятому с эксплуатации ПП	

Вспомогательные процессы ЖЦ ПС



Процесс документирования (The Process)

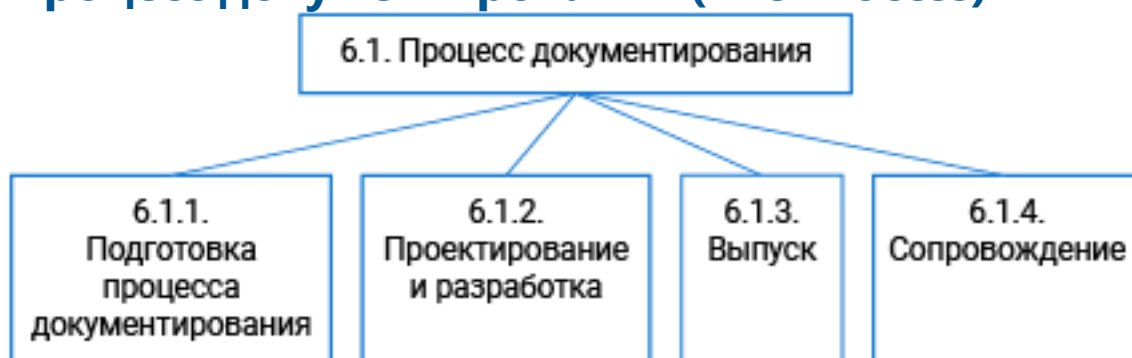


Таблица 6 содержит задачи, реализуемые при выполнении соответствующих работ процесса документирования, и типы выходных результатов данных задач, которые должны быть документально оформлены.

Таблица 6 – Задачи работ процесса документирования

№ задачи в ИСО/МЭК 12207	Описание задачи	Тип выходного результата задачи
6.1.1. Задачи работы <i>Подготовка процесса документирования</i>		
6.1.1.1	Разработка и реализация плана обозначения документов, выпускаемых в процессах ЖЦ ПП	План (ссылка)
6.1.2. Задачи работы <i>Проектирование и разработка</i>		
6.1.2.1	Проектирование документов согласно стандартам на документацию	Описание

№ задачи в ИСО/МЭК 12207	Описание задачи	Тип выходного результата задачи
6.1.2.2	Подтверждение источника и соответствия исходных материалов для документов	
6.1.2.3	Проверка и редактирование документов согласно стандартам, утверждение компетентными лицами	
6.1.3. Задачи работы <i>Выпуск</i>		
6.1.3.1	Издание и распространение документов в соответствии с планом	
6.1.3.2	Управление документированием в соответствии с процессом управления конфигурацией	
6.1.4. Задачи работы <i>Сопровождение</i>		
6.1.4.1	Внесение изменений в документацию согласно процессам сопровождения и управления конфигурацией	

Процесс управления конфигурацией (The Configuration Management Process)

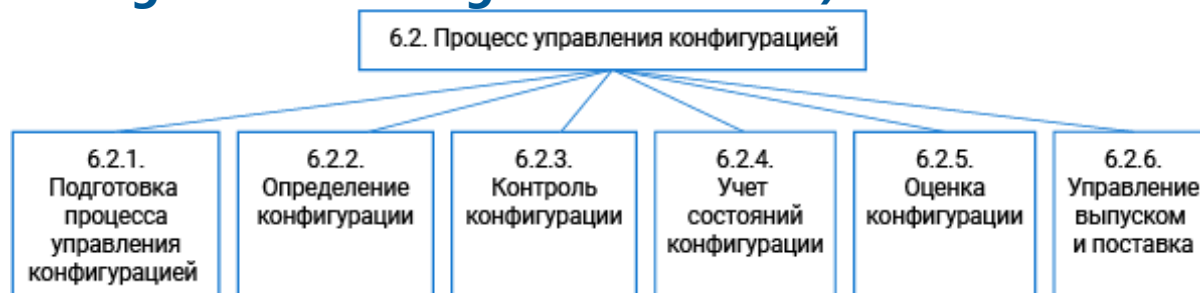


Таблица 7 содержит задачи, реализуемые при выполнении соответствующих работ процесса управления конфигурацией, и типы выходных результатов данных задач, которые должны быть документально оформлены.

Таблица 7 – Задачи работ процесса управления конфигурацией

№ задачи в ИСО/МЭК 12207	Описание задачи	Тип выходного результата задачи
6.2.1. Задачи работы <i>Подготовка процесса управления конфигурацией</i>		
6.2.1.1	Разработка плана управления конфигурацией	План
6.2.2. Задачи работы <i>Определение конфигурации</i>		
6.2.2.1	Определение схемы обозначения программных объектов и их версий (объектов программной конфигурации) и документации, в которой фиксируется состояние их конфигурации	Описание

№ задачи в ИСО/МЭК 12207	Описание задачи	Тип выходного результата задачи
6.2.3. Задачи работы <i>Контроль конфигурации</i>		
6.2.3.1	Регистрация заявок на внесение изменений; анализ и оценка изменений; принятие или непринятие заявки; реализация, верификация и выпуск измененного программного объекта; обеспечение аудиторских проверок изменений	
6.2.4. Задачи работы <i>Учет состояний конфигурации</i>		
6.2.4.1	Подготовка протоколов управления конфигурацией и отчетов о состоянии контролируемых программных объектов	Протокол, отчет
6.2.5. Задачи работы <i>Оценка конфигурации</i>		
6.2.5.1	Определение и обеспечение функциональной законченности и физической завершенности программных объектов	Протокол, отчет
6.2.6. Задачи работы <i>Управление выпуском и поставка</i>		
6.2.6.1	Контроль выпуска и поставки ПП и документации	

Процесс обеспечения качества (The Quality Assurance Process)

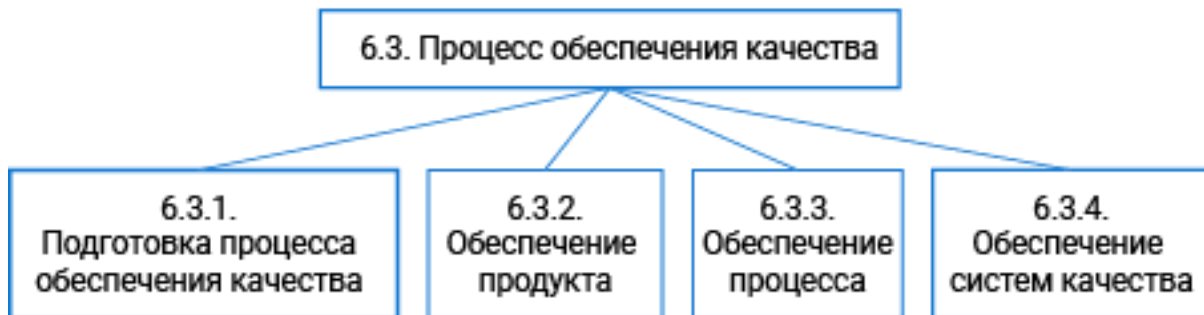


Таблица 8 содержит задачи, реализуемые при выполнении соответствующих работ процесса обеспечения качества, и типы выходных результатов данных задач, которые должны быть документально оформлены.

Таблица 8 – Задачи работ процесса обеспечения качества

№ задачи в ИСО/МЭК 12207	Описание задачи	Тип выходного результата задачи
6.3.1. Задачи работы <i>Подготовка процесса обеспечения качества</i>		
6.3.1.1	Адаптация процесса обеспечения качества к условиям конкретного проекта, определение целей процесса обеспечения качества	Описание (в составе результата задачи 6.3.1.3)
6.3.1.2	Координация с процессами верификации, аттестации, совместного анализа и аудита	

№ задачи в ИСО/МЭК 12207	Описание задачи	Тип выходного результата задачи
6.3.1.3	Разработка и реализация плана обеспечения качества	План
6.3.1.4	Выполнение запланированных и традиционных работ и задач по обеспечению качества, при необходимости организация связи с процессом решения проблем	Отчет, протокол
6.3.1.5	Предоставление отчетов о работах и задачах по обеспечению качества заказчику согласно договору	
6.3.1.6	Обеспечение организационной независимости лиц, реализующих работы и задачи по обеспечению качества	
6.3.2. Задачи работы <i>Обеспечение продукта</i>		
6.3.2.1	Обеспечение соответствия условиям договора и выполнения всех планов	
6.3.2.2	Обеспечение соответствия ПП и документации условиям договора и планам	

№ задачи в ИСО/МЭК 12207	Описание задачи	Тип выходного результата задачи
6.3.2.3	Обеспечение соответствия поставляемых ПП требованиям, установленным в договоре	
6.3.3. Задачи работы <i>Обеспечение процесса</i>		
6.3.3.1	Обеспечение соответствия процессов жизненного цикла программных средств условиям договора и утвержденным планам	
6.3.3.2	Обеспечение соответствия технологий программирования, условий разработки, условий испытаний и архивных библиотек условиям договора	
6.3.3.3	Обеспечение соответствия ПП, разработанных субподрядчиком, требованиям основного договора	
6.3.3.4	Обеспечение взаимной поддержки заказчика и других участников договора в соответствии с условиями договора и планами	
6.3.3.5	Обеспечение соответствия характеристик ПП и процессов установленным стандартам и процедурам	

№ задачи в ИСО/МЭК 12207	Описание задачи	Тип выходного результата задачи
6.3.3.6	Обеспечение соответствия и обучения исполнителей проекта	
6.3.4. Задачи работы <i>Обеспечение систем качества</i>		
6.3.4.1	Обеспечение проведения дополнительных работ по управлению качеством в соответствии с разделами СТБ ИСО 9001, указанными в договоре	Протокол

Процесс верификации (The Verification Process)

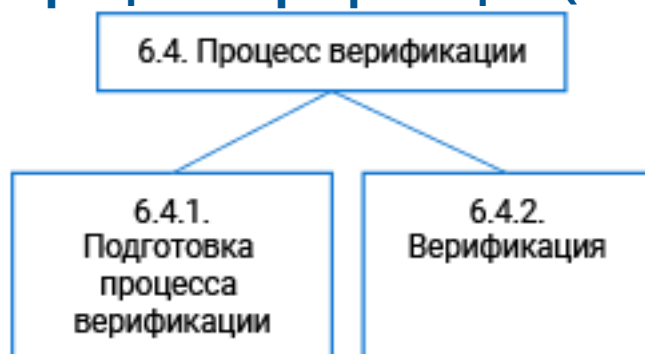


Таблица 9 содержит задачи, реализуемые при выполнении соответствующих работ процесса верификации, и типы выходных результатов данных задач, которые должны быть документально оформлены.

Таблица 9 – Задачи работ процесса верификации

№ задачи в ИСО/МЭК 12207	Описание задачи	Тип выходного результата задачи
6.4.1. Задачи работы <i>Подготовка процесса верификации</i>		
6.4.1.1	Определение необходимости в проекте работ по верификации и степени их организационной независимости, анализ критичности проектных требований	Описание (в составе результата задачи 6.4.1.5)
6.4.1.2	Установка процесса верификации (при необходимости)	Описание (в составе результата задачи 6.4.1.5)
6.4.1.3	Выбор соответствующей независимой квалифицированной организации (при необходимости)	Описание (в составе результата задачи 6.4.1.5)
6.4.1.4	Определение верифицируемых работ и продуктов, выбор работ и задач верификации	Описание (в составе результата задачи 6.4.1.5)
6.4.1.5	Разработка плана верификации	План
6.4.1.6	Реализация плана проведения верификации, обеспечение доступности его результатов	Отчет, протокол

№ задачи в ИСО/МЭК 12207	Описание задачи	Тип выходного результата задачи
	заказчику, при необходимости организация связи с процессом решения проблем	
6.4.2. Задачи работы <i>Верификация</i>		
6.4.2.1	Верификация договора по критериям	
6.4.2.2	Верификация процесса по критериям	
6.4.2.3	Верификация требований по критериям	
6.4.2.4	Верификация результатов проектирования по критериям	
6.4.2.5	Верификация исходных текстов программных модулей по критериям	
6.4.2.6	Верификация сборки по критериям	
6.4.2.7	Верификация документации по критериям	

Процесс аттестации (The Validation Process)

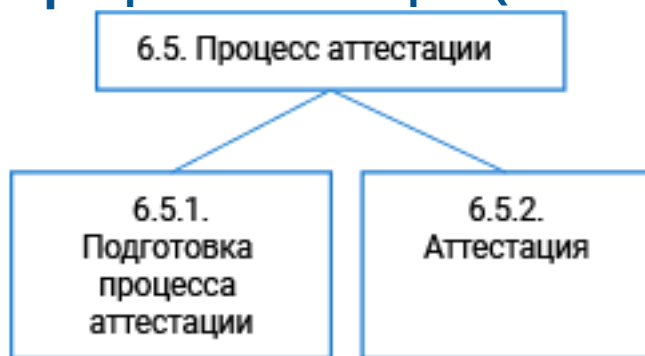


Таблица 10 содержит задачи, реализуемые при выполнении соответствующих работ процесса аттестации, и типы выходных результатов данных задач, которые должны быть документально оформлены.

Таблица – 10 Задачи работ процесса аттестации

№ задачи в ИСО/МЭК 12207	Описание задачи	Тип выходного результата задачи
6.5.1. Задачи работы <i>Подготовка процесса аттестации</i>		
6.5.1.1	Определение необходимости в проекте работ по аттестации и степени их организационной независимости	
6.5.1.2	Установка процесса, задач, методов, методик и средств аттестации (при необходимости)	Описание (в составе результата задачи 6.5.1.4)

№ задачи в ИСО/МЭК 12207	Описание задачи	Тип выходного результата задачи
6.5.1.3	Выбор соответствующей независимой квалифицированной организации (при необходимости)	
6.5.1.4	Разработка плана аттестации	План
6.5.1.5	Реализация плана проведения аттестации, обеспечение доступности его результатов заказчику, при необходимости организация связи с процессом решения проблем	Отчет
6.5.2. Задачи работы <i>Аттестация</i>		
6.5.2.1	Подготовка выбранных требований к испытаниям (тестированию), контрольных примеров и технических условий испытаний к анализу результатов испытаний	Описание
6.5.2.2	Обеспечение соответствия требований к испытаниям (тестированию), контрольных примеров и технических условий испытаний требованиям к объектам аттестации	
6.5.2.3	Проведение испытаний	Протокол

№ задачи в ИСО/МЭК 12207	Описание задачи	Тип выходного результата задачи
6.5.2.4	Подтверждение соответствия ПП заданным возможностям	Протокол
6.5.2.5	Проведение испытаний ПП в заданном контексте использования	Протокол

Процесс совместного анализа (The Joint Review Process)

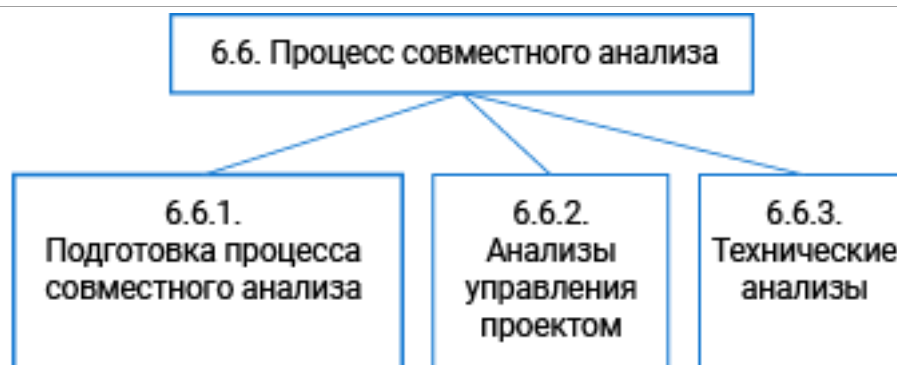


Таблица 11 содержит задачи, реализуемые при выполнении соответствующих работ процесса совместного анализа, и типы выходных результатов данных задач, которые должны быть документально оформлены.

Таблица – 11 Задачи работ процесса совместного анализа

№ задачи в ИСО/МЭК 12207	Описание задачи	Тип выходного результата задачи
6.6.1. Задачи работы <i>Подготовка процесса совместного анализа</i>		

№ задачи в ИСО/МЭК 12207	Описание задачи	Тип выходного результата задачи
6.6.1.1	Организация проведения периодических и целевых анализов хода работ в установленные сроки	
6.6.1.2	Согласование ресурсов, необходимых для проведения анализа, между участвующими в нем сторонами	Протокол
6.6.1.3	Согласование плана, объема, процедур и критериев проведения анализа, состава анализируемых результатов работ	Протокол
6.6.1.4	Передача выявленных при проведении анализа проблем в процесс решения проблем	
6.6.1.5	Передача результатов анализа заинтересованным сторонам	
6.6.1.6	Согласование результатов анализа, принимаемых обязательств и критериев завершения анализа	Протокол
6.6.2. Задачи работы <i>Анализы управления проектом</i>		

№ задачи в ИСО/МЭК 12207	Описание задачи	Тип выходного результата задачи
6.6.2.1	Оценка соответствия состояния проекта проектным планам, графикам, стандартам и руководствам	Отчет, протокол
6.6.3. Задачи работы <i>Технические анализы</i>		
6.6.3.1	Оценка соответствия состояния создаваемых программных продуктов или услуг проектным планам, графикам, стандартам и руководствам	Отчет

Процесс аудита (The Audit Process)

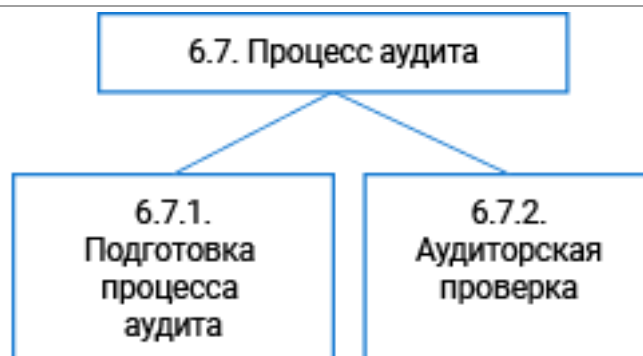


Таблица 12 содержит задачи, реализуемые при выполнении соответствующих работ процесса аудита, и типы выходных результатов данных задач, которые должны быть документально оформлены.

Таблица – 12 Задачи работ процесса аудита

№ задачи в ИСО/МЭК 12207	Описание задачи	Тип выходного результата задачи
6.7.1. Задачи работы <i>Подготовка процесса аудита</i>		
6.7.1.1	Организация проведения аудиторских проверок в сроки согласно плану	
6.7.1.2	Обеспечение независимости аудиторского персонала	
6.7.1.3	Согласование ресурсов, необходимых для проведения аудита, между участвующими в нем сторонами	Протокол
6.7.1.4	Согласование плана, объема, процедур и критериев проведения аудиторской проверки, состава проверяемых результатов работ	Протокол
6.7.1.5	Передача выявленных при проведении аудиторской проверки проблем в процесс решения проблем	Протокол
6.7.1.6	Передача результатов аудиторской проверки заинтересованным сторонам	

№ задачи в ИСО/МЭК 12207	Описание задачи	Тип выходного результата задачи
6.7.1.7	Согласование результатов, принимаемых обязательств и критериев завершения аудиторской проверки	Протокол
6.7.2. Задачи работы <i>Аудиторская проверка</i>		
6.7.2.1	Проведение аудиторской проверки по критериям (ссылка)	Отчет

Процесс решения проблем (The Problem Resolution Process)

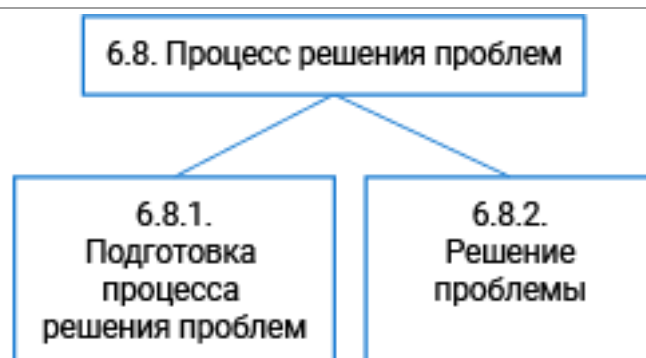
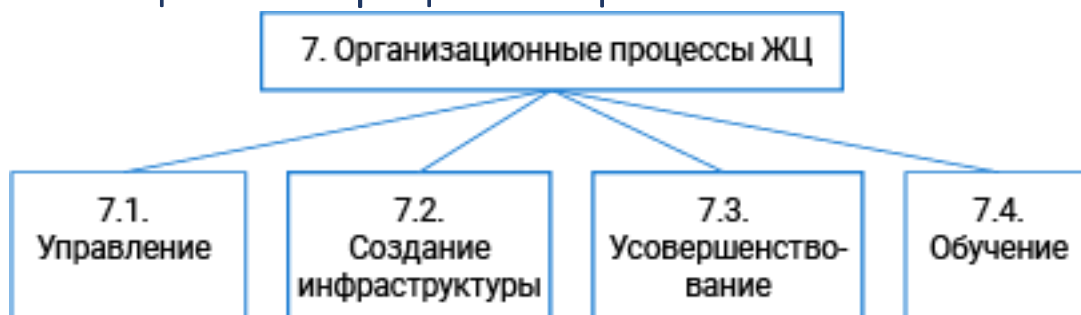


Таблица 13 содержит задачи, реализуемые при выполнении соответствующих работ процесса решения проблем, и типы выходных результатов данных задач, которые должны быть документально оформлены.

Таблица – 13 Задачи работ процесса решения проблем

№ задачи в ИСО/МЭК 12207	Описание задачи	Тип выходного результата задачи
6.8.1. Задачи работы Подготовка процесса решения проблем		
6.8.1.1	Установка процесса решения проблем для обработки всех проблем, выявленных в программных продуктах и работах	Протокол
6.8.2. Задачи работы <i>Решение проблемы</i>		
6.8.2.1	Отслеживание проблем путем их выявления, исследования, анализа и решения и их документирование	Отчет

Организационные процессы ЖЦ ПС



Процесс управления (The Management Process)

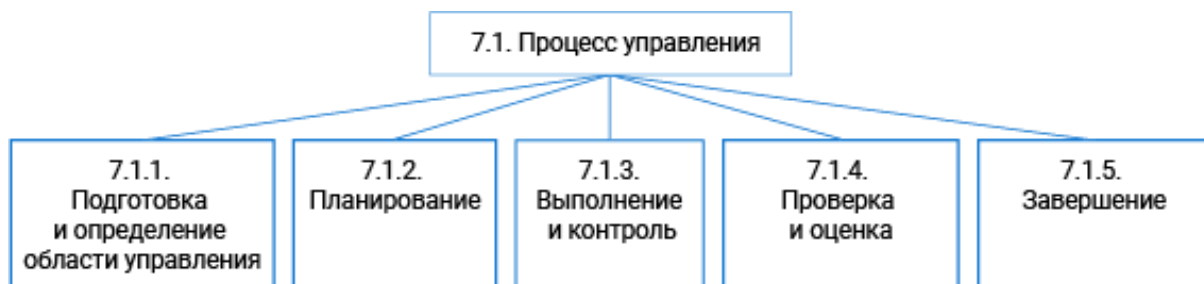


Таблица 14 содержит задачи, реализуемые при выполнении соответствующих работ процесса управления, и типы выходных результатов данных задач, которые должны быть документально оформлены.

Таблица 14 – Задачи работ процесса управления

№ задачи в ИСО/МЭК 12207	Описание задачи	Тип выходного результата задачи
7.1.1. Задачи работы <i>Подготовка и определение области управления</i>		
7.1.1.1	Установка требований к реализуемому процессу	Описание
7.1.1.2	Определение возможности реализации процесса, проверка соответствия ресурсов и сроков	Описание
7.1.1.3	Изменение требований к процессу для удовлетворения критериев его завершения (при необходимости)	Протокол

№ задачи в ИСО/МЭК 12207	Описание задачи	Тип выходного результата задачи
7.1.2. Задачи работы <i>Планирование</i>		
7.1.2.1	Разработка планов выполнения процессов	План (ссылка)
7.1.3. Задачи работы <i>Выполнение и контроль</i>		
7.1.3.1	Управление процессом на базе разработанного плана	
7.1.3.2	Текущий надзор за выполнением процесса	Отчет (в составе результата задачи 7.1.3.4)
7.1.3.3	Исследование и решение проблем, обнаруженных при выполнении процесса, при необходимости изменение планов	Отчет (в составе результата задачи 7.1.3.4)
7.1.3.4	Отчет в установленные сроки о реализации процесса	Отчет
7.1.4. Задачи работы <i>Проверка и оценка</i>		

№ задачи в ИСО/МЭК 12207	Описание задачи	Тип выходного результата задачи
7.1.4.1	Обеспечение оценки ПП и планов на соответствие требованиям	
7.1.4.2	Проверка результатов оценок программных продуктов, работ и задач на соответствие целям и планам	Отчет
7.1.5. Задачи работы <i>Завершение</i>		
7.1.5.1	Оценка всех программных продуктов, работ и задач по критериям, установленным в договоре	Отчет
7.1.5.2	Контроль результатов и полноты документации созданных ПП и выполненных работ, передача их в архив	

Процесс создания инфраструктуры (The Infrastructure Process)



Таблица 15 содержит задачи, реализуемые при выполнении соответствующих работ процесса создания инфраструктуры, и типы выходных результатов данных задач, которые должны быть документально оформлены.

Таблица – 15 Задачи работ процесса создания инфраструктуры

№ задачи в ИСО/МЭК 12207	Описание задачи	Тип выходного результата задачи
7.2.1. Задачи работы <i>Подготовка процесса создания инфраструктуры</i>		
7.2.1.1	Определение инфраструктуры, удовлетворяющей требованиям к процессу	Описание (в составе результата задачи 7.2.1.2)
7.2.1.2	Разработка плана создания установленной инфраструктуры	План
7.2.2. Задачи работы <i>Создание инфраструктуры</i>		
7.2.2.1	Разработка конфигурации инфраструктуры с учетом характеристик продукта и процесса	Описание
7.2.2.2	Создание инфраструктуры к требуемому сроку	
7.2.3. Задачи работы <i>Сопровождение инфраструктуры</i>		

№ задачи в ИСО/МЭК 12207	Описание задачи	Тип выходного результата задачи
7.2.3.1	Сопровождение, контроль и изменение инфраструктуры для удовлетворения требований к процессу, обеспечение связи с процессом управления конфигурацией	Описание

Процесс усовершенствования (The Improvement Process)

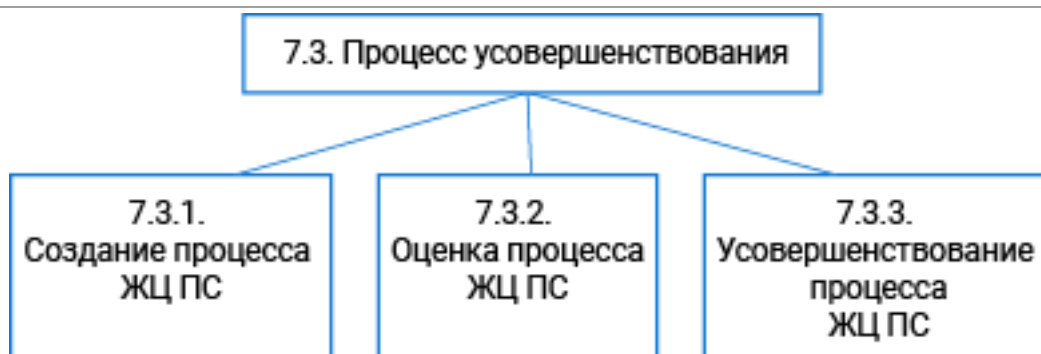


Таблица 16 содержит задачи, реализуемые при выполнении соответствующих работ процесса усовершенствования, и типы выходных результатов данных задач, которые должны быть документально оформлены.

Таблица – 16 Задачи работ процесса усовершенствования

№ задачи в ИСО/МЭК 12207	Описание задачи	Тип выходного результата задачи
7.3.1. Задачи работы <i>Создание процесса</i>		
7.3.1.1	Определение набора организационных процессов для всех процессов ЖЦ ПС и	Процедура

№ задачи в ИСО/МЭК 12207	Описание задачи	Тип выходного результата задачи
	механизма управления процессом усовершенствования	
7.3.2. Задачи работы <i>Оценка процесса</i>		
7.3.2.1	Разработка и применение процедуры оценки процесса	Процедура
7.3.2.2	Планирование и выполнение анализов процессов в установленные сроки	План, отчет
7.3.3. Задачи работы <i>Усовершенствование процесса</i>		
7.3.3.1	Усовершенствование процесса (при необходимости)	Отчет
7.3.3.2	Сбор и анализ архивных, технических и оценочных данных по выполняемым процессам для их усовершенствования	
7.3.3.3	Сбор, обновление и использование данных о расходах для усовершенствования организационных процессов административной деятельности	

Процесс обучения (The Training Process)

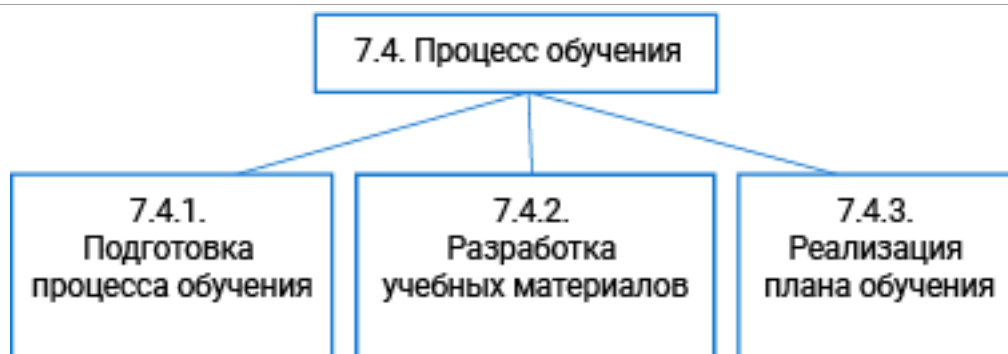


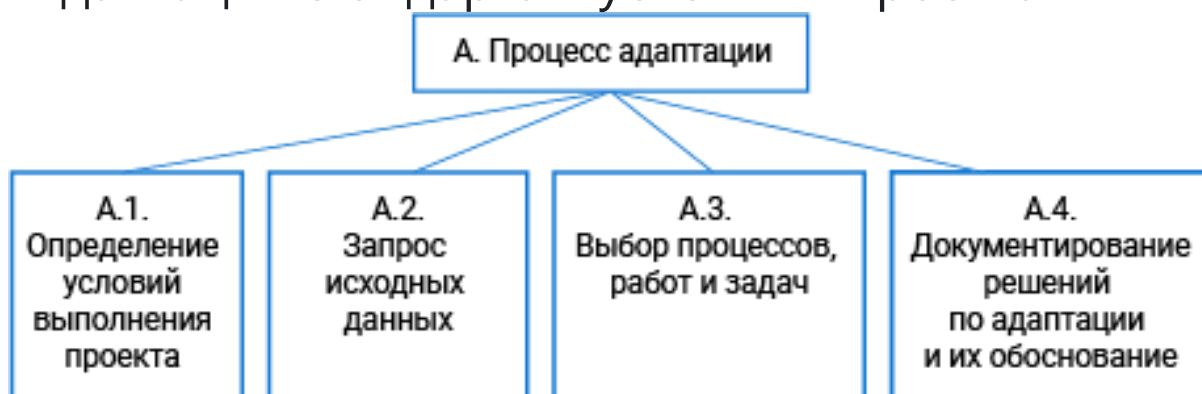
Таблица 17 содержит задачи, реализуемые при выполнении соответствующих работ процесса обучения, и типы выходных результатов данных задач, которые должны быть документально оформлены.

Таблица – 17 Задачи работ процесса обучения

№ задачи в ИСО/МЭК 12207	Описание задачи	Тип выходного результата задачи
7.4.1. Задачи работы <i>Подготовка процесса обучения</i>		
7.4.1.1	Анализ требований к проекту для определения необходимости обучения персонала, разработка плана обучения	План
7.4.2. Задачи работы <i>Разработка учебных материалов</i>		
7.4.2.1	Разработка руководств и материалов для обучения	Описание
7.4.3. Задачи работы <i>Реализация плана обучения</i>		

№ задачи в ИСО/МЭК 12207	Описание задачи	Тип выходного результата задачи
7.4.3.1	Реализация плана обучения с сохранением протоколов	Протокол
7.4.3.2	Обеспечение своевременной подготовки персонала к выполнению запланированных работ и задач	

Адаптация стандарта к условиям проекта



Процесс адаптации стандарта определен в *Приложении А СТБ ИСО/МЭК 12207-2003*.

Таблица 18 содержит задачи, реализуемые при выполнении соответствующих работ процесса адаптации, и типы выходных результатов данных задач, которые должны быть документально оформлены.

Таблица 18 – Задачи работ процесса адаптации

№ задачи в ИСО/МЭК 12207	Описание задачи	Тип выходного результата задачи
А.1. Задачи работы Определение условий выполнения проекта		
А.1.1	Определение <u>характеристик условий выполнения проекта</u> (ссылка), влияющих на адаптацию	
А.2. Задачи работы Запрос исходных данных		
А.2.1	Запрос и получение исходных данных, влияющих на адаптацию, от участвующих в проекте организаций	
А.3. Задачи работы Выбор процессов, работ и задач		
А.3.1	Определение необходимых процессов, работ и задач с учетом разрабатываемой документации и обязанностей исполнителей	
А.3.2	Установка в договоре дополнительных процессов, работ и задач, не описанных в стандарте СТБ ИСО/МЭК 12207-2003, оценка соответствия организационных процессов	
А.3.3	Отдельный анализ на предмет исключения из ЖЦ обязательных (по стандарту СТБ ИСО/МЭК 12207-2003) задач, с учетом факторов риска, стоимости,	

№ задачи в ИСО/МЭК 12207	Описание задачи	Тип выходного результата задачи
	графика работ, выполнимости, объема, критичности и интерфейса с пользователем	
А.4. Задачи работы Документирование решений по адаптации и их обоснование		
А.4.1	Документальное оформление решений по адаптации с их обоснованием	Отчет

Дополнительный материал. Стандартизация жизненного цикла программного обеспечения

Содержание плана управления проектом:

- планирование организационной структуры проекта, полномочий и обязанностей каждого участника проекта;
- планирование технической среды разработки, эксплуатации и сопровождения (*условия проведения испытаний, оборудование, организация архивной библиотеки, средства, стандарты, процедуры, инструментарий*);
- планирование структуры распределения заданий по процессам и работам жизненного цикла совместно со сметами, составом исполнителей, требуемыми материальными ресурсами, необходимыми программными средствами, графиками выполнения задач;
- планирование управления характеристиками качества создаваемого программного продукта или программной услуги;
- планирование управления безопасностью, защитой и другими критическими требованиями к программному продукту или программной услуге;
- планирование обеспечения защиты, *включая правила доступа к информации на уровне каждой проектной организации*;
- планирование управления критическими ситуациями (*областями проекта, связанными с потенциальными техническими, финансовыми и плановыми затруднениями*);
- планирование управления субподрядчиками;
- планирование обеспечения качества;
- планирование верификации и аттестации;
- планирование взаимоотношений с заказчиками, реализуемых *совместными анализами, аудиторскими проверками, совещаниями, отчетами, модификациями и изменениями программного продукта, сдачей, утверждением, приемкой ПП и договорами*;
- планирование взаимоотношений с пользователями, реализуемых *посредством выполнения требуемых настроек, демонстраций прототипов и оценок*;

- планирование подтверждения статуса поставляемой продукции
(*инструкции, обязательная сертификация, права собственности, использования и распространения, гарантии и лицензионные права*);

- планирование средств для планирования, надзора и отчетности;

- планирование обучения персонала.

Содержимое плана обозначения документов

- заголовок или наименование;

- назначение;

- пользователи документа;

- процедуры и обязанности по подготовке исходных материалов, разработке, проверке, изменению, утверждению, выпуску, хранению, распространению, сопровождению и управлению конфигурацией документов.

Вопросы плана выполнения процессов

- установка графиков выполнения задач;

- оценка необходимых трудозатрат;

- определение ресурсов, необходимых для выполнения задач;

- распределение задач по исполнителям;

- определение обязанностей исполнителей;

- определение критических ситуаций, связанных с задачами или самим процессом;

- установка используемых в процессе критериев управления качеством;

- определение затрат, связанных с реализацией процесса;

- обеспечение условий и определение инфраструктуры выполнения процесса.

Критерии оценки результатов процесса разработки

Объект оценки	Критерии
Архитектура системы и требования к ее объектам (см. задачу 5.3.3.2)	<ul style="list-style-type: none"> – учет требований к системе; – соответствие требованиям к системе; – соответствие используемых стандартов и методов проектирования; – возможность программных объектов архитектуры выполнять установленные для них требования; – возможность эксплуатации и сопровождения
Архитектура программного объекта и эскизных проектов интерфейсов и базы данных(см. задачу 5.3.5.6)	<ul style="list-style-type: none"> – учет требований к программному объекту; – внешняя согласованность с требованиями к программному объекту; – внутренняя согласованность между компонентами объекта; – соответствие методов проектирования и используемых стандартов; – возможность технического проектирования; – возможность эксплуатации и сопровождения.
Технический проект и требования к тестированию(см. задачу 5.3.6.7)	<ul style="list-style-type: none"> – учет требований к программному объекту; – внешнее соответствие спроектированной архитектуре;

Объект оценки	Критерии
	<ul style="list-style-type: none"> – внутренняя согласованность между компонентами программного объекта и программными модулями; – соответствие методов проектирования и используемых стандартов; – возможность тестирования; – возможность эксплуатации и сопровождения.
Результаты программирования и тестирования (см. задачу 5.3.7.5)	<ul style="list-style-type: none"> – учет требований к программному объекту и проекту объекта в целом; – внешнее соответствие требованиям и проекту программного объекта; – внутреннее соответствие между требованиями к программным модулям; – тестовое покрытие всех модулей; – соответствие методов программирования и используемых стандартов; – возможность сборки и тестирования; – возможность эксплуатации и сопровождения.
Планы и результаты сборки и тестирования, а также документация пользователя (см. задачу 5.3.8.5)	<ul style="list-style-type: none"> – учет требований к системе; – внешнее соответствие требованиям к системе; – внутренняя согласованность между программными объектами; – тестовое покрытие требований к программному объекту;

Объект оценки	Критерии
	<ul style="list-style-type: none"> – соответствие используемых стандартов и методов испытаний; – соответствие ожидаемым результатам; – выполнимость квалификационного испытания программного объекта; – возможность эксплуатации и сопровождения.
Результаты испытаний и документация пользователя (см. задачу 5.3.9.3)	<ul style="list-style-type: none"> – тестовое покрытие требований к программному объекту; – соответствие ожидаемым результатам; – возможность сборки и тестирования системы; – возможность эксплуатации и сопровождения.
Собранная систем (см. задачу 5.3.10.3)	<ul style="list-style-type: none"> – тестовое покрытие требований к системе; – соответствие методов тестирования и используемых стандартов; – соответствие ожидаемым результатам; – возможность квалификационных испытаний системы; – возможность эксплуатации и сопровождения.
СИСТЕМА (см. задачу 5.3.11.2)	<ul style="list-style-type: none"> – тестовое покрытие требований к системе; – соответствие ожидаемым результатам; – возможность эксплуатации и сопровождения.

Работы по проектированию программных средств

Отражена структура основных результатов работ процесса разработки, связанных с проектированием ПС, и взаимосвязь данных результатов.

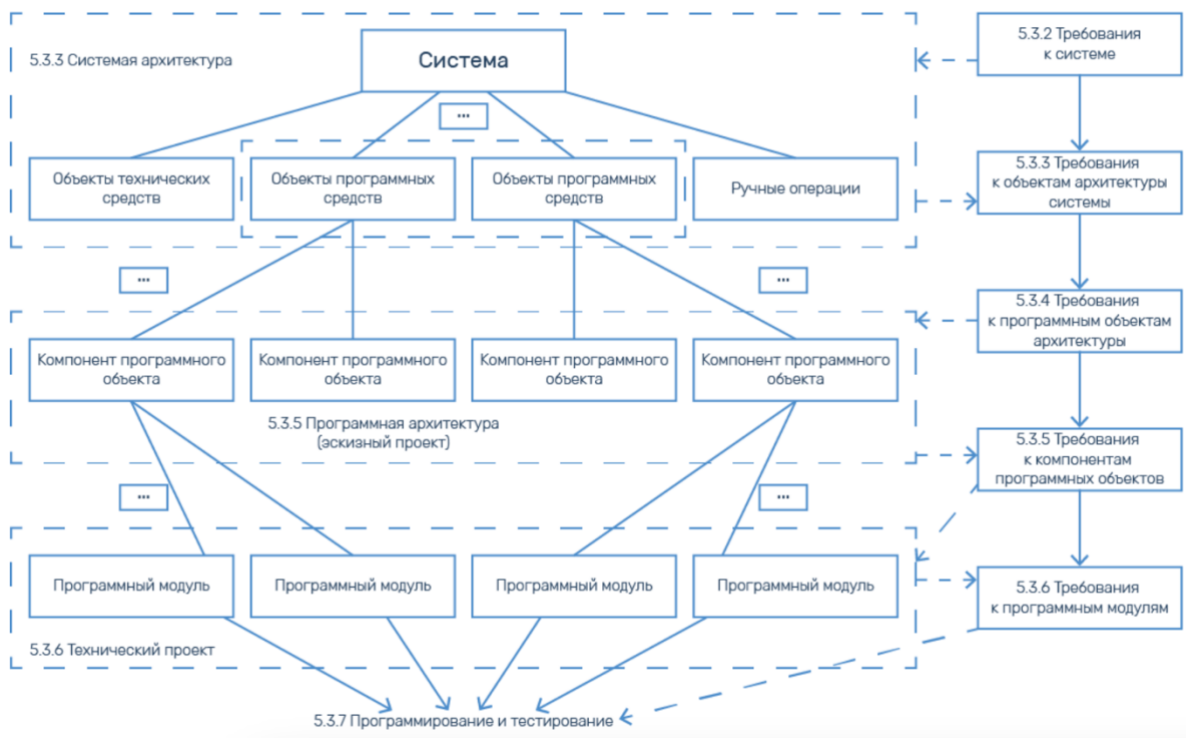


Рисунок 1 – Программирование и тестирование

Критерии верификации

Объект оценки	Критерии
Договор(см. задачу 6.4.2.1)	<ul style="list-style-type: none">– возможность поставщика удовлетворять установленным требованиям;– непротиворечивость требований и охват ими потребностей пользователя;

Объект оценки	Критерии
	<ul style="list-style-type: none"> – наличие соответствующих процедур для внесения изменений в установленные требования и для решения проблем; – наличие процедур по взаимодействию и кооперации между участниками договора; – наличие критериев и процедур, предусмотренных в соответствии с установленными требованиями.
Процесс (см. задачу 6.4.2.2)	<ul style="list-style-type: none"> – соответствие и своевременность установления требований к планированию проекта; – пригодность, реализуемость, выполнимость в соответствии с планом и условиями договора выбранных для проекта процессов; – применимость стандартов, процедур и условий к процессам проекта; – укомплектованность и обученность персонала в соответствии с условиями договора.
Требования (см. задачу 6.4.2.3)	<ul style="list-style-type: none"> – непротиворечивость, выполнимость и тестируемость требований к системе; – распределение требований к системе между объектами технических и программных средств и ручных операций в соответствии с критериями проектирования; – непротиворечивость, выполнимость, тестируемость и точность отражения требований к системе в требованиях к программным средствам; – правильность (подтвержденная соответствующими методами) критических требований к программным средствам, в том числе по безопасности и защите.
Результаты проектирования (см. задачу 6.4.2.4)	<ul style="list-style-type: none"> – правильность, соответствие установленным требованиям и учет этих требований; – реализация соответствующей последовательности событий, исходных данных, выходных результатов, интерфейсов, логики; соответствие временным

Объект оценки	Критерии
	<p>ограничениям и ограничениям размера; обнаружение, локализация и устранение ошибок;</p> <p>– возможность дальнейшего использования с учетом требований;</p> <p>– правильность, подтвержденная соответствующими методами, реализации требований безопасности, защиты и других критических требований.</p>
Исходные тексты программных модулей (см. задачу 6.4.2.5)	<p>– трассируемость с результатами проектирования и требованиями; тестируемость, правильность и соответствие требованиям и стандартам программирования;</p> <p>– реализация соответствующей последовательности событий, соответствующих интерфейсов, правильных данных и логики управления; завершенность; соответствие временным ограничениям и ограничениям размера; обнаружение, локализация и устранение ошибок;</p> <p>– соответствие результатам проектирования и требованиям;</p> <p>– возможность дальнейшего использования с учетом результатов тестирования и требований;</p> <p>– правильность, подтвержденная соответствующими методами, реализации требований безопасности, защиты и других критических требований.</p>
Сборка (см. задачу 6.4.2.6)	<p>– полнота и правильность сборки программных компонентов и модулей каждого программного объекта в соответствующий программный объект;</p> <p>– полнота и правильность сборки технических и программных объектов и ручных операций в систему;</p> <p>– выполнение задач сборки в соответствии с планом сборки.</p>
Документация (см. задачу 6.4.2.7)	<p>– соответствие, полнота и непротиворечивость документации;</p>

Объект оценки	Критерии
	<ul style="list-style-type: none"> – своевременность подготовки документации; – соблюдение установленных процедур управления конфигурацией документов.

Критерии проведения аудиторской проверки

- соответствие запрограммированных программных продуктов (объектов) проектной документации;
- пригодность подготовки приемки и требований к тестированию, установленных в документации, для приемки программных продуктов (объектов);
- соответствие тестовых данных установленным техническим требованиям;
- успешность тестирования программных продуктов и их соответствие установленным к ним требованиям;
- правильность отчетов об испытаниях (тестировании), устранение расхождений между фактическими и ожидаемыми результатами;
- соответствие документации пользователя установленным стандартам;
- выполнение работ в соответствии с утвержденными требованиями, планами и договором;
- соответствие стоимости и графика проведения работ утвержденным планам.

Характеристики условий выполнения проекта

- модель жизненного цикла;
- влияние жизненного цикла существующей системы;
- требования к системе и программным средствам;
- организационные подходы, процедуры и цели;

- размер, сложность, критичность и типы системы, программного продукта или программной услуги;
- методы разработки;
- количество задействованного персонала и участвующих в проекте сторон;
- методы и политика заказа.

Метрология качества программных средств и систем

1. Общие понятия о метрике

Атрибуты программной системы, характеризующие ее качество, измеряются с использованием метрик качества.

Метрика

Комбинация конкретного метода измерения (способа получения значений) атрибута сущности и шкалы измерения (средства, используемого для структурирования получаемых значений).

Метрика определяет меру атрибута – переменную, которой присваивается значение в результате измерения (рисунок 1).

Термин **«мера»** – эквивалент употребляемого в международных стандартах термина «measure» (– *мера, измерять, единица измерения*).



Рисунок 1 – Метрика в системе измерения качества

По определению стандарта СТБ ИСО/МЭК 9126 **метрика качества программной системы** представляет собой «*модель измерения атрибута, связываемого с характеристикой качества ПС. Служит индикатором одного или многих атрибутов*».

Метрика называется **базовой**, если в ее основе лежит элементарный метод (примитив) измерения атрибута. По определению того же стандарта «*базовая метрика сама по себе не является индикатором характеристики или подхарактеристики качества*». То есть, базовые метрики используются только в составе модели измерения атрибута.

Для того чтобы правильно пользоваться результатами измерений, для каждой меры нужно установить **шкалу измерения**.

Принято использовать одну из следующих шкал: **номинальная, порядковая, интервальная, относительная** или **абсолютная**.

Для разработки процедур сбора данных, интерпретации мер и их нормализации с целью сравнения, нужно различать следующие типы мер, определяемых метриками:

- Меры размера.

Представляют размер ПС в разных единицах измерения.

- Меры времени.

Представляют периоды реального времени (в секундах, минутах или часах), процессорного времени (в секундах, минутах или часах работы процессора) или календарного времени (в рабочих часах, календарных днях, месяцах, годах).

- Меры усилий.

Представляют полезное (продуктивное) время, связанное с определенной задачей проекта.

- Меры интервалов между событиями.

Представляют интервалы времени между наступлением событий, происходящих в определенный период наблюдения. Вместо этой меры может использоваться частота наступления событий.

- Счетные меры.

Представляют собой статические счетчики (для учета определенных элементов в рабочих продуктах или документах) или кинетические (динамические) счетчики (для учета событий или действий человека).

При выполнении измерений базовые меры размера, времени и счета могут использоваться в различных комбинациях. Они служат основой для нормализации и обеспечивают возможность сопоставления метрик.

В исследовании метрик программных средств принято выделять 2 основных направления:

- поиск набора метрик, которые характеризуют наиболее специфические свойства программного обеспечения;
- использование метрик для оценки технических характеристик и факторов разработки программ.

Для удобства применения общих приемов измерений метрики обычно классифицируют как:

- объективные или субъективные;
- примитивные или вычисляемые;
- динамические или статические.

По виду информации, которую можно получить при оценке качества программного обеспечения, метрики можно разбить на 3 группы:

- **Прогнозирующие** – метрики, позволяющие прогнозировать качество разрабатываемого программного средства.
- **Диагностические** – метрики, оценивающие отклонение от нормы характеристик исходных проектных материалов.
- **Ретроспективные** – метрики, по которым принимается решение о соответствии конечного программного средства заданным требованиям.

По отношению к виду объекта измерения (работающая программа или совокупность документов) меры и соответствующие метрики подразделяются на **внешние, внутренние и метрики использования ПС**.

Внешние метрики используют меры работающего на компьютере программного продукта, полученные в результате измерения его поведения в ходе тестирования и функционирования.

Внешние метрики разрабатываются с целью:

- демонстрации качества программного продукта, представленного характеристиками и подхарактеристиками качества, на стадии тестирования и эксплуатации;
- использования для подтверждения (валидации) того, что программный продукт удовлетворяет внешним требованиям к качеству;
- предсказания реального эксплуатационного качества;
- определения степени, в которой программный продукт будет удовлетворять установленным и предполагаемым требованиям пользователей в ходе реальной эксплуатации.

Можно сказать, что совокупность внешних метрик предназначена для оценивания **внешнего качества** – *степени, в которой продукт удовлетворяет установленным (заявленным) и подразумеваемым потребностям при использовании в определенных условиях.*

Разработка внешних метрик основывается на выполнении следующих измерений:

- поведения программного продукта при тестировании и функционировании в сочетании с другими программными продуктами, аппаратным обеспечением или системой обработки информации в целом;
- поведения пользователя (сценариев использования ПС).

Под измерением (оценкой) поведения понимается оценка масштабов возможных последствий неадекватного поведения, которые угрожают жизни или здоровью людей, природным ресурсам, могут привести к разрушению данных, несогласованности или недостоверности информации, потере безопасности, деградации сервиса (услуг), экономическим потерям и др. Примерами внешних метрик для такой характеристики качества как надежность, могут быть среднее время между отказами, число устраненных дефектов при тестировании, интенсивность отказов и др.

Внутренние метрики обеспечивают возможность пользователям, разработчикам, тестировщикам и менеджерам оценивать качество промежуточных и конечных продуктов ПС непосредственно по их свойствам, без выполнения на компьютере.

Внутренние метрики разрабатываются таким образом, чтобы они могли:

- представлять (отражать) качество не выполняющихся на компьютере промежуточных и конечных программных продуктов по тем характеристикам и подхарактеристикам качества, которые определены в модели качества ПС;
- служить руководством к действию при планировании и улучшении процессов, которые воздействуют на промежуточные и конечные продукты;
- использоваться при верификации того, что промежуточные и конечные продукты удовлетворяют требованиям к внутреннему качеству ПС, предусмотренным планами совершенствования процессов;
- предсказывать внешние метрики качества.

Можно сказать, что совокупность внутренних метрик предназначена для оценивания **внутреннего качества** – множества атрибутов продукта, которое определяет его способность удовлетворять установленным или реальным потребностям при использовании в определенных условиях.

Разработка внутренних метрик основывается на выполнении измерений статических атрибутов, которые определены и могут быть оценены по тексту исходного кода, графическому или табличному представлению потоков управления и данных, структур перехода состояний или по документам ПС. Примерами внутренних метрик для надежности могут быть число ошибок, найденных при инспекции кода, число устраненных дефектов в результате инспекции кода, прогнозируемое число оставшихся ошибок и др.

Метрики качества в использовании (метрики эксплуатационного качества) измеряют степень, в которой программный продукт, установленный и эксплуатируемый в определенной среде, удовлетворяет потребности пользователей в эффективном, продуктивном и безопасном решении задач.

Метрики качества в использовании помогают оценить не свойства самой ПС, а видимые результаты ее эксплуатации – *эксплуатационное качество*.

Очевидно, что для правильного измерения эксплуатационного качества важно учитывать контекст применения ПС – особенности категорий ее пользователей, специфику решаемых ими задач, а также физические и социальные факторы среды их работы. Примерами эксплуатационных метрик качества могут быть точность и полнота достижения целей пользователей, производительность труда, ресурсы, потраченные в связи с достижением эффективного решения задач, мнение пользователей относительно свойств ПС и др.

Внутренние, внешние и эксплуатационные метрики качества взаимосвязаны.

Достижение эксплуатационного качества зависит от удовлетворения критериев внешнего качества, основанных на внешних мерах и метриках качества, которые, в свою очередь, зависят от удовлетворения соответствующих критериев внутреннего качества, основанных на внутренних мерах и метриках, связанных с внешними. Обычно требования пользователей к качеству специфицируются с помощью внешних метрик и эксплуатационных метрик качества, а внутренние метрики выбираются таким образом, чтобы они могли использоваться для предсказания значений внешних метрик. Построить строгую теоретическую модель, устанавливающую взаимосвязь внешних и внутренних метрик, сложно, поэтому, как правило, строится гипотетическая модель, взаимосвязь метрик в которой моделируется статистически в ходе использования метрик.

2. Методы определения показателей качества программного средства

Межгосударственный стандарт СНГ **ГОСТ 28195–99** классифицируют методы определения показателей качества ПС по следующим критериям:

- **по способам получения информации** о показателе качества:
 - измерительный;
 - регистрационный;
 - органолептический;
 - расчетный;
- **по источникам получения информации** о показателе качества:
 - экспертный;
 - социологический;

– традиционный.

Измерительный метод – это метод получения информации о свойствах и характеристиках ПС *путем измерений с помощью инструментальных средств*.

Например, так может определяться количество операторов в программе, количество выполненных операторов, количество операций или операндов, время выполнения программы при определенных наборах исходных данных и т.п.

Результаты непосредственных измерений при необходимости приводятся путем соответствующих пересчетов к нормализованным или стандартным условиям.

Регистрационный метод – это метод получения информации о свойствах и характеристиках ПС в процессе его *испытания или функционирования*, когда регистрируются некоторые события.

Например, количество сбоев и отказов при запусках ПС.

Органолептический метод – это метод получения информации о свойствах и характеристиках ПС, основанный *на восприятии органов чувств* (зрения и слуха) человека.

Например, так могут определяться свойства ПС, связанные с удобством его использования.

Точность метода зависит от квалификации, опыта и способностей лиц, проводящих оценку. При органолептическом методе могут дополнительно использоваться технические средства, повышающие способности органов чувств (лупа, микроскоп, слуховая трубка и т.п.).

При органолептическом методе рекомендуется использовать балльные оценки показателей качества. При такой оценке вначале необходимо установить перечень признаков, наиболее полно характеризующих качество продукта. Обычно применяется совместно с экспертным.

Расчетный метод – это метод получения информации о свойствах и характеристиках ПС, основанный на *использовании эмпирических и теоретических зависимостей* (на ранних этапах разработки), а также *статистических данных*, накапливаемых при испытаниях, эксплуатации и сопровождении ПС.

Например, так может определяться точность вычислений в программе.

Традиционный метод – это метод получения информации о свойствах и характеристиках ПС *на основе непосредственного наблюдения* за их функционированием в процессе работы.

Традиционный метод осуществляется лицами специализированных экспериментальных и расчетных подразделений предприятий, учреждений.

Например, этим методом можно определять некоторые из показателей функциональности и удобства использования.

Социологический метод – это метод получения информации о свойствах и характеристиках ПС *на основе обработки специальных анкет-опросников*.

Социологический метод определения показателей качества продукции используется фактическими или потенциальными потребителями продукции. Для применения метода необходимо разработать системы опроса и обработки результатов.

Таким методом, например, можно определять отдельные показатели удобства использования.

Экспертный метод – это метод получения информации о свойствах и характеристиках ПС *на основании мнений группы экспертов* (специалистов, компетентных в решении данной задачи).

Экспертный метод применяется в том случае, когда невозможно или слишком трудоемко выполнить оценку показателей качества с помощью других методов. Для исключения необъективных оценок в состав экспертной комиссии не должны входить авторы изделия. Чтобы уменьшить степень субъективности оценки, целесообразно провести несколько туров опроса экспертов с публичным обоснованием выставленных оценок.

В основном этот метод используется при определении значений некоторых эргономических и эстетических показателей. Часто данный метод применяется в сочетании с органолептическим методом для принятия решения при аттестации качества продукции.

Данным методом рекомендуется определять, например, показатели понимаемости и осваиваемости ПС.

Существуют различные формы проведения экспертных оценок. Одна из них — **«метод Дельфы»**, в основе которого введение специальных мер, чтобы исключить влияние на конечный результат экспертов, обладающих даром убеждать других. Для этого исключают личные контакты между экспертами и применяют итерационную процедуру, в которой анонимное мнение каждого эксперта

подвергается критике со стороны всех остальных экспертов при сохранении анонимности мнений.

Методы оценки уровня качества продукции одного вида могут быть: *дифференциальным, комплексным, смешанным*.

Дифференциальный метод оценки уровня качества продукции осуществляется сравнением показателей качества оцениваемого вида продукции с соответствующим базовым показателем. То есть показатели качества ($P_1...P_n$) сопоставляются с показателями качества базового образца ($P_{1_base}...P_{n_base}$) где n – число оцениваемых параметров.

Затем для каждого из показателей рассчитываются относительные показатели качества оцениваемой продукции – как отношение реального показателя к базовому, или наоборот.

Значения показателей качества продукции при дифференциальном методе оценки не должны выходить за допустимые пределы. Нижний предел, как правило, диктуется технико-эксплуатационными требованиями, верхний — экономической целесообразностью.

Если количество показателей качества велико, оценить уровень качества дифференциальным методом может быть затруднительно. В этом случае целесообразно производить оценку другим методом, например, комплексным.

Комплексный метод оценки уровня качества предусматривает использование комплексного (обобщенного) показателя качества. Данный метод применяется в случаях, когда оказывается целесообразным выразить уровень качества только одним числом. Уровень качества, согласно комплексному методу, определяется отношением обобщенного показателя качества оцениваемой продукции Q к обобщенному показателю базового образца Q_{base} .

Сложность комплексной оценки заключается в объективном нахождении обобщенного показателя, который может быть выражен:

- главным показателем, отражающим основное назначение продукции;
- интегральным показателем качества продукции;
- средневзвешенным показателем.

Дифференциальный и комплексный методы имеют свои недостатки. В отношении сложной продукции при помощи дифференциального метода трудно сделать конкретный вывод, а использование только комплексного метода не позволяет объективно учесть все значимые свойства оцениваемой продукции. В этом случае используются одновременно и комплексный и дифференциальный методы, то есть оценку производят **смешанным методом**.

Метод предполагает следующую последовательность:

1. Единичные показатели качества объединяются в ряд групп (с учетом цели оценки), для которых определяют групповой комплексный показатель качества. Наиболее значимые единичные показатели можно в группы не включать, а рассматривать отдельно. Объединение показателей в группы должно производиться исходя из целей оценки. *Единичные показатели качества зачастую определяются с помощью измерений.*

2. Найденные величины групповых комплексных и отдельно выделенных наиболее важных единичных показателей сравнивают с соответствующими значениями базовых показателей, т. е. применяют дифференциальный метод.

3. Применение метрик качества ПС

Применение конкретного вида метрик определяется стадией жизненного цикла программного средства.

Однако,

все **метрики должны поддерживать следующие свойства:**

1. Надежность;

предполагается, что метрика свободна от случайной ошибки, если случайные изменения не влияют на результаты метрики.

2. Повторяемость;

повторное использование метрики для того же продукта теми же специалистами по оценке, используя ту же спецификацию оценки (включая ту же окружающую среду), тот же тип пользователей и окружения, должно привести к тем же результатам с соответствующими допусками; соответствующие допуски должны учитывать такие компоненты, как усталость и результат накопленных познаний.

3. Однотипность;

применение метрики для того же продукта различными специалистами по оценке, используя ту же спецификацию оценки (включая ту же окружающую среду), тот же тип пользователей и окружения, должно привести к тем же результатам с соответствующими допусками.

4. Применимость;

метрика должна четко указывать условия (например, наличие определенных атрибутов), которые ограничивают её употребление.

5. Показательность;

метрика должна быть способной идентифицировать части или элементы программы, которые должны быть улучшены, на основании сравнения измеренных и ожидаемых результатов.

6. Корректность;

- **объективность;**

результаты метрики и её входные данные должны быть основаны на фактах и не подвластны чувствам или мнениям специалистов по оценке или тестированию;

- **беспристрастность;**

измерение не должно быть направлено на получение какого-либо специфического результата;

- **адекватность точности;**

точность определяется при проектировании метрики и особенно при выборе описаний фактов, используемых как основа для метрики. необходимо описать точность и чувствительность метрики.

7. Значимость;

измерение должно давать значащие результаты, касающиеся поведения программы или характеристик качества.

Метрика должна также быть эффективной по отношению к стоимости. Это значит, что более дорогие метрики должны обеспечивать лучшие результаты оценки.

Кроме того, для применения в оценке качества,

метрика должна удовлетворять хотя бы одному из следующих **критериев обоснованности метрик:**

1. Кореляция;

изменение в значениях характеристик качества (оперативно определенных по результатам измерения основных метрик), обусловленное изменением в значениях метрики, должно определяться линейной зависимостью;

2. Трассировка;

если метрика M непосредственно связана с величиной характеристики качества Q (оперативно определенной по результатам измерения основных метрик), то изменение величины $Q(T_1)$, имеющейся в момент времени T_1 , к величине $Q(T_2)$ полученной в момент времени T_2 , должно сопровождаться изменением значения метрики от $M(T_1)$ до $M(T_2)$ в том же направлении (например, если увеличивается Q , то M тоже увеличивается);

$$Q(t_1) \rightarrow Q(t_2) \Rightarrow M(t_1) \rightarrow M(t_2) \quad (1)$$

3. Непротиворечивость;

если значения характеристик качества (оперативно полученные по результатам измерения основных метрик) Q_1, Q_2, \dots, Q_n , связанные с продуктами или процессами $1, 2, \dots, n$, определяются соотношением $Q_1 > Q_2 > \dots > Q_n$, то соответствующие значения метрики должны удовлетворять соотношению $M_1 > M_2 > \dots > M_n$.

$$Q_1 > Q_2 > \dots > Q_n \Rightarrow M_1 > M_2 > \dots > M_n \quad (2)$$

4. Предсказуемость;

если метрика используется в момент времени T_1 для прогноза значения (оперативно полученного по результатам измерения основных метрик) характеристики качества Q в момент времени T_2 , то ошибка прогнозирования, определяемая выражением (3), должна попадать в допустимый диапазон ошибок прогнозирования;

$$(A - B) / B \quad (3)$$

где

A – прогнозное $Q(T_2)$, B – фактическое $Q(T_2)$.

5. Селективность;

метрика должна быть способной различать высокое и низкое качество программного средства.

В стандарте СТБ ИСО/МЭК 9126 и отчетах международного стандарта ISO/IEC TR 9126 2–4:2004 для каждой подхарактеристики внешнего и внутреннего качества и характеристики качества в использовании приведены таблицы, в которых даны примеры метрик качества.

Таблицы имеют следующую структуру:

1. название метрики;
 2. назначение метрики (*изложено в виде вопроса, на который отвечает применение метрики*);
 3. метод применения;
 4. способ измерения, формула, исходные и вычисляемые данные;
 5. интерпретация измеренного значения (*диапазон и предпочтительные значения*);
 6. тип шкалы, используемой при измерении метрики;
 7. тип измеренного значения;
- используются следующие типы измеренных значений:
- тип размера (*например, функциональный размер, размер исходного текста*);
 - тип времени (*например, затраченное время, необходимое пользователю время*);
 - тип количества (*например, количество изменений, количество отказов*);
8. источники входных данных для измерения;
 9. ссылка на ISO/IEC 12207:1995 (*процессы жизненного цикла ПС, при выполнении которых применима метрика*);
 10. целевая аудитория (*под аудиторией понимается категория пользователей, предъявляющих к некоторой документации одинаковые или аналогичные требования, определяющие содержание, структуру и назначение данной документации*).

Для обеспечения возможности совместного использования различных метрик (независимо от их физического смысла, единиц измерения и диапазонов значений)

при количественной оценке качества программных продуктов метрики в стандартах по возможности представляются в относительных единицах в виде двух формул:

$$X=A/B \quad (4)$$

или

$$X=1-A/B \quad (5)$$

где

X – значение метрики;

A – абсолютное (измеренное) значение некоторого свойства (атрибута) оцениваемого продукта или документации;

B – базовое значение соответствующего свойства.

Из двух вышеназванных формул (4 и 5) для конкретной метрики выбирается та, которая соответствует критериям трассировки и непротиворечивости: с увеличением относительного значения метрики значение подхарактеристики и характеристики качества должно увеличиваться.

Вычисление метрик по формуле (4) или (5) позволяет привести их относительные значения в диапазон (6), что упрощает их совместное использование при интегральной оценке качества программных средств.

$$0 \leq X \leq 1 \quad (6)$$

4. Метрики качества программных средств и систем по стандарту ISO/IEC 9126

В стандарте ISO/IEC 9126 есть рекомендации для **внутренних** и **внешних** метрик качества программных средств, а также метрик качества программных средств **в использовании** (ISO/IEC TR 9126 2–4 : 2004).

Следует отметить, что не все метрики, приведенные в таблицах отчета стандарта ISO/IEC TR 9126, удовлетворяют таким критериям обоснованности метрик, как корреляция, трассировка и непротиворечивость. Это затрудняет их использование при интегральной оценке качества программных продуктов.

Внутренние метрики качества программных средств

- **Внутренние метрики функциональности** предназначены для предсказания того, удовлетворяет ли разрабатываемый программный продукт требованиям к функциональности и предполагаемым потребностям пользователя.
- **Внутренние метрики надежности** используются во время разработки программного продукта для предсказания того, удовлетворяет ли ПП заявленным потребностям в надежности.
- **Внутренние метрики практичности** используются во время разработки программного продукта для предсказания степени, в которой ПП может быть понят, изучен, управляем, привлекателен и соответствует договоренностям и руководствам по практичности.
- **Внутренние метрики эффективности** используются во время разработки программного продукта для предсказания эффективности поведения ПП во время тестирования или эксплуатации.
- **Внутренние метрики сопровождаемости** используются для предсказания уровня усилий, необходимых для модификации программного продукта.
- **Внутренние метрики мобильности** используются для предсказания воздействия программного продукта на поведение исполнителя или системы при проведении работ по переносу.

Внешние метрики качества программных средств

- **Внешние метрики функциональности** должны измерять свойства (атрибуты) функционального поведения системы, содержащей ПС.
- **Внешние метрики надежности** должны измерять свойства, связанные с поведением системы, содержащей ПС, во время тестирования, чтобы показать степень надежности ПС в системе в процессе эксплуатации.
- **Внешние метрики практичности** показывают, в какой мере программное средство может быть понято, изучено, управляемо, привлекательно и соответствует договоренностям и руководствам по практичности.
- **Внешние метрики эффективности** должны измерять такие атрибуты, как характер изменения затрат времени и использования ресурсов компьютерной системы, включающей ПС, во время тестирования или эксплуатации.
- **Внешние метрики сопровождаемости** измеряют такие атрибуты, как поведение персонала сопровождения, пользователя или системы, включающей ПС, при модификации ПС во время тестирования или сопровождения.
- **Внешние метрики мобильности** измеряют такие атрибуты, как поведение оператора или системы при проведении работ по переносу.

Метрики качества программных средств в использовании

- **Метрики результативности** оценивают, достигают ли задачи, выполняемые пользователем, заданных целей с точностью и полнотой в заданном контексте использования.
- **Метрики продуктивности** оценивают ресурсы, которые затрачивают пользователи в соответствии с достигнутой результативностью в заданном контексте использования.
- **Метрики безопасности** оценивают уровень риска причинения вреда людям, бизнесу, программному обеспечению, имуществу или окружающей среде в заданном контексте использования.
- **Метрики удовлетворенности** оценивают отношение пользователя к использованию продукта в заданном контексте использования.

Метрики оценки сложности программных средств и систем

Метрики оценки программного кода

Применение метрик программного кода позволяет разработчикам и другим участникам жизненного цикла программного средства оценивать различные свойства создаваемого или существующего программного обеспечения, прогнозировать, давать количественную характеристику тех или иных проектных решений, оценивать качество разработанных систем и их частей, характеризовать сложность или надежность программного обеспечения.

В настоящее время в мировой практике используется несколько сотен метрик программ.

Существующие качественные оценки программ можно сгруппировать по 6 направлениям

-
1. оценки топологической и информационной сложности программ;

2. оценки надежности программных систем, позволяющие прогнозировать отказовые ситуации;
3. оценки производительности ПО и повышения его эффективности путем выявления ошибок проектирования;
4. оценки уровня языковых средств и их применения;
5. оценки трудности восприятия и понимания программных текстов, ориентированные на психологические факторы, существенные для сопровождения и модификации программ;
6. оценки производительности труда программистов для прогнозирования сроков разработки программ и планирования работ по созданию программных комплексов.

Метрики оценки сложности программ разбивают на 3 основные группы

1. метрики размера программ
2. метрики сложности потока управления программ
3. метрики сложности потока данных программ

Основная цель метрик сложности – выявить наиболее критичные участки программного проекта, которые являются потенциальными источниками ошибок и повышенных рисков на всех стадиях его жизненного цикла.

Метрики первой группы (количественная оценка) базируются на определении количественных характеристик, связанных с размером программы, и ориентированы на анализ исходного текста программ. Могут использоваться для оценки сложности промежуточных продуктов разработки.

Метрики второй группы (оценка сложности потока управления программ) базируются на анализе управляющего графа программы, поэтому эти метрики могут применяться для оценки сложности промежуточных продуктов разработки. Эти метрики используются главным образом для апостериорного анализа, однако могут применяться и на ранних стадиях работы при осуществлении проектирования.

Как правило, с помощью этих оценок оперируют либо плотностью управляющих переходов внутри программ, либо взаимосвязями этих переходов. И в том, и в другом случае стало традиционным представление программ в виде управляющего ориентированного графа $G=(V,E)$, где V – вершины,

соответствующие операторам, а E – дуги, соответствующие переходам от оператора к оператору.

Структурная сложность программ определяется:

- числом взаимодействующих компонент;
- числом связей между компонентами;
- сложностью взаимодействия компонент.

При функционировании программы разнообразие ее поведения и разнообразие связей между ее входными и результирующими данными в значительной степени определяется маршрутами (чередующихся последовательностей вершин и дуг графа управления), по которым выполняется программа. Для повышения качества программы маршруты возможной обработки данных должны быть тщательно проверены при создании программы.

УСТАНОВЛЕНО, что *сложность программного модуля связана не столько с размером (числом команд) программы, сколько с числом маршрутов ее исполнения и их сложностью.*

Метрика "SLOC"

Традиционной характеристикой размера программ является количество строк кода. Непосредственное измерение размера программы дает хорошие результаты для классификации программ, существенно различающихся объемами. НО следует помнить, что оценка размера программы **недостаточна для принятия решения о ее сложности.**

В зависимости от того, каким образом учитывается исходный код, выделяют два основных показателя:

- ОЦЕНКА КОЛИЧЕСТВА «ФИЗИЧЕСКИХ» СТРОК КОДА;

LOC (Lines of Code) – определяется как общее число строк исходного кода, включая комментарии и пустые строки.

SLOC (Source Lines of Code) – определяется как число строк исходного кода (без учета строк комментариев и пустых строк).

- ОЦЕНКА КОЛИЧЕСТВА «ЛОГИЧЕСКИХ» СТРОК КОДА

LSI (Lines of Source Instructions) – определяется как количество команд языка в коде программы.

DSI (Delivered of Source Instructions) – количество команд языка, вошедших в конечную поставку программы.

Оценки *LOC*(и производные) **не зависят от языка программирования!**

Оценки *LSI* (и производные) – **зависимы от используемого языка программирования.**

В том случае, если язык не допускает размещение нескольких команд на одной строке, то количество «логических» строк *LSI* будет соответствовать числу «физических» *SLOC*.

Выделяют 2 основных подхода к оценке числа строк кода, которые обеспечивают приемлемый результат:

- **оценка показателя *SLOC* по аналогии.**

Данная методика позволяет получать оценку показателя для всего разрабатываемого проекта или отдельных его составляющих посредством сравнения функциональных свойств с существующими аналогами. При оценке показателя по данной методике подбирается близкий по функциональности аналог с известным значением *SLOC*, далее на его основании определяется прогнозная оценка с учетом различий в функциональности, применяемых языках, возможности оптимизации на основе кода аналога и пр.;

- **оценка показателя *SLOC* «снизу-вверх» экспертным методом.**

Этой методикой предусмотрено разбиение проекта на иерархическую структуру задач (**Work Breakdown Structure, WBS**), на основании которых производится экспертная оценка показателя, в итоге суммируемая для всего проекта. Как правило, экспертами предоставляются три оценки (наиболее вероятная, максимальная и минимальная), а далее они усредняются с помощью бета-распределения (рассчитывается сумма минимальной, максимальной и четырехкратной наиболее вероятной оценок, которая затем делится на шесть).

Для метрики *SLOC* имеется много производных, призванных получить отдельные показатели проекта.

Основными среди них являются:

- число пустых строк (Blank Lines Of Code), *BLOC*;
- число строк, содержащих комментарии (Comment Lines Of Code), *CLOC*;

- число строк, содержащих исходный код и комментарии (Lines with Both Code and Comments), (C&SLOC);
- число строк, содержащих декларативный исходный код;
- число строк, содержащих императивный исходный код;
- среднее число строк для функций (методов);
- среднее число строк, содержащих исходный код для функций (методов);
- среднее число строк для модулей;
- среднее число строк для классов.

Нередко показатель *SLOC* используется на ранних стадиях работы над проектом с целью получения предварительных оценок ее общего объема.

При уменьшении различий в объеме программ на первый план выдвигаются оценки других факторов, оказывающих влияние на сложность. Таким образом, оценка размера программы есть оценка по номинальной шкале, на основе которой определяются только категории программ без уточнения оценки для каждой категории.

Метрики группы *LOC* не отражает трудоемкости по созданию программы.

Также наряду со *SLOC* применяются другие показатели, отражающие количественную оценку отдельных составляющих проекта: **число модулей, функций/методов, классов, страниц документации и пр.**

Метрика Холстеда

К группе оценок размера программ можно отнести также **метрику Холстеда**. За базу в метрике принят подсчет количества операторов и операндов, используемых в программе, т.е. определение размера программы по ее составляющим.

Основу метрики составляют 4 измеряемых характеристики программы:

n_1 – число уникальных операторов программы, включая символы-разделители, имена процедур и знаки операций (словарь операторов);

n_2 – число уникальных операндов программы (словарь операндов);

N_1 – общее число операторов в программе;

N_2 – общее число операндов в программе.

$$N_1 = \sum_{j=1}^{n_1} f_{1j}, N_2 = \sum_{i=1}^{n_2} f_{2i}$$

где f_{1j} – число вхождений j -го оператора, $j=1,2,\dots,n_1$; f_{2i} – число вхождений i -го операнда, $i=1,2,\dots,n_2$.

При определении этих характеристик программы принимаются во внимание следующие моменты:

- Операнды программы представляют собой используемые в ней переменные и константы, которые участвуют в операторах языка.

- Под операторами программы в метрике Холстеда подразумеваются входящие в ее состав: имена функций; имена операций; скобки, которые не являются составной частью других операторов; условные и безусловные переходы и другие известные операторы языка.

- Несколько служебных слов, входящих в состав одного оператора, считаются одним оператором (например, If-Else или Do-While).

- Директивы не учитываются.

Очевидно, что **совокупность операторов программы и их количество зависят от языка программирования**, на котором написана программа.

Величины количества операторов и операндов независимы и могут принимать произвольные значения. Однако между величинами N_1 и N_2 можно установить приблизительное соответствие, причем оно будет взаимно однозначным. Можно утверждать, что $N_1 \approx N_2$, хотя величины словарей n_1 и n_2 могут сильно отличаться друг от друга.

Если расчетные значения длины программы и длины реализации отличаются более чем на 10 %, то это свидетельствует о возможном наличии в программе несовершенств (неоднозначные операторы или операнды, синонимичные операнды, лишние присваивания и т.п.). Длина программы представляет собой математическое ожидание количества слов в тексте программы при фиксированном словаре.

Опираясь на эти характеристики, получаемые непосредственно при статическом анализе исходных текстов программ, рассчитываются 3 базовые оценки размера программы:

- словарь программы

$$n = n_1 + n_2 \text{ (ед.)}, (1)$$

- длину программы

$$N=N_1+N_2 \text{ (ед.)}, (2)$$

- объем программы (число битов, т.е. логических единиц информации, необходимых для записи программы)

$$V=N \cdot \log_2(n) \text{ (бит)}. (3)$$

Через эти базовые оценки в свою очередь различными формулами определяются: уровень качества программирования, сложность понимания программы, трудоемкость кодирования программы, информационное содержание программы и другие оценки программного кода.

Сложность программы по метрике Холстеда рассчитывается по формуле:

$$H_{diff}=(n_1/2) \cdot (N_2/n_2)$$

И в отличие от *LOC*, данная метрика позволяет вычислить трудоемкость разработки:

$$E=H_{diff} \cdot V=(n_1 \cdot N_2 \cdot N \cdot \log_2(n))/(2 \cdot n_2)$$

ABC-метрика (Fitzpatrick)

Предложил метрику Джерри Фицпатрик / Jerry Fitzpatrick, поэтому она известна еще как метрика Фитцпатрика. Метрику было предложено как исправление недостатков метрики *LOC* и метрик на основе токенов (например, таких как метрика Холстеда). *ABC*-метрика может быть применена для оценки размеров и сложности фрагментов анализируемого приложения, а также для автоматического поиска (по нулевому значению данной меры) тривиальных функций – заглушек или частей запутанного кода.

Основана на подсчете 3 характеристики:

- **Assignment** – присваиваний значений переменным,
- **Branch** – явных передач управления за пределы области видимости, т.е. вызовов функций,
- **Condition** – логических проверок.

Мера записывается тройкой значений $\langle A, B, C \rangle$

Но для оценки сложности программы зачастую вычисляется скалярное значение. Например, модуль вектора ABC или взвешенная сумма его компонентов. Но чаще рассчитывается как квадратный корень из суммы квадратов показателей A , B и C .

Эта метрика легко вычисляется, может быть вычислена для разных фрагментов кода и наглядна при визуализации (вектор в трехмерном пространстве). К числу ее недостатков относят тот факт, что она может иметь нулевое значение для некоторых непустых программных единиц.

Метрика Маккейба

Одна из самых распространенных метрик сложности потока управления программ – **цикломатическая сложность**, предложена Томасом МакКейбом (Thomas McCabe). Данная метрика предназначена для оценивания сложности потока управления программы (*control flow graph*).

Формула вычисления цикломатической сложности выглядит следующим образом:

$$Z(G)=l-v+2\cdot p, (1)$$

где

l – число ребер (дуг),

v – число узлов (вершин),

p – число компонентов связности графа.

Число компонентов связности графа можно рассматривать как количество дуг, которые необходимо добавить для преобразования графа в сильносвязный. **Сильносвязным** называется граф, любые две вершины которого взаимно достижимы. Для графов корректных программ, т.е. графов, не имеющих недостижимых от точек входа участков и «висячих» входа и выхода, сильносвязный граф, как правило, получается путем замыкания одной вершины, обозначающей конец программы на вершину, обозначающую точку входа в эту программу.

Цикломатическая сложность определяет число линейно независимых контуров в сильносвязном графе. Иначе говоря, цикломатическое число Маккейба показывает требуемое число проходов для покрытия всех контуров сильносвязанного графа или количество тестовых прогонов программы, необходимых для исчерпывающего тестирования по критерию «работает каждая ветвь».

Метрика цикломатической сложности может быть рассчитана для модуля, метода и других структурных единиц программы.

Для метрики цикломатической сложности существует множество вариаций, в частности:

- **«модифицированная» цикломатическая сложность** – рассматривает не каждое ветвление оператора множественного выбора (switch, case), а весь оператор как единое целое;
- **«строгая» цикломатическая сложность** – включает логические операторы;
- **«упрощенная» цикломатическая сложность** – предусматривает вычисление на основе не графа, а подсчета управляющих операторов;
- **«актуальная» сложность** – определяется как число независимых путей, которые проходит программа при тестировании;
- **метрика сложности глобальных данных** – вычисляется как цикломатическая сложность модуля и увеличивается на количество взаимосвязей с глобальными данными.

Метрика Джилба

Одной из наиболее простых, но, как показывает практика, достаточно эффективных оценок сложности программ является метрика Джилба, в которой логическая сложность программы определяется как насыщенность программы условными выражениями.

При этом вводятся две характеристики:

CL – абсолютная сложность программы, характеризующаяся количеством операторов условия;

cl – относительная сложность программы, характеризующаяся насыщенностью программы операторами условия, *т. е. cl определяется как отношение CL к общему числу операторов.*

Используя метрику Джилба, её дополнили её еще одной составляющей, а именно характеристикой максимального уровня вложенности оператора *CLI*, что позволило не только уточнить анализ по условным операторам, но и успешно применить метрику Джилба к анализу циклических конструкций.

Метрика граничных значений

Большой интерес представляет оценка сложности программ по методу граничных значений.

Пусть $G=(V,E)$ – ориентированный граф программы с единственной начальной и единственной конечной вершинами. В этом графе число входящих в вершину дуг называется **отрицательной степенью вершины**, а число исходящих из вершины дуг – **положительной степенью вершины**.

Тогда набор вершин графа можно разбить на две группы:

- **принимающие вершины** – у которых положительная степень ≤ 1 ;
- **вершины отбора** – у которых положительная степень ≥ 2 .

Для получения оценки по методу граничных значений необходимо разбить граф G на максимальное число подграфов G' , удовлетворяющих следующим условиям:

- вход в подграф осуществляется только через вершину отбора;
- каждый подграф включает вершину (называемую нижней границей подграфа), в которую можно попасть из любой другой вершины подграфа.

Каждая принимающая вершина имеет скорректированную сложность равную 1. Конечная вершина (частный случай принимающей) имеет скорректированную сложность – 0. Скорректированная сложность вершины отбора равна числу вершин в её подграфе (исключая вершину выбора, через которую осуществляется вход в подграф). Вершина отбора входит в свой подграф, если образует петлю в графе. Вершина отбора соединенная сама с собой дугой петлей, образует свой подграф.

Скорректированные сложности всех вершин графа суммируются, образуя **абсолютную граничную сложность программы** (S_a).

После этого определяется **относительная граничная сложность программы**:

$$S_0 = 1 - (v - 1) / S_a,$$

где

S_0 – относительная граничная сложность программы;

S_a – абсолютная граничная сложность программы,

v – общее число вершин графа программы.

Третья группа метрик сложности программ (оценка сложности потока данных) применяется для верификации использования, конфигурации и размещения данных в программах.

Метрика Чепина

Суть метода состоит в анализе характера использования переменных в коде программы. Существует несколько ее модификаций.

Все множество переменных программы разбивается на 4 функциональные группы:

- P – **параметрические** (вводимые переменные для расчетов и для обеспечения вывода).
- M – **модифицируемые**, или создаваемые внутри программы переменные.
- C – **контролирующие** (переменные, участвующие в управлении работой программного модуля).
- T – **временные** (не используемые в программе переменные), «паразитные».

Поскольку каждая переменная может выполнять одновременно несколько функций, необходимо учитывать ее в каждой соответствующей функциональной группе.

Расчет метрики Чепина выполняется по формуле:

$$Q=a_1\cdot P+a_2\cdot M+a_3\cdot C+a_4\cdot T,$$

где a_1, a_2, a_3, a_4 – весовые коэффициенты.

Весовые коэффициенты использованы для отражения различного влияния на сложность программы каждой функциональной группы.

По мнению автора метрики, наибольший вес, равный трем, имеет функциональная группа CC , так как она влияет на поток управления программы. Весовые коэффициенты остальных групп распределяются следующим образом: $a_1=1; a_2=2; a_4=0.5$. Весовой коэффициент группы T не равен нулю, поскольку «паразитные» переменные не увеличивают сложности потока данных программы, но иногда затрудняют ее понимание.

С учетом весовых коэффициентов выражение примет вид:

$$Q=1\cdot P+2\cdot M+3\cdot C+0.5\cdot T.$$

Помимо полной метрики Чепина распространены ее варианты, когда анализу и разбиению на четыре группы подвергаются только переменные из определенного списка программы (например, только переменные, которые содержатся в списке параметров операторов ввода/вывода).

Метрика Кафура

Метрика создана на основе концепции информационных потоков.

Для использования данной меры вводятся понятия локального и глобального потока. На основе этих понятий вводится величина «*информационная сложность процедуры*» (I), которая рассчитывается как произведение:

$$I=length \cdot (fan_{in} \cdot fan_{out}),$$

где

length – сложность текста подпрограммы;

fan_in – число локальных потоков входящих внутрь процедуры плюс число структур данных, из которых процедура берёт информацию;

fan_out – число локальных потоков, исходящих из процедуры плюс число структур данных, которые обновляются процедурой.

Сложность текста процедуры измеряется через какую-нибудь из метрик размерности – Холстеда, число строк кода или другие.

Информационную сложность модуля можно определить как сумму информационных сложностей входящих в него процедур.

Однако более интересной для применения является характеристика «Информационная сложность модуля относительно его структур данных», которая рассчитывается по формуле:

$$J=W \cdot R+W \cdot RW+RW \cdot R+RW \cdot (RW-1),$$

где

W – число процедур, которые только обновляют заданную структуру данных;

R – число процедур, которые только читают информацию из заданной структуры данных;

RW – число процедур, которые и читают, и обновляют информацию в структурах данных.

Важно учитывать, что одна процедура может работать с несколькими структурами данных и отнесение её к той или иной группе рассматривается в контексте общего влияния на каждую из структур.

Следует отметить, что рассмотренные метрики сложности программы основаны на анализе исходных текстов программ, что обеспечивает единый подход к автоматизации их расчета.

Стандартизация качества программных средств и систем в РБ

1. Общие сведения о стандартах в области оценки качества в Республике Беларусь

На сегодняшний день имеется множество определений качества ПО, которые, по существу, сводятся к совокупности технических, технологических и эксплуатационных характеристик продукции или процессов, посредством которых они способны отвечать требованиям потребителя и удовлетворять его при применении.

В области оценки качества ПС широко известны такие определения:

- [1061-1998 IEEE Standard for Software Quality Metrics Methodology]

Качество ПО – это степень, в которой ПО обладает требуемой комбинацией свойств.

- [IEEE Std 610.12-1990]

Качество ПО – степень, в которой система, компонент или процесс удовлетворяют потребностям или ожиданиям заказчика или пользователя.

- [ISO/IEC 25000:2014]

Качество ПО – способность программного продукта при заданных условиях удовлетворять установленным или предполагаемым потребностям.

- [ISO 8402:1994 Quality management and quality assurance]

Качество ПО – это совокупность характеристик ПО, относящихся к его способности удовлетворять установленные и предполагаемые потребности.

В условиях, когда техническая сложность изделий возрастает, равно как запросы потребителей и объем предложения на рынке, возникает необходимость в точном определении уровня качества конкретного продукта. Причем зачастую необходимо определить этот уровень у еще не созданного продукта или услуги. В соответствии с выводами Европейской комиссии по качеству, большинство ошибок, из-за которых изделия выходят из строя, происходят именно на стадии проектирования (~70%), а не при производстве (20%) или эксплуатации (10%).

На процесс разработки и деятельность по оценке качества ПС оказывают влияние следующие обобщенные показатели ПС:

- область применения и назначение ПС;
- тип решаемых задач;
- объем и сложность ПС;

- необходимый состав и требуемые значения характеристик качества ПС, и величина допустимого ущерба из-за их недостаточного качества;
- степень связи решаемых задач с реальным масштабом времени или допустимой длительностью ожидания результатов решения задачи;
- прогнозируемые значения длительности эксплуатации и перспектива создания множества версий ПС;
- предполагаемый тираж производства и применения ПС;
- степень необходимой документированности ПС.

В настоящее время на территории Республики Беларусь действуют следующие основные стандарты в области оценки качества ПС:

- межгосударственный стандарт стран СНГ **ГОСТ 28806–90. Качество программных средств. Термины и определения;**
- межгосударственный стандарт стран СНГ **ГОСТ 28195–99. Оценка качества программных средств. Общие положения;**
- национальный стандарт Беларуси **СТБ ИСО/МЭК 9126–2003. Информационные технологии. Оценка программной продукции. Характеристики качества и руководства по их применению.**

В стандарте ГОСТ 28806–90 даются основные термины и определения, принятые в области обеспечения качества программного обеспечения.

ГОСТ 28195–99 определяет **ОЦЕНКУ КАЧЕСТВА программных средств** как:
– совокупность операций, включающих выбор номенклатуры характеристик качества оцениваемого программного средства, определение значений этих характеристик и сравнение их с базовыми значениями.

В соответствии с данным стандартом оценка качества должна проводиться применительно ко всем работам ЖЦ ПС при планировании характеристик качества ПС, контроле качества в процессе разработки, проверке эффективности модификации ПС в процессе сопровождения.

Основными задачами, решаемыми при оценке качества программного средства, по ГОСТ 28195–99 являются:

- 1) планирование номенклатуры характеристик и показателей качества;
- 2) планирование уровня качества;
- 3) выбор методов контроля показателей качества;
- 4) контроль значений показателей качества в процессе ЖЦ ПС;
- 5) выбор базовых образцов по подклассам и группам;
- 6) принятие решения о соответствии реальных значений показателей качества установленным требованиям.

ГОСТ 28195–99 определяет **ПОКАЗАТЕЛЬ КАЧЕСТВА ПРОДУКТА** как:
– количественная характеристика одного или нескольких свойств продукции,

составляющих ее качество, рассматриваемая применительно к определенным условиям ее создания и эксплуатации или потребления.

Обеспечение и оценка качества выполняются в течение ЖЦ ПС и неразрывно связаны с положениями стандарта **СТБ ИСО/МЭК 12207–2003**.

2. Основные понятия и определения

Программное обеспечение сегодня находит широкое применение во всех сферах жизнедеятельности человека. Естественно, от правильности его работы зависят результаты труда и безопасность многих людей. В этой связи особое внимание уделяется вопросам оценки и управления качеством на протяжении всего ЖЦ ПС.

Предполагается, что если ПС разрабатывается на основе спецификации с описанием требований и видения продукта, то качеством может выступать точное соответствие этим спецификациям (при условии их корректности).

Спецификация (*specificatio*, от лат. *species* — вид, разновидность и *facio* – делаю)

- Определение и перечень специфических особенностей, уточнённая классификация чего-либо;
- Один из основных документов системы технической документации.

В большинстве случаев, либо при отсутствии точной спецификации, критерием качества служит то, насколько пользователи удовлетворены ПС и/или соответствующими услугами.

Таким образом:

Качество (*quality*) ПС

Совокупность черт и характеристик ПС, которые влияют на его способность удовлетворять заданные установленные и подразумеваемые потребности пользователей.

Существует ряд международных, межгосударственных и национальных стандартов, посвященных вопросам управления качеством ПС, включая планирование, обеспечение и контроль качества ПС. Основным компонентом контроля качества ПС и основой для его обеспечения является оценка качества.

В данном курсе будут использоваться основные термины, которые соответствуют определениям в стандартах ГОСТ 28806–90, СТБ ИСО/МЭК 9126–2003, ISO/IEC 25023:2016, ISO/IEC 25040:2011.

Свойство ПС (*software attribute*)

Отличительная особенность ПС, которая может проявляться при его создании, использовании, анализе или изменении.

Характеристика качества ПС (*software quality characteristic*)

Набор свойств ПС, посредством которых описывается и оценивается его качество.

Подхарактеристика качества ПС (*software quality subcharacteristic*)

Характеристика качества ПС, входящая в состав другой характеристики качества.

Уровень качества функционирования (*Уровень пригодности, level of performance*)

Степень удовлетворения потребности, представленная конкретным набором значений характеристик качества.

Подразумеваемые потребности (*implied needs*)

Потребности, которые не были установлены, но являются действительными потребностями при использовании продукта в конкретных условиях.

Ранжирование (*rating*)

Действие по отнесению измеренного значения к соответствующему уровню ранжирования.

Уровень ранжирования (*Уровень оценки, rating level*)

Точка на порядковой шкале, которая используется для категоризации шкалы измерения.

Уровень оценки позволяет ранжировать ПС в соответствии с установленными или подразумеваемыми потребностями.

Соответствующие уровни ранжирования могут быть связаны с различными точками зрения на качество. Например, пользователей, администраторов или разработчиков.

Атрибут (*attribute*)

Измеримое физическое или абстрактное свойство ПС.

Атрибуты могут быть внешними или внутренними.

Измерение (*measurement*)

Использование метрики для присвоения атрибуту продукта значения (числа или категории) из шкалы.

Индикатор (*indicator*)

Мера, которая может использоваться для оценки или прогнозирования другой меры.

Мера (*measure*)

Число или категория, присваиваемая атрибуту продукта путем измерения.

Мера косвенная (*indirect measure*)

Мера атрибута, которая получена из мер одного или большего числа других атрибутов.

Мера прямая (*direct measure*)

Мера атрибута, которая не зависит от меры любого другого атрибута.

Метрика (*metric*)

Определенный метод и шкала измерения.

*Метрики могут быть **внутренними**, **внешними** или **метриками качества в использовании**; **прямыми** или **косвенными**.*

Метрики включают методы для категоризации качественных данных (данных, которые нельзя измерить количественно).

Внешнее качество (*external quality*)

Степень, в которой продукт удовлетворяет установленные и подразумеваемые потребности при использовании в заданных условиях.

Внешняя мера (*external measure*)

Косвенная мера продукта, полученная из мер поведения системы, частью которой он является.

Внешние меры могут использоваться для оценки атрибутов качества промежуточных продуктов ближе к конечным целям проекта.

Внутреннее качество (*internal quality*)

Полный набор атрибутов продукта, определяющих его способность удовлетворять установленные и подразумеваемые потребности при использовании в заданных условиях.

Внутренняя мера (*internal measure*)

Собственная мера продукта.

*Внутренняя мера может быть **прямая** или **косвенная**.*

Качество в использовании (*quality in use*)

Степень, в которой программный продукт, используемый заданными пользователями, удовлетворяет их потребности в достижении заданных целей с результативностью, продуктивностью, безопасностью и удовлетворенностью в заданном контексте использования.

Контекст использования (*context of use*)

Пользователи, задания, среда (аппаратное обеспечение, программное обеспечение и материалы), а также физические и социальные среды, в которых используется данное ПС.

Модель качества (*quality model*)

Набор характеристик и связей между ними, обеспечивающий основу для определения требований к качеству и для оценки качества.

Критерий оценки качества (*software quality assessment criteria*)

Совокупность принятых в установленном порядке правил и условий, с помощью которых устанавливается приемлемость общего качества ПС.

Оценка качества (*quality evaluation*)

Систематическое исследование степени, в которой продукт способен к выполнению указанных требований.

Показатель качества ПС (*software quality feature*)

Признак, определяющий свойство ПС, которое может быть соотнесено с некоторой характеристикой качества.

Отказ (*failure*)

Прекращение способности продукта выполнять требуемую функцию или его неспособность работать в пределах заданных ограничений.

Ошибка (*fault*)

Некорректный шаг, процесс или определение данных в программе.

Оценочный модуль (*evaluation module*)

Пакет технологии оценивания для конкретной характеристики или подхарактеристики качества ПС.

Промежуточный программный продукт (*intermediate software product*)

Продукт процесса разработки программного обеспечения, который используется в качестве входных данных для другой стадии процесса разработки программного обеспечения.

Промежуточный продукт может также быть конечным продуктом.

Шкала (*scale*)

Набор значений с определенными свойствами.

При оценке качества используются следующие типы шкал:

- **номинальная** – соответствует набору категорий; классифицирует программы по признаку наличия или отсутствия некоторого свойства без учета градаций (например, «да», «нет»);
- **порядковая (упорядоченная)** – соответствует упорядоченному набору делений шкалы; позволяет ранжировать свойства путем сравнения с опорными значениями; имеет небольшое количество делений (например, шкала с четырьмя градациями — «отлично», «хорошо», «удовлетворительно», «неудовлетворительно», или с двумя градациями — «удовлетворительно», «неудовлетворительно»);
- **интервальная** – соответствует упорядоченной шкале с равноудаленными делениями; обычно содержит достаточно большое количество делений с количественными значениями (например, шкала с делениями 0, 1, 2, ..., 10);

- **относительная** – соответствует упорядоченной шкале с равноудаленными делениями, оцененными в относительных единицах относительно некоторой абсолютной величины (часто в диапазоне от 0 до 1).

Два первых типа шкал применяются для оценки качественных атрибутов ПС, которые нельзя измерить количественно, и для ранжирования измеренных значений, третий и четвертый типы – для оценки количественных атрибутов.

3. Иерархическая модель качества программных средств и систем

Стандарты **ГОСТ 28806–90, ГОСТ 28195–99, СТБ ИСО/МЭК 9126–2003** регламентируют выполнение оценки качества ПС и систем на основе **иерархической модели качества**.

В соответствии с данной моделью совокупность свойств, отражающих качество программного средства, представляется в виде многоуровневой структуры. Характеристики на первом (верхнем) уровне соответствуют основным свойствам ПС. Характеристики каждого уровня оцениваются посредством характеристик последующих уровней.

Стандарт ГОСТ 28195–99 определяет **четырёхуровневую иерархическую модель оценки качества ПС**. Номенклатура характеристик и подхарактеристик первых двух уровней является обязательной, а номенклатура подхарактеристик третьего и четвертого уровней – рекомендуемой.

Стандарт СТБ ИСО/МЭК 9126–2003 представляет собой аутентичный перевод международного стандарта ISO/IEC 9126:1991. В нем приведен метод оценки качества ПС, основанный на **трехуровневой иерархической модели качества**.

Стандарты ГОСТ 28806–90 и СТБ ИСО/МЭК 9126–2003 описывают первые два уровня иерархической модели качества. При этом номенклатура характеристик первого уровня является обязательной, а номенклатура характеристик второго уровня (подхарактеристик) – рекомендуемой.

Вышеназванные стандарты определяют шесть основных характеристик качества ПС, находящихся на верхнем уровне модели качества. Следует отметить, что характеристики верхнего уровня, регламентированные ГОСТ 28806–90 и СТБ ИСО/МЭК 9126–2003, соответствуют принятым в мировой практике. В то же время характеристики и подхарактеристики, определенные в ГОСТ 28195–99, частично не соответствуют иерархической модели качества, принятой в международных стандартах.

В стандартах ГОСТ 28806–90 и СТБ ИСО/МЭК 9126–2003 определены следующие характеристики качества ПС (характеристики верхнего уровня):

1. **Функциональность** (Functionality) – совокупность свойств ПС, определяемая наличием и конкретными особенностями набора функций, способных удовлетворять заданные или подразумеваемые потребности.
2. **Надежность** (Reliability) – совокупность свойств, характеризующая способность ПС сохранять заданный уровень пригодности в заданных условиях в течение заданного интервала времени.
3. **Удобство использования** (Практичность, Usability) – совокупность свойств программного средства, характеризующая усилия, необходимые для его использования, и индивидуальную оценку результатов его использования заданным или подразумеваемым кругом пользователей.
4. **Эффективность** (Efficiency) – совокупность свойств программного средства, характеризующая те аспекты его уровня пригодности, которые связаны с характером и временем использования ресурсов, необходимых при заданных условиях функционирования.
5. **Сопровождаемость** (Maintainability) – совокупность свойств программного средства, характеризующая усилия, которые необходимы для его модификации.
6. **Мобильность** (Portability) – совокупность свойств программного средства, характеризующая приспособленность для переноса из одной среды функционирования в другие.

4. ГОСТ 28195–99

Стандартом **ГОСТ 28195–99** рекомендован **метод интегральной оценки качества ПС**, основанный на иерархической модели качества.

4.1. Метод оценки качества

В соответствии с методом **интегральной оценки качества ПС** выбор номенклатуры показателей качества для конкретного программного средства осуществляется с учетом его назначения и требований области применения в зависимости от принадлежности ПС к тому или иному подклассу, определяемому общесоюзным классификатором продукции (ОКП).

Оценка качества ПС производится на всех фазах жизненного цикла

ГОСТ 28195–99 базируется на следующих процессах и фазах ЖЦ ПС:

1. Процесс разработки:

- фаза анализа;
- фаза проектирования;
- фаза реализации;
- фаза тестирования;
- фаза изготовления.

2. Процесс применения:

- фаза внедрения;
- фаза эксплуатации;
- фаза сопровождения.

Вышеприведенные фазы представляют собой временные периоды, соответствующие работам, совокупностям работ или процессам ЖЦ ПС, определенным стандартом СТБ ИСО/МЭК 12207–2003.

Оценка качества ПС заключается в следующих действиях:

- выбор номенклатуры показателей,
- их оценка,
- сопоставление их с базовыми значениями.

Основу рассматриваемого метода оценки качества ПС составляет четырехуровневая **ИЕРАРХИЧЕСКАЯ МОДЕЛЬ КАЧЕСТВА**.

ГОСТ 28195–99 предлагает следующую терминологию для показателей качества каждого уровня:

уровень 1 - ФАКТОРЫ КАЧЕСТВА (в терминологии, принятой в международных стандартах, соответствуют *характеристикам качества*);

уровень 2 - КРИТЕРИИ КАЧЕСТВА (в международной терминологии – *подхарактеристики качества*);

уровень 3 - МЕТРИКИ (полностью соответствует международной терминологии);

уровень 4 - ОЦЕНОЧНЫЕ ЭЛЕМЕНТЫ, или **единичные показатели** (в международных стандартах данный уровень отсутствует).

На рисунке 1 приведены факторы и критерии качества, определенные в стандарте ГОСТ 28195–99. Для каждого из факторов качества (первый уровень) составляется своя иерархическая модель, отражающая взаимосвязь факторов, критериев, метрик и оценочных элементов.

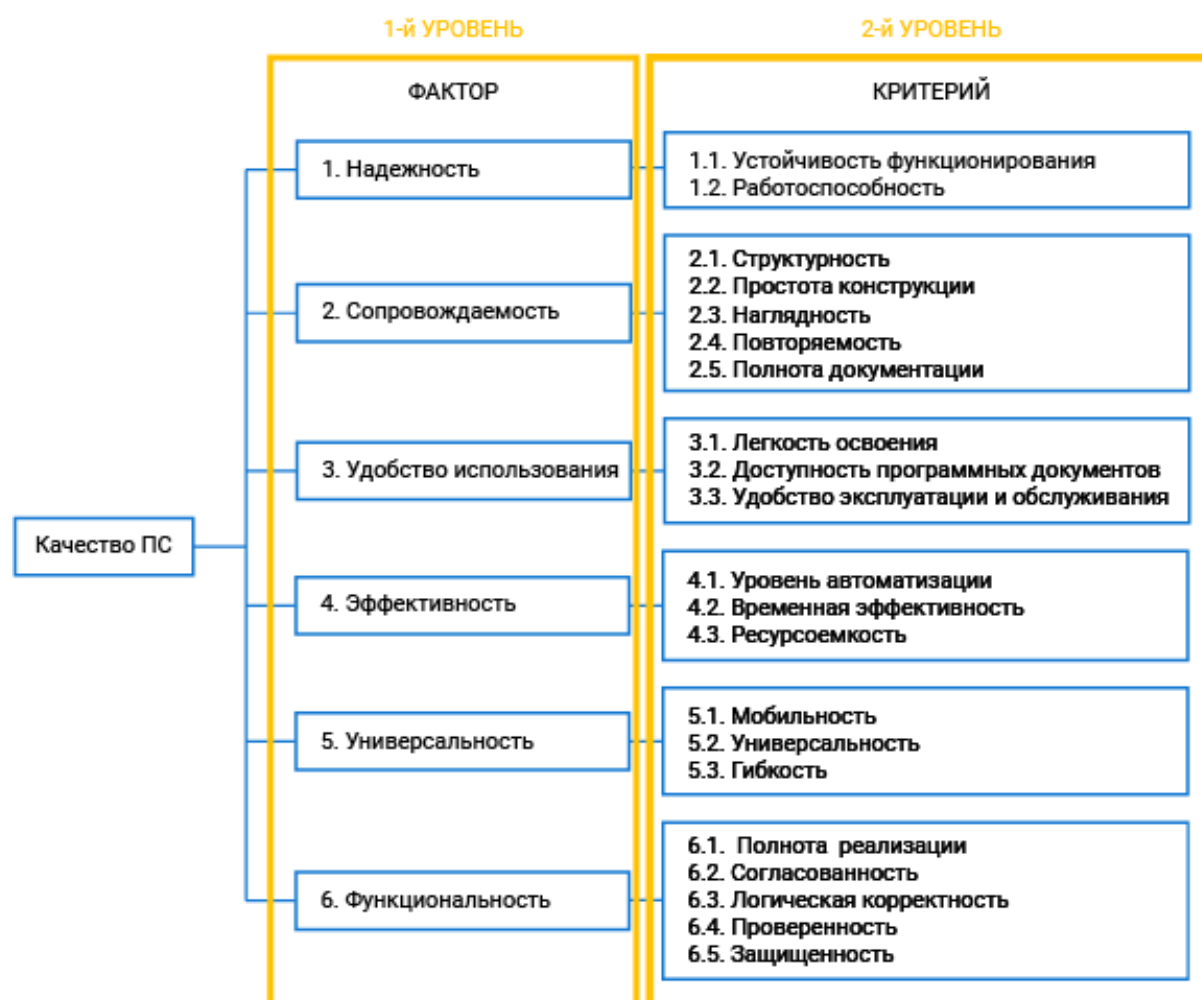


Рисунок — Иерархическая модель качества ПС по ГОСТ 28195–99

Вид данной модели зависит от фазы ЖЦ ПС. Выбор оценочных элементов для каждой метрики зависит от функционального назначения ПС и формируется с учетом данных, ранее полученных при проведении испытаний ПС и эксплуатации аналогичных программ.

Для выбора оценочных элементов ГОСТ 28195–99 предлагает перечень специальных таблиц, содержащих наименование элемента, метод его оценки и применяемость элемента для различных подклассов ПС.

В каждой такой таблице код оценочного элемента формируется из пяти символов: **SNNPP**.

Первый символ (буква **S**) указывает на принадлежность элемента фактору.

Два следующих символа (**NN**) – это номер метрики, которой принадлежит оценочный элемент.

Четвертый и пятый символы (**PP**) – это порядковый номер данного оценочного элемента внутри метрики.

В ГОСТ 28195–99 приняты следующие обозначения факторов:

H – Надежность;

C – Сопровождаемость;

Y – Удобство применения;

E – Эффективность;

G – Универсальность (Гибкость);

K – Функциональность (Корректность).

Достоинства метода оценки качества ПС, основанного на рассмотренной иерархической модели:

– Метод позволяет накапливать статистический материал о состоянии различных подклассов ПС в отношении значений метрик и оценочных элементов. Это создает предпосылки для определения их нормативных (базовых) значений по подклассам ПС и может служить основой для деятельности по стандартизации в области программного обеспечения.

– Списки значений метрик и оценочных элементов являются основой для деятельности по управлению качеством в процессе разработки ПС.

– Возможно создание инструментальных средств с целью автоматизации оценки качества ПС для тех показателей, которые такую оценку допускают.

4.2. Расчет оценки качества

В соответствии с **ГОСТ 28195–99**, для оценки качества ПС необходимо выполнить следующую последовательность действий из десяти шагов:

1. На фазе анализа проводится выбор показателей и их базовых значений.
2. Для показателей качества на всех уровнях принимается единая шкала оценки (например, от 0 до 1).
3. В процессе оценки качества на каждом уровне (кроме уровня оценочных элементов) проводится вычисление двух величин:

- абсолютного показателя качества P_{ij} ,
- относительного показателя качества R_{ij}

где j – порядковый номер показателя данного уровня для i -го показателя вышестоящего уровня.

Относительный показатель качества R_{ij} является функцией отношения показателя P_{ij} и его базового значения:

$$R_{ij}=P_{ij}/P_{bij}$$

ГОСТ 28195–99 содержит таблицу с базовыми значениями для характеристик качества второго уровня (критериев). Данные значения определяются подклассом программного средства в соответствии с ОКП.

Базовые значения для показателей первого и третьего уровней формируются методом экспертного опроса с учетом назначения ПС или на основании показателей существующих аналогов или расчетного эталонного ПС. Значения базовых показателей ПС должны соответствовать значениям показателей, отражающих современный уровень качества и прогнозируемый мировой уровень.

4. Каждый показатель качества второго и третьего уровней характеризуется двумя параметрами:

- количественным значением,
- весовым коэффициентом V_{ij} .

Сумма весовых коэффициентов всех показателей некоторого уровня, относящихся к показателю вышестоящего уровня, постоянна и равна 1:

$$\sum_{j=1}^J V_{ij} = 1$$

где J – общее количество всех показателей j -го уровня, относящихся к i -му показателю вышестоящего уровня, определенных в стандарте.

ГОСТ 28195–99 содержит таблицы, содержащие перечни весовых коэффициентов для характеристик второго и третьего уровней (критериев и метрик). Количественные величины весовых коэффициентов зависят от фазы ЖЦ ПС и подкласса ПС в соответствии с ОКП.

5. Определение усредненной оценки m_{kq} оценочного элемента по нескольким его значениям (измерениям) m_{qt} осуществляется по формуле (**формула для вычисления значений показателей качества 4-го уровня**):

$$m_{kq} = \frac{1}{T} \sum_{t=1}^T m_{qt}$$

где

k – порядковый номер метрики;

q – порядковый номер оценочного элемента;

T – число значений (измерений) оценочного элемента;

t – номер значения оценочного элемента.

6. Итоговая оценка k -ой метрики j -го критерия рассчитывается как среднее оценочных элементов по формуле (**формула для вычисления значений показателей качества 3-го уровня**):

$$PK_{ij} = \frac{\sum_{n=1}^N (PM_{jk} VM_{jk})}{\sum_{n=1}^N VM_{jk}}$$

где

M – признак метрики;

Q – число оценочных элементов, реально используемых при оценке k -й метрики.

7. Абсолютные показатели j -го критерия i -го фактора вычисляются как отношение суммы показателей соответствующих метрик с учетом их вклада к сумме учитываемых весовых коэффициентов (**формула для вычисления значений показателей качества 2-го уровня**):

$$PK_{ij} = \frac{\sum_{n=1}^N (PM_{jk} VM_{jk})}{\sum_{n=1}^N VM_{jk}}$$

где

n – число метрик, относящихся к j -му критерию, реально используемых при оценке;

K – признак критерия.

8. Относительные значения R_{kij} j -го критерия i -го фактора P_{kij} по отношению к базовому значению P_{kbij} определяются по формуле

$$R_{kij} = P_{kij} / P_{kbij}$$

9. Абсолютные и относительные значения i -го фактора качества определяются по формулам из абсолютных и относительных значений соответствующих критериев качества и их весовых коэффициентов (**формулы для вычисления значений показателей качества 1-го уровня**)

$$P_{Fi} = \sum_{j=1}^N (P_{kij} V_{kij}) / \sum_{j=1}^N V_{kij}$$

$$R_{Fi} = \sum_{j=1}^N (R_{kij} V_{kij}) / \sum_{j=1}^N V_{kij}$$

где

F – признак фактора;

N – число критериев качества, относящихся к i -му фактору, реально используемых при оценке.

10. Общая оценка качества в целом формируется экспертами по набору полученных значений факторов качества. Общая оценка качества ПС может быть получена также как усредненное значение показателей факторов, реально используемых при оценке.

Рассмотрим представление иерархической модели качества ПС для фактора «Сопровождаемость» для различных фаз жизненного цикла программных средств.

На рисунке 1 представлены первый и второй уровни модели качества фактора «Сопровождаемость».

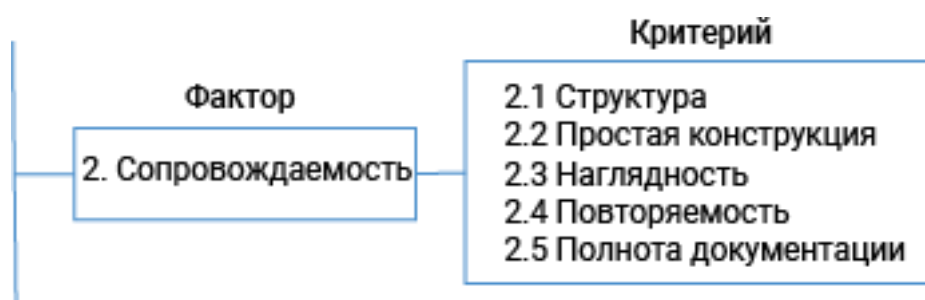


Рис. 1 — Фактор качества ПС «Сопровождаемость» и его критерии

На рис. 2–7 приведены три верхних уровня иерархической модели для фаз ЖЦ, соответственно: анализа, проектирования, реализации, тестирования, изготовления, сопровождения.

Примечание

Номера на рисунках 2–7 возле метрик соответствуют номерам метрик рассматриваемого фактора в стандарте.



Рис. 2 — Модель сопровождаемости для фазы анализа



Рис. 3 — Модель сопровождаемости для фазы проектирования

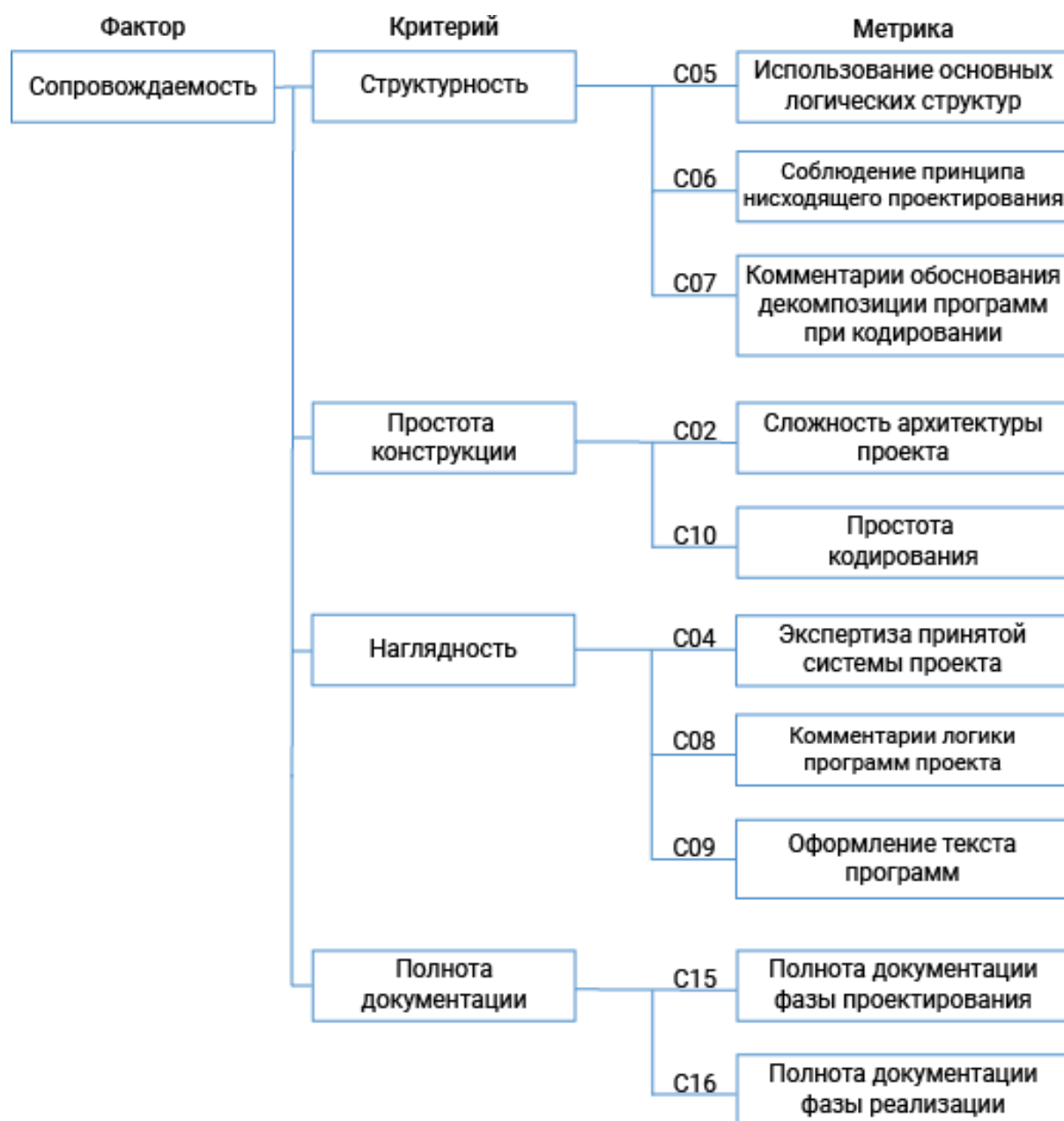


Рис. 4 — Модель сопровождаемости для фазы реализации

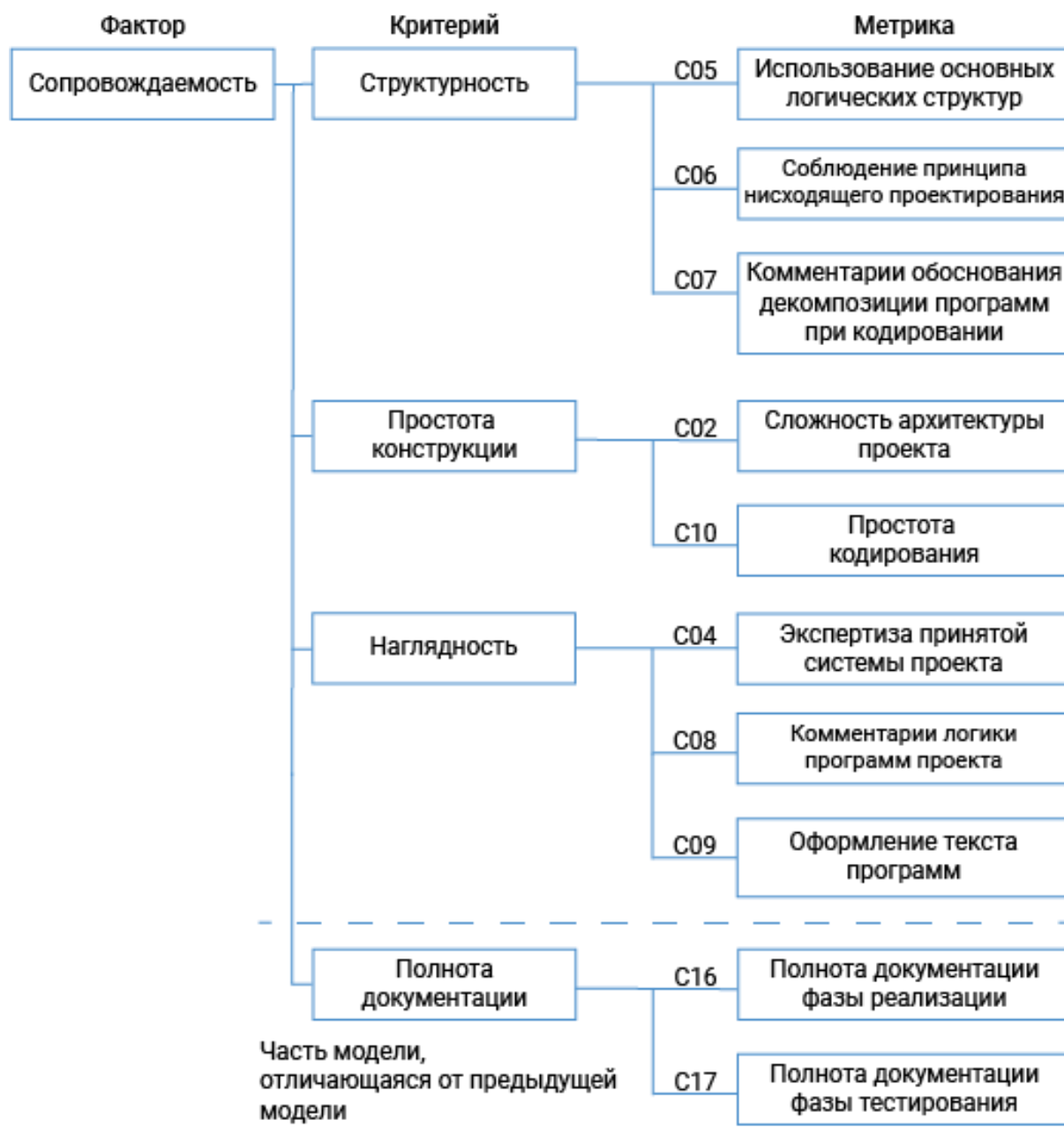


Рис. 5 — Модель сопровождаемости для фазы тестирования

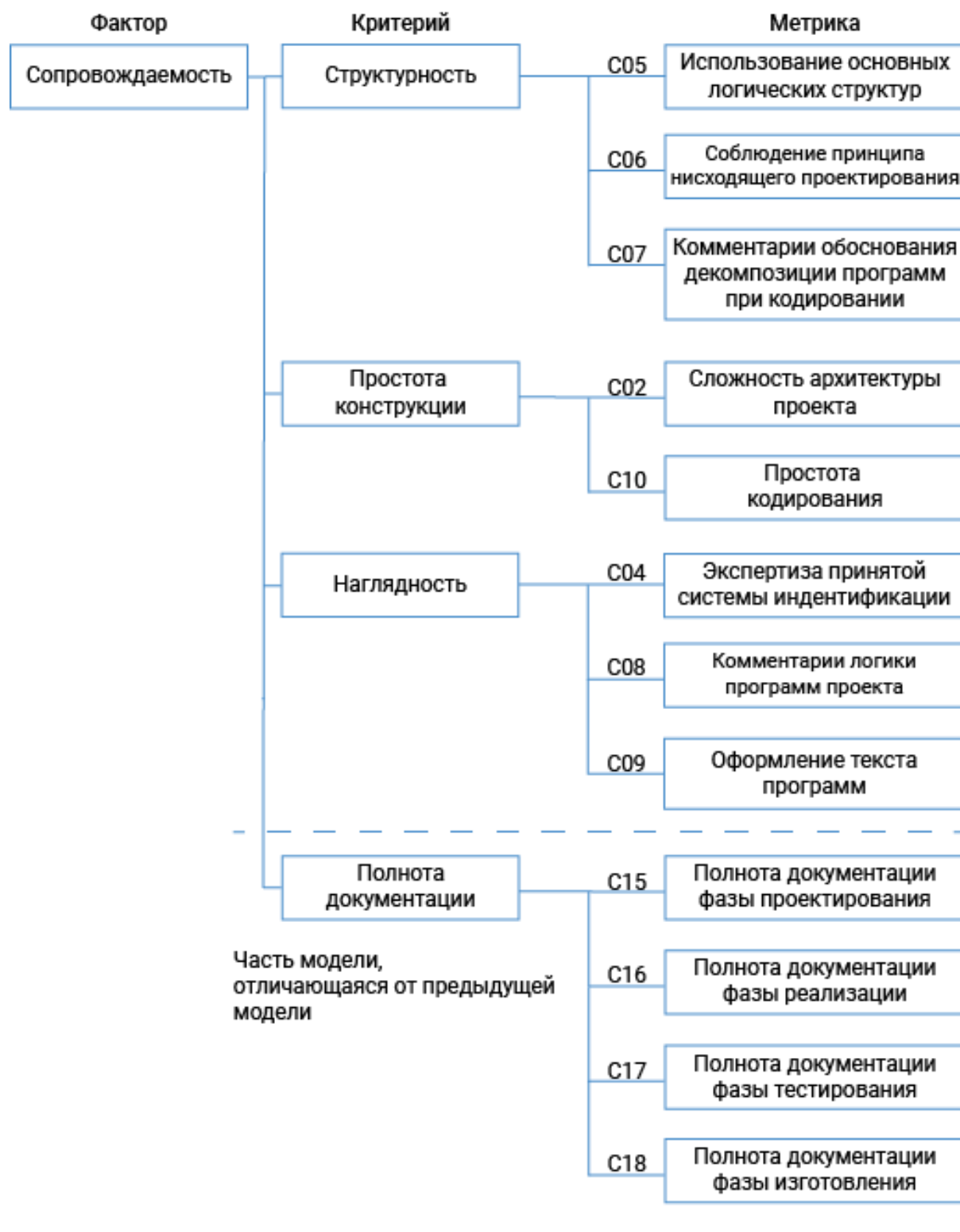


Рис. 6 — Модель сопровождаемости для фазы изготовления

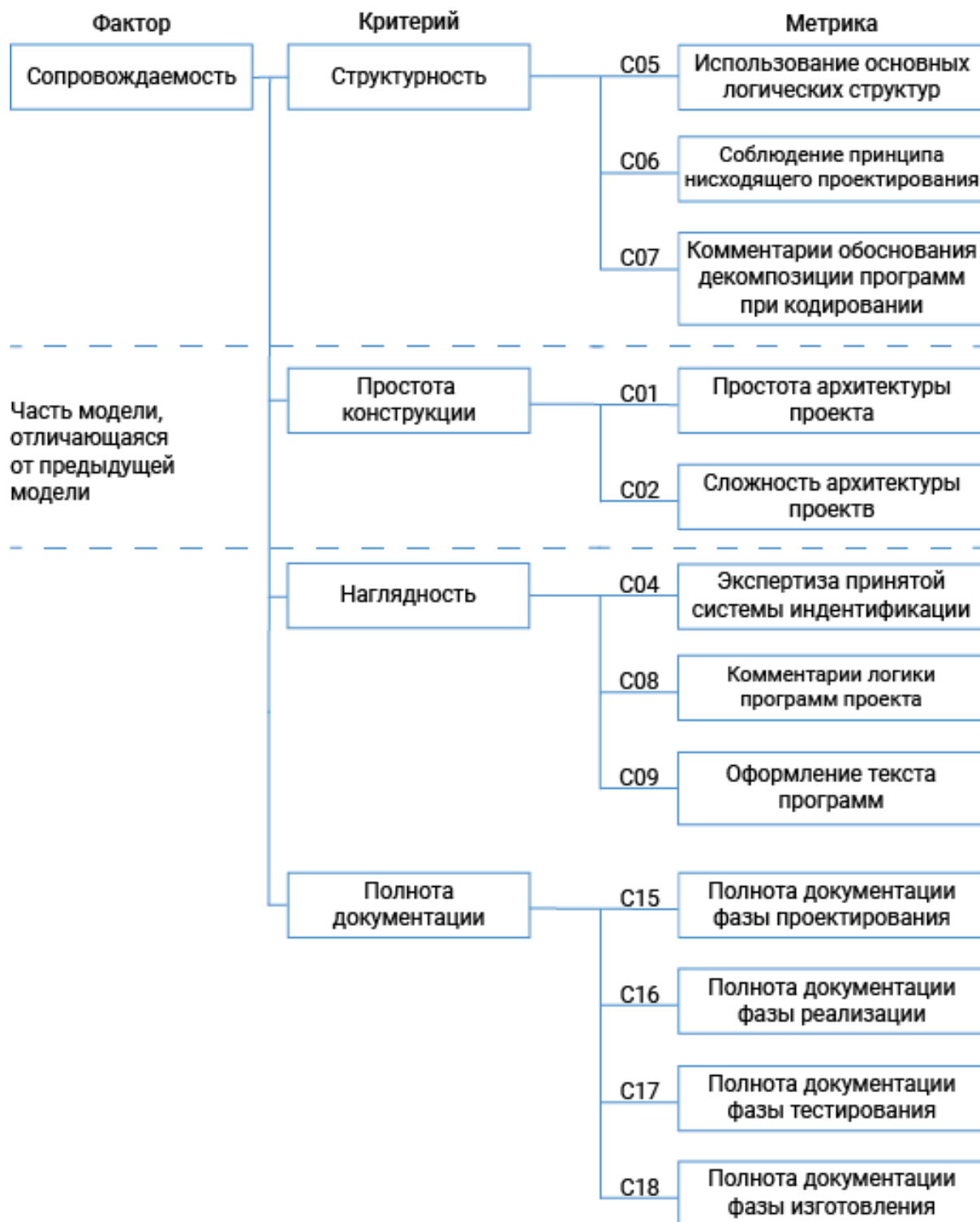


Рис. 7 — Модель сопровождаемости для фазы сопровождения

Таблица 1 — Оценочные элементы фактора Сопровождаемость

Код элемента	Наименование объектов	Метод оценки
C0101	Наличие модульной схемы программы	Экспертный
C0102	Оценка программы по числу уникальных модулей	Экспертный
C0201	Наличие ограничений на размеры модуля	Экспертный
C0301	Наличие проверки корректности передаваемых данных	Экспертный
C0302	<p>Оценка простоты программы по числу точек входа и выхода W:</p> $W=1(D+1) \cdot (F+1)$ <p>где</p> <p>D – общее число точек входа в программу;</p> <p>F – общее число точек выхода из программы</p>	Экспертный
C0303	Осуществляется ли передача результатов работы модуля через вызывающий его модуль	Экспертный
C0304	Осуществляется ли контроль за правильностью данных, поступающих в вызывающий модуль от вызываемого	Экспертный
C0305	Наличие требований к независимости модулей программы от типов и форматов выходных данных	Экспертный
C0401	Наличие требований к системе идентификации	Экспертный

Код элемента	Наименование объектов	Метод оценки
C0501	Наличие требований по использованию основных логических структур	Экспертный
C0601	Использование при построении программ метода структурного программирования	Экспертный + измерительный
C0602	Соблюдение принципа разработки программы сверху вниз	Экспертный
C0603	Оценка программы по числу циклов с одним входом и одним выходом	Экспертный + измерительный
C0604	Оценка программы по числу циклов	Экспертный + измерительный
C0701	Наличие комментариев обоснования декомпозиции программ при кодировании	Экспертный
C0801	Наличие комментариев ко всем машинно-зависимым частям программы	Экспертный
C0802	Наличие комментариев ко всем машинно-зависимым операторам программы	Экспертный
C0803	Наличие комментариев в точках входа и выхода программы	Экспертный
C0901	Соответствие комментариев принятым соглашениям	Экспертный
C0902	Наличие комментариев-заголовков программы с указанием ее структурных и функциональных характеристик	Экспертный
C0903	Оценка ясности и точности описания последовательности функционирования всех элементов программы	Экспертный
C1001	Используется ли язык высокого уровня	Экспертный

Код элемента	Наименование объектов	Метод оценки
C1002	<p>Оценка простоты программы по числу переходов по условию:</p> $U = (1 - A/B)$ <p>где</p> <p>A – общее число переходов по условию;</p> <p>B – общее число исполняемых операторов</p>	Измерительный + расчетный
C1301	Использование типовых компонентов ПС	Экспертный
C1401	Использование типовых проектных решений	Экспертный
C1501	Наличие программных спецификаций и требований, предъявляемых к программным средствам	Экспертный
C1502	Наличие документов, содержащих детальное описание принятых проектных решений	Экспертный
C1503	Наличие заключений по принятым проектным решениям, требованиям и спецификациям	Экспертный
C1601	Наличие описания и схемы иерархии модулей программы	Экспертный
C1602	Наличие описания основных функций	Экспертный
C1603	Наличие описания частных функций	Экспертный
C1604	Наличие описания данных	Экспертный
C1605	Наличие описания алгоритмов	Экспертный
C1606	Наличие описания интерфейсов	Экспертный

Код элемента	Наименование объектов	Метод оценки
C1607	Наличие описания интерфейсов с пользователем	Экспертный
C1608	Наличие описания используемых числовых методов	Экспертный
C1609	Наличие описания всех параметров	Экспертный
C1610	Наличие описания методов настройки системы	Экспертный
C1611	Наличие описания всех диагностических сообщений	Экспертный
C1612	Реализация всех исходных модулей	Экспертный
C1701	Наличие описания всех диагностических сообщений эталонного образца	Экспертный
C1702	Наличие требований к тестированию программ	Экспертный
C1703	Достаточность требований к тестированию программ	Экспертный
C1801	Наличие описания процедуры изготовления эталонного образца	Экспертный
C1802	Наличие описания процедуры изготовления рабочих копий	Экспертный
C1803	Наличие описания процедуры контроля на идентичность рабочих копий с эталонным образцом	Экспертный

Как видно из таблицы 1, разные оценочные элементы могут быть получены разными методами оценки. Например, оценочный элемент **C0101** оценивается при помощи группы экспертов (экспертный метод). В тоже время для элемента **C0302** используется два метода оценки – измерительный и расчетный (сначала измеряются показатели D и F , а затем рассчитывается параметр W).

Международная стандартизация качества программных средств и систем

1. ИСО/МЭК 9126

Международный стандарт **ISO/IEC 9126:1991** (Информационная технология – Оценка программного продукта – Характеристики качества и руководства по их применению) положен в основу **СТБ ИСО/МЭК 9126–2003**.

В 2001 году был преобразован в группу:

- **ISO/IEC 9126–1–4**
- **ISO/IEC 14598–1–6**

ISO/IEC 9126–1:2001 – Часть 1: Модель качества;

ISO/IEC TR 9126–2:2003 – Часть 2: Внешние метрики;

ISO/IEC TR 9126–3:2003 – Часть 3: Внутренние метрики;

ISO/IEC TR 9126–4:2004 – Часть 4: Метрики качества в использовании.

Первая часть стандарта ISO/IEC 9126–1:2001 по существу является пересмотренной редакцией стандарта ISO/IEC 9126:1991. В данной части определены два верхних уровня (характеристики и подхарактеристики) иерархической модели качества, приведены общие требования к метрикам качества, даны рекомендации по их выбору. При этом сохранена та же номенклатура из шести базовых характеристик качества ПС.

В отличие от ISO/IEC 9126:1991 подхарактеристики второго уровня стали нормативными, а не рекомендуемыми, определены две части модели качества (модель внутреннего и внешнего качества и модель качества в использовании) и исключен процесс оценки качества (он теперь содержится в стандарте **ISO/IEC 14598**).

Во второй части стандарта ISO/IEC TR 9126–2:2003 определяются метрики количественного измерения внешнего качества ПС. Внешние метрики.

В третьей части стандарта ISO/IEC TR 9126–3:2003 определяются метрики количественного измерения внутреннего качества ПС. Внутренние метрики.

В четвертой части стандарта ISO/IEC TR 9126–4:2004 определяются метрики количественного измерения качества в использовании. Метрики качества в использовании.

ISO/IEC 14598–1:1999 – Информационная технология – Оценка программного продукта – Часть 1: Общий обзор;

ISO/IEC 14598–2:2000 – Программная инженерия – Оценка продукта – Часть 2: Планирование и управление;

ISO/IEC 14598–3:2000 – Программная инженерия – Оценка продукта – Часть 3: Процесс для разработчиков;

ISO/IEC 14598–4:1999 – Программная инженерия – Оценка продукта – Часть 4: Процесс для заказчиков;

ISO/IEC 14598–5:1998 – Информационная технология – Оценка программного продукта – Часть 5: Процесс для оценщиков;

ISO/IEC 14598–6:2001 – Программная инженерия – Оценка продукта – Часть 6: Документация модулей оценки.

В первой части стандарта ISO/IEC 14598–1:1999 приведен обзор остальных частей, определена связь ISO/IEC 14598 со стандартами ISO/IEC 9126–1–4 и ISO/IEC 12207:1995. В данной части содержатся общие требования к спецификации и оценке качества, разъясняются концепции оценки.

Вторая часть стандарта ISO/IEC 14598–2:2000 содержит требования и руководство по поддержке оценки. Эта часть стандарта предназначена для применения на уровне организации или ее подразделений.

Третья часть стандарта ISO/IEC 14598–3:2000 предназначена для организаций – разработчиков ПС. Данная часть ориентирована на выполнение оценки ПП, используя собственный технический персонал.

Четвертая часть стандарта ISO/IEC 14598–4:1999 предназначена для организаций, которые планируют приобретать готовый или разрабатываемый программный продукт. В ней определена связь работ процесса заказа из стандарта ISO/IEC 12207:1995 с работами, выполняемыми при оценке ПП.

Пятая часть стандарта ISO/IEC 14598–5:1998 предназначена для использования оценщиком, выполняющим независимую оценку программного продукта. Как правило, персонал оценки работает в независимой организации. В данной части приводятся концепции оценки и требования к процессу оценки. Рассмотрена структура отчета об оценке. Приводятся рекомендации по выбору уровней ранжирования при проведении измерений. Оценка качества программного продукта оценщиком может выполняться по запросу разработчика, заказчика (покупателя) или другой стороны.

Шестая часть стандарта ISO/IEC 14598–6:2001 предназначена для поддержки оценки программного продукта и содержит руководство по документированию модулей оценки.

1.1. Метод оценки качества программных средств и систем

СТБ ИСО/МЭК 9126–2003 определяет метод оценки качества ПС, который основан на **трехуровневой иерархической модели качества**.

На первом уровне модели находятся шесть характеристик качества. Второй уровень составляют подхарактеристики и третий – метрики качества.

Модель процесса оценки, положенная в основу рассматриваемого метода, приведена на рисунке. Данная модель отражает основные стадии и этапы, требуемые для оценки качества ПС.

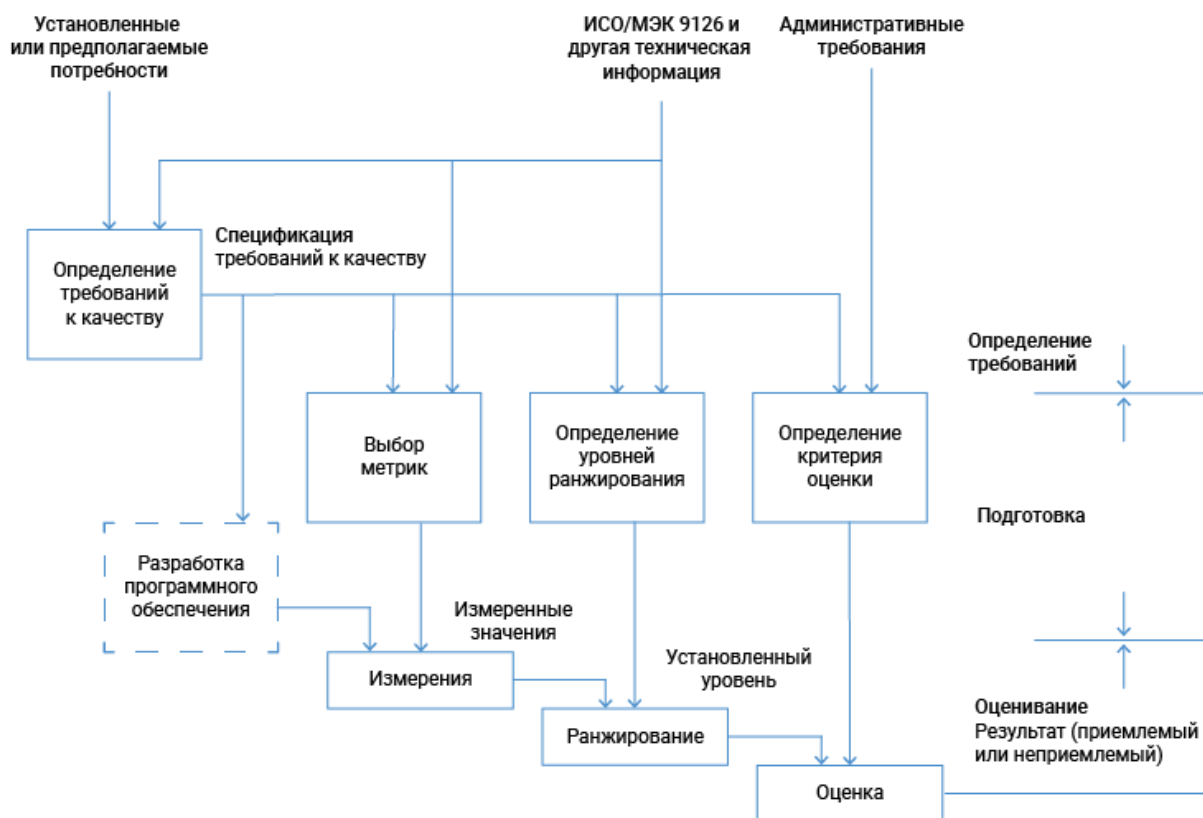


Рисунок. Модель процесса оценки по СТБ ИСО/МЭК 9126–2003

Данный процесс может применяться после любой подходящей работы жизненного цикла для каждого компонента программного продукта.

Процесс оценки состоит из трех стадий:

1. определение требований к качеству ПС,
2. подготовка к оцениванию,
3. процедура оценивания.

Стадия 1. Определение требований к качеству

Целью данной стадии является установка требований в терминах характеристик и подхарактеристик качества. Требования выражают потребности внешнего окружения ПС и должны быть определены до начала разработки. Так как ПС разделяется на компоненты, то требования для ПС в целом могут отличаться от требований для отдельных компонентов.

Стадия 2. Подготовка к оцениванию

Целью второй стадии является подготовка основы для оценивания. Данная стадия состоит из трех этапов.

Этап 2.1. Выбор метрик качества

С учетом иерархической модели качества регламентированной в СТБ ИСО/МЭК 9126–2003 уровень характеристик качества ПС определяется уровнем входящих в них подхарактеристик, а значения подхарактеристик в свою очередь определяются значениями входящих в них метрик.

В стандарте СТБ ИСО/МЭК 9126–2003 набор рекомендуемых метрик отсутствует. В связи с этим существует потребность в установлении метрик, которые успешно соотносятся с подхарактеристиками, а также, соответственно, и с характеристиками ПС. Каждый количественный признак и каждое количественно оцениваемое взаимодействие ПС с его окружением, которые соотносятся с характеристикой, могут быть приняты в качестве метрики. Метрики, используемые в процессе разработки, должны быть соотнесены с соответствующими метриками пользователя, потому что метрики пользователя являются решающими.

Этап 2.2. Определение уровней ранжирования

Для измерения количественных признаков ПС используются метрики качества. Измеренные значения отображаются на некоторой шкале. Данные значения не показывают уровень удовлетворения требований к качеству ПС. Для этой цели шкалы метрик должны быть разделены на диапазоны, соответствующие различным степеням удовлетворения требований.

В стандарте ISO/IEC 25040:2011 приведен пример следующих диапазонов ранжирования:

- разделение шкалы на две категории: неудовлетворительно и удовлетворительно;
- разделение шкалы на четыре категории (отлично, хорошо, удовлетворительно, неудовлетворительно), ограниченные соответственно запланированным уровнем, текущим уровнем для существующего или альтернативного продукта и уровнем худшего случая.

На рисунке 2 представлены предложенные варианты ранжирования.

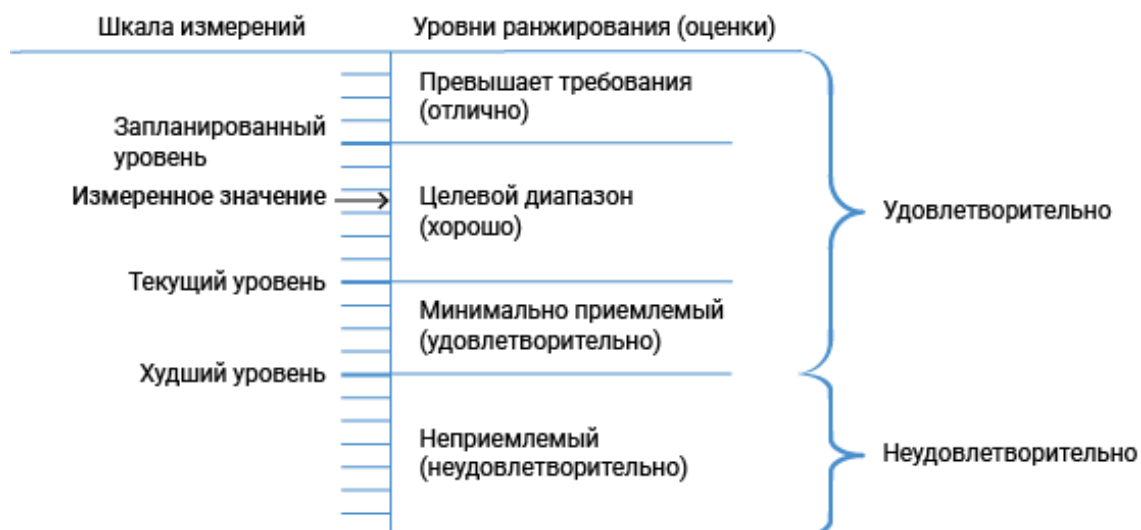


Рис. 2. Варианты ранжирования измеренных значений метрик

Текущий уровень определяется для управления тем, чтобы новая система не становилась хуже по сравнению с существующей.

Запланированный уровень определяет уровень, который считается достижимым при доступных ресурсах.

Уровень худшего случая определяет границу принятия пользователем в случае, если изделие не удовлетворяет запланированному уровню.

Так как качество ПС связано с конкретными потребностями, общие уровни ранжирования невозможны и должны определяться для каждого конкретного оценивания.

Этап 2.3. Определение критерия оценки

Для определения общего качества ПС должна быть учтена вся совокупность результатов оценивания различных метрик. Оценщик должен подготовить для этого процедуры, используя, например, таблицы решений или средние взвешенные значения. Обычно при этом учитываются и другие аспекты, такие как время и стоимость, которые являются косвенными факторами качества ПС.

Стадия 3. Процедура оценивания

Последняя стадия модели процесса оценивания реализуется тремя этапами: «Измерение», «Ранжирование» и «Оценка».

Этап 3.1. Измерение

Для измерения выбранные метрики применяются к ПС.

Результатом являются значения в масштабах метрик.

Этап 3.2. Ранжирование

На этапе ранжирования устанавливается уровень ранжирования для измеренного значения.

Этап 3.3. Оценка

Оценка является последним этапом процесса оценивания ПС, на котором обобщается множество установленных уровней.

Результатом является заключение о качестве ПС (приемлемый или неприемлемый уровень качества).

Недостаток метода

Отсутствие рекомендуемых вариантов метрик и представление метода лишь в общем виде (в виде модели). Это затрудняет его конкретное использование.

2. Стандарты SQaRE

На данный момент в области оценки качества программных средств и систем действуют международные стандарты ISO/IEC серии **Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQaRE)**. Организация серии международных стандартов SQaRE представлена на рис.1.

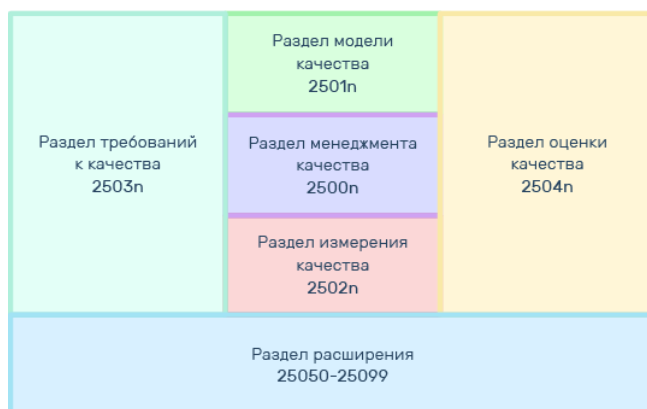


Рисунок 1 – Серия международных стандартов SQaRE

Международные стандарты серии SQaRE разделены на следующие группы:

- **управление качеством (2500n);**

ISO/IEC 25000:2014 Guide to SQaRE.

ISO/IEC 25001:2014 Planning and management.

- **модели качества (2501n);**

ISO/IEC 25010:2011 SQaRE – System and software quality models.

ISO/IEC TS 25011:2017 Service quality models.

ISO/IEC 25012:2008 Data quality model.

- **измерения качества (2502n);**

ISO/IEC 25020:20 19 Measurement reference model and guide.

ISO/IEC 25021:2012 Quality measure elements.

ISO/IEC 25022:2016 Measurement of quality in use.

ISO/IEC 25023:2016 Measurement of system and software product quality.

ISO/IEC 25024:2015 Measurement of data quality.

ISO/IEC CD TS 25025.2 Measurement of IT service quality.

- **требования к качеству (2503n);**

ISO/IEC 25030:20 19 Quality requirements.

- **оценки качества (2504n);**

ISO/IEC 25040:2011 Evaluation process.

ISO/IEC 25041:2012 Evaluation guide for developers, acquirers and independent evaluators.

ISO/IEC 25045:2010 Evaluation module for recoverability.

- **группа расширения (2505n - 25099);**

ISO/IEC 25051:2014 Requirements for quality of Ready to Use Software Product (RUSP) and instructions for testing.

Серия отраслевых форматов ISO/IEC TR 2506n – Common Industry Format (CIF).

Стандарты раздела **«Менеджмент качества» (ISO/IEC 2500n)** определяют общие модели, термины и определения, которые используются далее во всех других международных стандартах серии SQuaRE, регламентируются общие принципы планирования и управления качеством систем и программных средств. Данная группа стандартов содержит также методические материалы по использованию стандартов серии SQuaRE.

Стандарты раздела **«Модели качества» (ISO/IEC 2501n)** содержат описание моделей качества компьютерных систем, программных продуктов и данных. Стандарты содержат также практическое руководство по использованию регламентированных моделей качества.

Стандарты группы **«Измерения качества» (ISO/IEC 2502n)** включают в себя эталонную модель измерения качества программной продукции, математические определения мер качества и практическое руководство по их использованию. В стандартах данной группы приведены меры внутреннего и внешнего качества и меры качества при использовании программного обеспечения. Кроме того, в разделе определены и представлены элементы мер качества, которые формируют основу для обозначенных мер.

Стандарты раздела **«Требования к качеству» (ISO/IEC 2503n)** позволяют определить соответственно требования к качеству на основе моделей и мер качества. Эти требования к качеству могут быть использованы в процессе формирования требований к качеству программного обеспечения до начала разработки или как входные данные для процесса оценки.

Стандарты группы **«Оценка качества» (ISO/IEC 2504n)** содержат требования, рекомендации и методические материалы для оценки программной продукции, выполняемой как оценщиками, так и заказчиками или разработчиками. Также регламентируются правила документирования мер в виде модуля оценки.

Международные стандарты из раздела **расширения (ISO/IEC 2505n–25099)** включают в себя требования к международным стандартам и техническим отчетам по качеству систем и программной продукции в специальных областях приложения, а также серию общих отраслевых форматов для отчетов по практичности (CIF).

Стандарт **ISO/IEC 25000:2014** определяет связь между стандартами серии SQuaRE, преемственность к стандартам *ISO/IEC 9126* и *ISO/IEC 14598*, а также связи с другими международными стандартами, в числе которых: *ISO/IEC*

12207:2017, ISO/IEC/IEEE 15288:2015, ISO/IEC/IEEE 15939:2017, ISO/IEC/IEEE DIS 29119, семейство стандартов ISO 9000.

ISO/IEC 25000:2014 содержит общие модели, в том числе:

- общую эталонную модель,
- модель качества в ЖЦ ПС,
- иерархическую структуру моделей качества, с которыми работает серия SQuaRE.

Общая эталонная модель применения стандартов серии SQuaRE приведена на рисунке 2. Данная модель разработана с целью оказания помощи пользователям в ориентации по стандартам серии. Выбор стандарта зависит от в первую очередь от роли пользователя и его задач при работе с информационной системой.

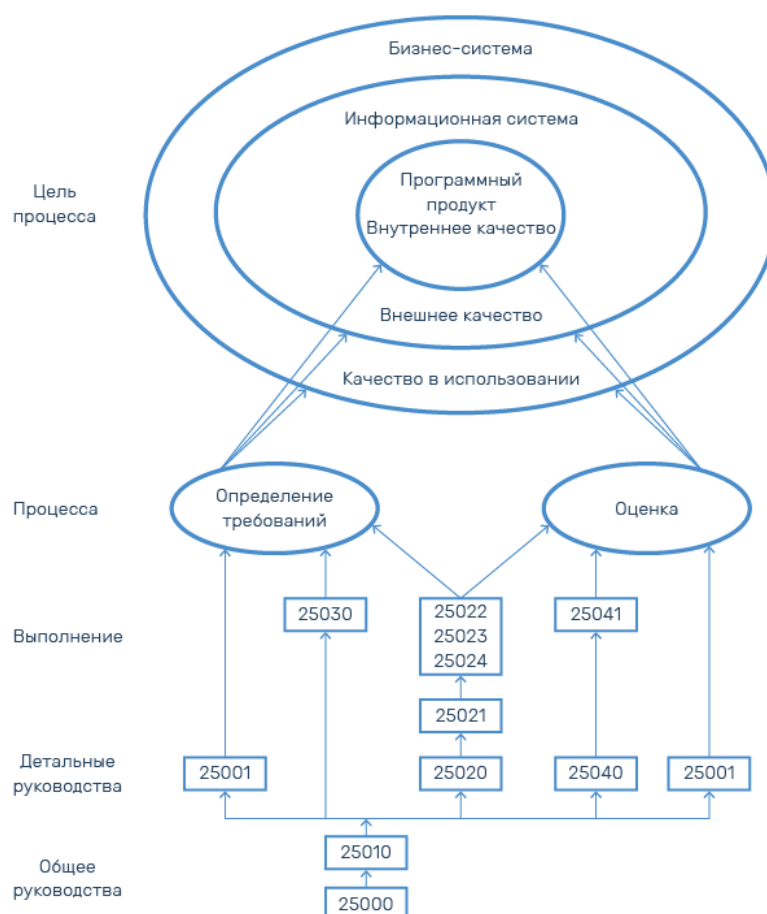


Рисунок 2 – Общая эталонная модель серии SQuaRE

Иерархическая структура моделей качества, которые содержатся в стандартах серии, определяет модель качества как набор характеристик, которые в свою очередь представлены подхарактеристиками. Характеристики и

подхарактеристики определяются атрибутами (свойствами) качества. Иерархическая структура моделей качества серии SQuaRE приведена на рисунке 3.

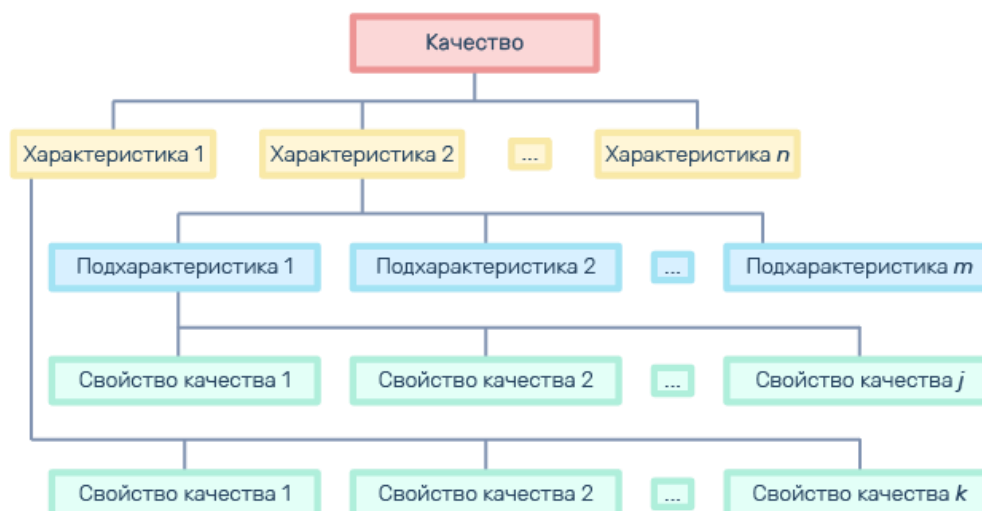


Рисунок 3 – Структура моделей качества по стандартам серии SQuaRE

Модель качества в ЖЦ ПС регламентирует взаимосвязи различных видов качества в жизненном цикле программных средств в соответствии с международным стандартом **ISO/IEC 25010:2011**.

Верхние уровни (характеристики и подхарактеристики) указанных моделей описаны в разделе стандартов *«Модели качества»*.

Для измерения значений атрибутов (свойств) качества используются меры качества, процессам измерения которых посвящены стандарты раздела *«Измерения качества»*.

Стандарт ISO/IEC 25000:2014 содержит рекомендации и руководства, с которыми стоит ознакомиться перед началом работы с конкретным стандартом серии.

Действие стандарта **ISO/IEC 25051:2014** распространяется уже на готовый продукт (RUSP). *В стандарте используется аббревиатура «RUSP» как сокращение от «Ready to Use Software Product», что собственно и обозначает «готовый к использованию программный продукт».*

Настоящий стандарт устанавливает:

- требования к RUSP;
- требования к документации по тестированию RUSP включая план тестирования, описание тестов и их результатов;
- инструкции по оценке соответствия RUSP.

В стандарт также входят рекомендации в отношении RUSP, критически важные для безопасности и ведения бизнеса. Настоящий стандарт предназначен исключительно для того, чтобы обеспечить пользователям уверенность в соответствии предлагаемых свойств RUSP их требованиям. Стандарт не относится к реализации продукции (*включая мероприятия по продаже и промежуточные продукты, например спецификации*). Система обеспечения качества поставщика не входит в область применения настоящего стандарта.

Структура требований к программным продуктам в соответствии со стандартом ISO/IEC 25051:2014 приведена на рисунке 4.

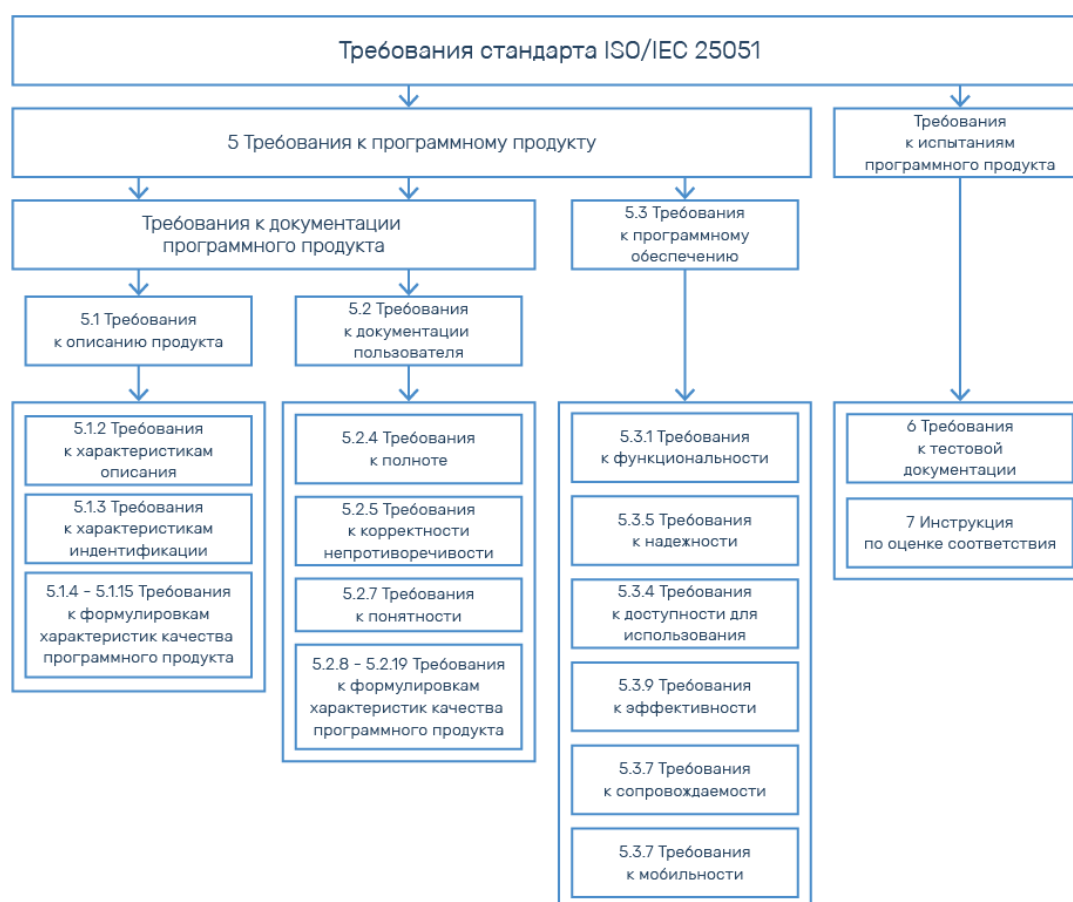


Рисунок 4 – Структура требований стандарта ISO/IEC 25051:2014

2.1. ISO/IEC 2501n (Модели качества)

В серии SQuaRE определены следующие модели качества:

- модели качества систем и программных средств;
- модель качества данных;
- модель качества ИТ-услуг.

Соответственно, эти модели описаны в стандартах группы **«Модели качества» (ISO/IEC 2501n)**:

- ISO/IEC 25010:2011 — System and software quality models.
- ISO/IEC TS 25011:2017 — Service quality models.
- ISO/IEC 25012:2008 — Data quality model.

Стандарт ISO/IEC 25010 (модели качества систем и программных средств) введен взамен ISO/IEC 9126.

Модели качества обеспечивают основу для сбора требований заинтересованных сторон.

Область применения моделей качества включает в себя спецификацию поддержки и оценку программного обеспечения и преимущественно программных вычислительных систем с разных точек зрения, которые связаны с их приобретением, требованиями, разработкой, использованием, оценкой, поддержкой, обслуживанием, обеспечением качества и управлением им, а также менеджментом и аудитом.

Качество системы

Степень удовлетворения системой заявленных и подразумеваемых потребностей различных заинтересованных сторон, которая позволяет, таким образом, оценить достоинства.

Эти заявленные и подразумеваемые потребности представлены в международных стандартах серии SQuaRE посредством моделей качества, которые представляют качество продукта в виде *иерархической структуры*. На верхнем уровне этой структуры – *характеристики*, которые в отдельных случаях далее разделяются на *подхарактеристики*. Измеримые свойства системы, связанные с её качеством, называют *свойствами (атрибутами) качества*. Свойства качества связаны (ассоциированы) с мерами качества.

К настоящему времени в серии SQuaRE имеются **три модели качества**:

1. *модель качества при использовании* (определена в ISO/IEC 25010);
2. *модель качества продукта* (определена в ISO/IEC 25010);
3. *модель качества данных* (определена в ISO/IEC 25012).

Совместное использование моделей качества дает основание считать, что учтены все характеристики качества.

Качество при использовании

Степень, в которой продукт или система могут использоваться конкретными пользователями для достижения определенных целей с эффективностью, производительностью, свободой от риска и удовлетворенностью в конкретных условиях использования для удовлетворения их потребностей.

Модель качества при использовании определяет пять характеристик, связанных с результатами взаимодействия с системой:

1. **Эффективность**, или результативность (**Effectiveness**) – точность и полнота, с которой пользователи достигают определенных целей.
2. **Производительность** (**Efficiency**) – связь точности и полноты достижения пользователями целей с израсходованными ресурсами.
3. **Удовлетворенность** (**Satisfaction**) – степень, с которой удовлетворены пользовательские потребности, когда продукт или система применяются в указанной среде использования.
4. **Свобода от риска** (**Freedom from risk**) – степень, с которой продукт или система смягчают потенциальный риск экономических потерь, человеческой жизни, здоровью или окружающей среде.
5. **Покрытие контекста** (**Context coverage**) – степень, в которой продукт или система могут быть применены с результативностью, эффективностью, свободой от риска и удовлетворением в определенных средах использования и в средах, вне первоначально явно идентифицированных.

На рисунке 1 представлена структура модели качества в использовании.

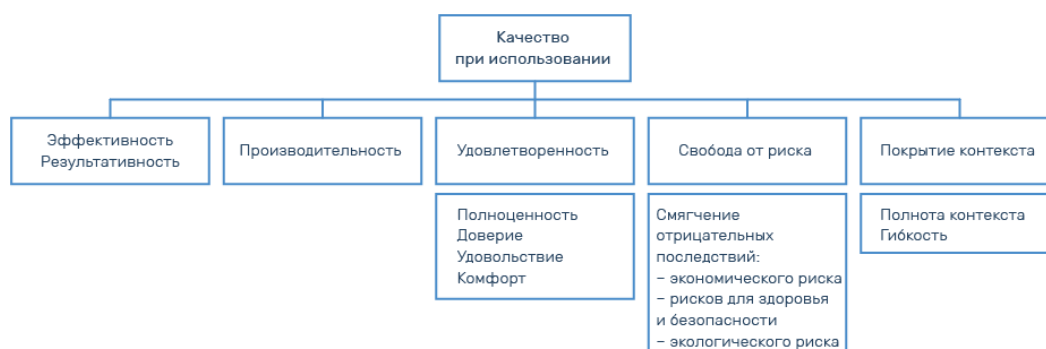


Рисунок 1 – Структура модели качества при использовании

Каждая характеристика применима для различных видов деятельности заинтересованных лиц, например, для взаимодействия оператора или поддержки разработчика.

Характеристики «Эффективность» и «Производительность» не разделяются на подхарактеристики. Их значения определяются как значения соответствующих мер качества.

В характеристике «Удовлетворенность» выделяют следующие подхарактеристики:

- **Доверие** (Trust) – степень, с которой пользователь или другое заинтересованное лицо уверены, что продукт или система будут вести себя как предназначено.
- **Удовольствие** (Pleasure) – степень, с которой пользователь получает удовольствие от использования продукта для личных нужд.
- **Комфорт** (Comfort) – степень, с которой пользователь удовлетворен физическим комфортом.

Подхарактеристиками «Свободы от риска» являются:

- **Смягчение отрицательных последствий экономического риска** (Economic risk mitigation) – степень, с которой продукт или система смягчают потенциальный риск для финансового положения, эффективной операции, коммерческой недвижимости, репутации или других ресурсов в намеченных средах использования.
- **Смягчение отрицательных последствий рисков для здоровья и безопасности** (Health and safety risk mitigation) – степень, с которой продукт или система смягчают потенциальный риск для людей в намеченных средах использования.
- **Смягчение отрицательных последствий экологического риска** (Environmental risk mitigation) – степень, с которой продукт или система смягчают потенциальный риск для собственности или экологии окружающей среды в намеченных средах использования.

Характеристика «Покрытие контекста» включает следующие подхарактеристики:

- **Полнота контекста** (Context completeness) – степень, с которой продукт или система могут быть применены с результативностью, эффективностью, свободой от риска и удовлетворением во всех указанных средах использования.
- **Гибкость** (Flexibility) – степень, с которой продукт или система могут быть применены с результативностью, эффективностью, свободой от риска и удовлетворением в средах, первоначально не определенных в требованиях.

Модель качества продукта применяется к целевой компьютерной системе и, связанному с ним, целевому программному обеспечению или целевому программному продукту. Целевая система может включать компьютерную технику,

нецелевые программные продукты, нецелевые данные, и целевые данные, которые являются предметом модели качества данных.

На рисунке 2 представлена структура модели качества продукта.



Рисунок 2 – Структура модели качества продукта

Модель качества продукта сводит свойства качества программного продукта (или системы) к восьми характеристикам:

1. **Функциональная пригодность** (*Functional suitability*) – степень, с которой продукт обеспечивает выполнение функций, которые удовлетворяют заявленным и подразумеваемым потребностям при использовании в указанных условиях.

2. **Уровень производительности**, или **Эффективность производительности** (*Performance efficiency*) – выполненная работа относительно количества ресурсов при установленных условиях.

3. **Совместимость** (*Compatibility*) – степень, с которой продукт, система или их компоненты могут обмениваться информацией с другими продуктами, системами или компонентами, и/или выполнить необходимые функции, разделяя те же самые аппаратные средства или окружающую среду программного обеспечения.

4. **Удобство пользования** (*Usability*) – степень, с которой продукт или система могут использоваться указанными пользователями для достижения определенных целей с результативностью, эффективностью и удовлетворением в указанном контексте использования.

5. **Надежность** (*Reliability*) – степень, с которой система, продукт или их компоненты выполняют определенные функции при указанных условиях в течение установленного периода времени.

6. **Защищенность**, или **Безопасность** (*Security*) – степень, с которой продукт или система защищают информацию и данные так, чтобы у пользователей или

других продуктов или систем была степень доступа к данным, соответствующего их типам и уровням разрешения.

7. **Сопровождаемость** (*Maintainability*) – степень результативности и эффективности, с которой продукт или система могут быть изменены специалистами по обслуживанию.

8. **Переносимость**, или **Мобильность** (*Portability*) – степень результативности и эффективности, с которой система, продукт или компонент могут быть перенесены на другие аппаратные средства, программное обеспечение или в другую эксплуатационную или окружающую среду использования.

Каждая характеристика, в свою очередь, состоит из ряда соответствующих подхарактеристик.

1) характеристика «Функциональная пригодность» включает в себя подхарактеристики:

- **Функциональная полнота** (*functional completeness*): степень покрытия совокупностью функций всех определенных задач и целей пользователя.
- **Функциональная корректность** (*functional correctness*): степень обеспечения продуктом или системой необходимой степени точности корректных результатов.
- **Функциональная целесообразность** (*functional appropriateness*): степень функционального упрощения выполнения определенных задач и достижения целей.

2) характеристика «Эффективность производительности» включает в себя подхарактеристики:

- **Временные характеристики** (*time behaviour*): степень соответствия требованиям по времени отклика, времени обработки и показателей пропускной способности продукта или системы
- **Использование ресурсов** (*resource utilization*): степень удовлетворения требований по потреблению объемов и видов ресурсов продуктом или системой при выполнении их функций.
- **Потенциальные возможности** (*capacity*): степень соответствия требованиям предельных значений параметров продукта или системы.

3) характеристика «Совместимость» включает в себя подхарактеристики:

- **Сосуществование (совместимость)** (*co-existence*): способность продукта совместно функционировать с другими независимыми продуктами в общей среде с разделением общих ресурсов и без отрицательного влияния на любой другой продукт.

- **Функциональная совместимость (интероперабельность)** (*interoperability*): способность двух или более систем, продуктов или компонент обмениваться информацией и использовать такую информацию.

4) характеристика «Удобство пользования» включает в себя подхарактеристики:

- **Определимость пригодности** (*appropriateness recognizability*): возможность пользователей понять, подходит ли продукт или система для их потребностей, сравним ли с функциональной целесообразностью (*functional appropriateness*).
- **Изучаемость** (*learnability*): возможность использования продукта или системы определенными пользователями для достижения конкретных целей обучения для эксплуатации продукта или системы с эффективностью, результативностью, свободой от риска и в соответствии с требованиями в указанном контексте использования.
- **Управляемость** (*operability*): наличие в продукте или системе атрибутов, обеспечивающих простое управление и контроль.
- **Эстетика пользовательского интерфейса** (*user interface aesthetics*): степень «приятности» и «удовлетворенности» пользователя интерфейсом взаимодействия с пользователем.
- **Доступность** (*accessibility*): возможность использования продукта или системы для достижения определенной цели.

5) характеристика «Надежность» включает в себя подхарактеристики:

- **Завершенность** (*maturity*): степень соответствия системы, продукта или компонента при нормальной работе требованиям надежности.
- **Готовность** (*availability*): степень работоспособности и доступности системы, продукта или компонента.
- **Отказоустойчивость** (*fault tolerance*): способность системы, продукта или компонента работать как предназначено, несмотря на наличие дефектов программного обеспечения или аппаратных средств.
- **Восстанавливаемость** (*recoverability*): способность продукта или системы восстановить данные и требуемое состояние системы в случае прерывания или сбоя.

6) характеристика «Безопасность» включает в себя подхарактеристики:

- **Конфиденциальность** (*confidentiality*): обеспечение продуктом или системой ограничения доступа к данным только для тех, кому доступ разрешен.
- **Целостность** (*integrity*): степень предотвращения системой, продуктом или компонентом несанкционированного доступа или модификации компьютерных программ или данных.

- **Неподдельность** (non-repudiation): степень, с которой может быть доказан факт события или действия таким образом, что этот факт не может быть отвергнут когда-либо позже.
- **Отслеживаемость** (accountability): степень, до которой действия объекта могут быть прослежены однозначно.
- **Подлинность** (authenticity): степень достоверности тождественности объекта или ресурса требуемому объекту или ресурсу.

7) характеристика «Сопровождаемость» включает в себя подхарактеристики:

- **Модульность** (modularity): степень представления системы или компьютерной программы в виде отдельных блоков таким образом, чтобы изменение одного компонента оказывало минимальное воздействие на другие компоненты.
- **Возможность многократного использования** (reusability): степень, в которой актив может быть использован в нескольких системах или в создании других активов.
- **Анализируемость** (analysability): степень простоты оценки влияния изменений одной или более частей на продукт или систему или простоты диагностики продукта для выявления недостатков и причин отказов, или простоты идентификации частей, подлежащих изменению.
- **Модифицируемость** (modifiability): степень простоты эффективного и рационального изменения продукта или системы без добавления дефектов и снижения качества продукта.
- **Тестируемость** (testability): степень простоты эффективного и рационального определения для системы, продукта или компонента критериев тестирования, а также простоты выполнения тестирования с целью определения соответствия этим критериям.

8) характеристика «Мобильность» включает в себя подхарактеристики:

- **Адаптируемость** (adaptability): степень простоты эффективной и рациональной адаптации для отличающихся или усовершенствованных аппаратных средств, программного обеспечения, других операционных сред или условий использования.
- **Устанавливаемость** (installability): степень простоты эффективной и рациональной, успешной установки и/или удаления продукта или системы в заданной среде.
- **Взаимозаменяемость** (replaceability): способность продукта заменить другой конкретный программный продукт для достижения тех же целей в тех же условиях.

Модель качества продукта можно применять как для программного продукта, так и для компьютерной системы, в состав которой входит программное обеспечение, поскольку большинство подхарактеристик применимо и к программному обеспечению, и к системам.

Применение моделей качества в жизненном цикле компьютерных систем в соответствии с международным стандартом ISO/IEC 25010:2011 представлено на рисунке 3.

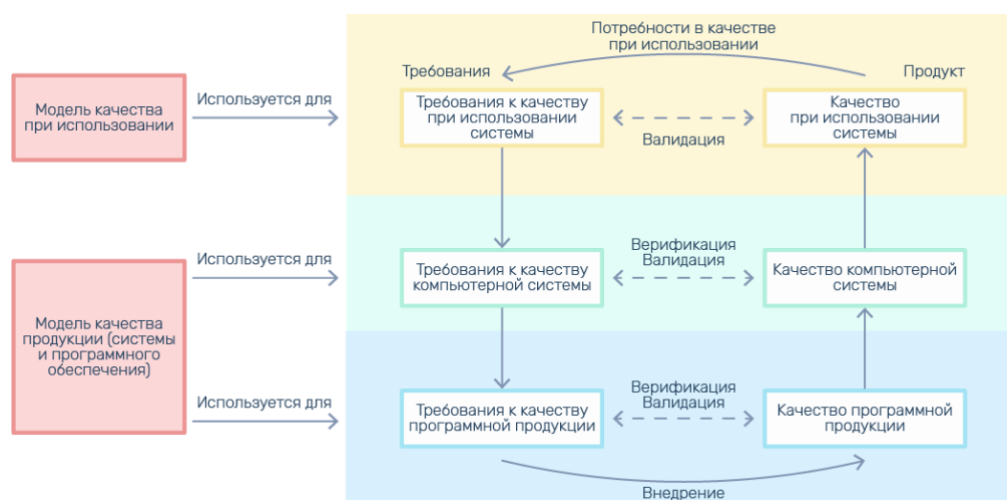


Рисунок 3 – Модели качества по ISO/IEC 25010 в жизненном цикле компьютерных систем

Модель качества данных представляет собой основу, на которой построена система оценки качества информационных продуктов. В модели качества данных устанавливаются основные характеристики качества данных, которые необходимо учитывать при оценке свойств предполагаемого информационного продукта.

Качество информационного продукта можно понимать как степень, в которой данные удовлетворяют требованиям, определенным организацией-владельцем продукта. В частности, эти требования отражаются в модели качества данных через ее характеристики: точность, полнота, согласованность, достоверность, актуальность, доступность и др.

Стандарт ISO/IEC 25012:2008 описывает модель качества данных компьютерной системы и определяет набор характеристик, которые формируют качество целевых данных. Под **целевыми данными** обозначены данные, которые организация решает подвергнуть анализу и аттестации (валидации) с помощью модели качества.

Модель качества данных, которая определена в международном стандарте ISO/IEC 25012, представляет собой двухуровневую структуру. Верхний уровень **состоит из 15 характеристик**, которые определяются непосредственно мерами качества данных.

Характеристики качества данных рассматриваются с двух точек зрения и на основании этого подразделяются на **две основные категории**:

1. **Собственное качество данных** (Inherent Data Quality);

2. Системно-зависимое качество данных (System-Dependent Data Quality).

Собственное качество данных относится к степени, в которой качественные характеристики данных обладают внутренним потенциалом для удовлетворения заявленных и подразумеваемых потребностей, когда данные используются в определенных условиях. С внутренней точки зрения качество данных относится к самим данным, в частности к таким составляющим: значения области данных и возможные ограничения (*например, бизнес-правила, регулирующие качество, требуемое для характеристики в данном приложении*); отношения значений данных (*например, согласованность*); метаданные.

Под системно-зависимым качеством данных (или *качеством данных, зависящим от системы*) понимается степень, в которой качество данных достигается и сохраняется в компьютерной системе, когда данные используются в определенных условиях. С этой точки зрения качество данных зависит от технологической области, в которой используются данные.

Это достигается за счет возможностей компонентов компьютерных систем, таких как: аппаратные устройства (*например, для предоставления данных или получения требуемой точности*); программное обеспечение компьютерной системы (*например, программное обеспечение резервного копирования для обеспечения возможности восстановления*); другое программное обеспечение (*например, инструменты миграции на добиться переносимости*).

Общая структура данной модели в разрезе значимости ее характеристик для каждой из точек зрения представлена на рисунке 4.

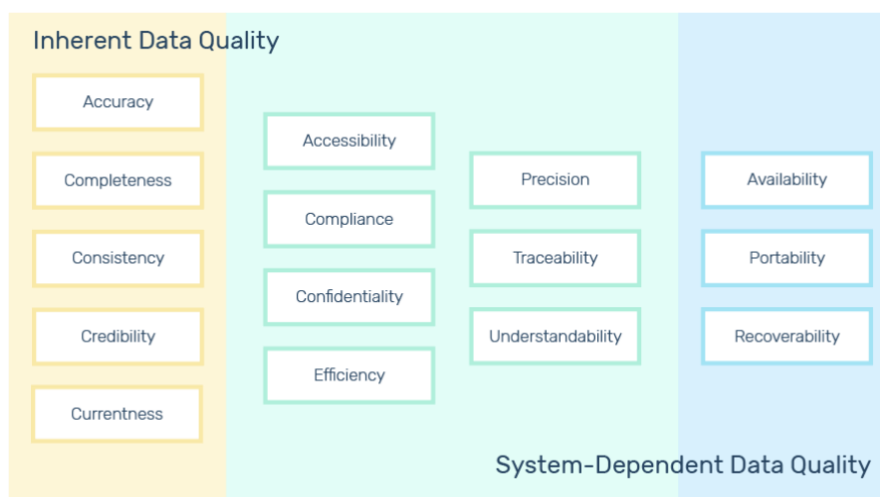


Рисунок 4 – Структура модели качества данных по ISO/IEC 25012:2008.

Взаимосвязь моделей качества и компонентов компьютерных систем, которые описаны в серии SQuaRE, представлена на рисунке 5.

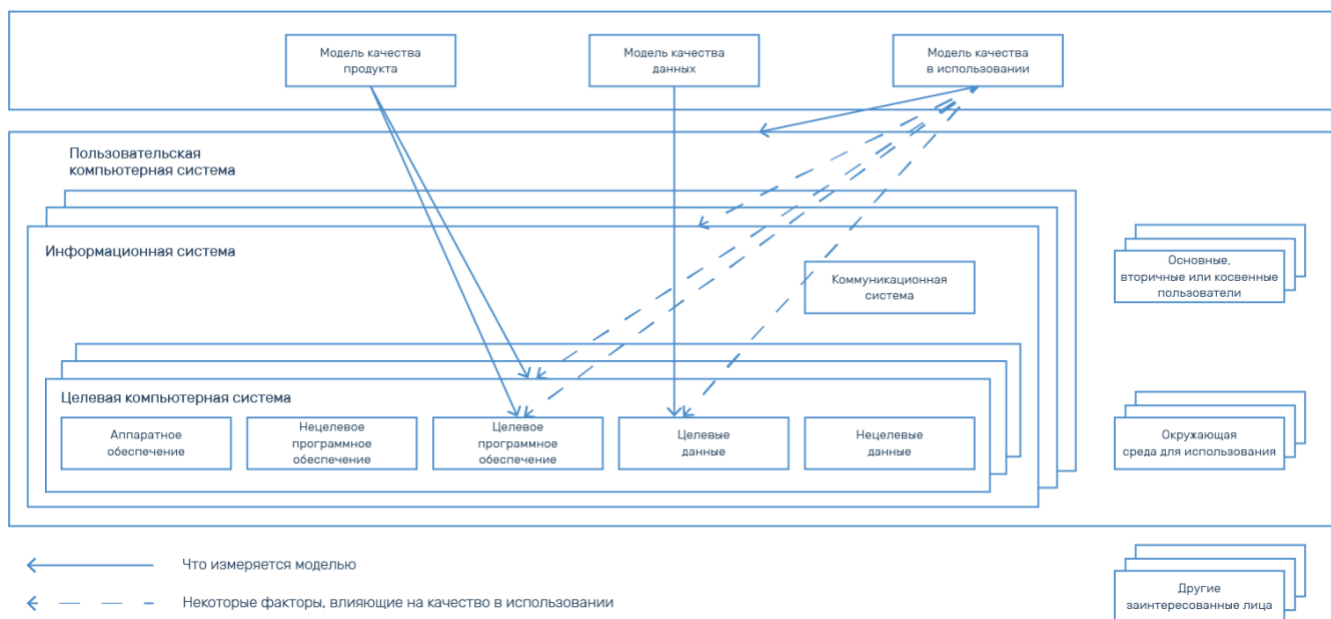


Рисунок 5 – Цели моделей качества серии SQaRE (ISO/IEC 2501n)

2.2. ISO/IEC 25020 (Связь качества программного средства с его жизненным циклом)

В течение всего жизненного цикла программного средства его качество изменяется.

И необходимо понимать, что *требуемое качество*, которое определено в начале жизненного цикла, отличается от *фактического качества* поставленного продукта. Существует несколько точек зрения на качество программного средства в течение его жизненного цикла, от выбора которых зависит как оценка качества, так и управление качеством на каждой стадии жизненного цикла.

Стандарт ISO/IEC 25020 содержит описание модели качества в жизненном цикле программных средств, которая построена на основе модели качества, определенной еще в стандартах ISO/IEC 9126 и ISO/IEC 14598. Данная модель отражает виды качества в жизненном цикле программных средств. В зависимости от стадии жизненного цикла выделены следующие основные **виды качества программных средств**: *внутреннее, внешнее и качество в использовании компьютерных систем*.

Качество программного продукта может быть оценено, измеряя внутренние свойства (*статические меры промежуточных продуктов*), или измеряя внешние

свойства (*как правило, измеряя поведение программного кода при его выполнении*), или измеряя качество в свойствах использования (*когда продукт находится в реальном или моделируемом употреблении*).

Пользовательские потребности в качестве включают требования по системному качеству в использовании в определенных контекстах использования. Эти идентифицированные потребности могут использоваться, определяя внешние и внутренние меры качества, используя характеристики и подхарактеристики качества программного продукта.

Зависимость и влияние видов качества в жизненном цикле программного продукта представлена на рисунке 1.

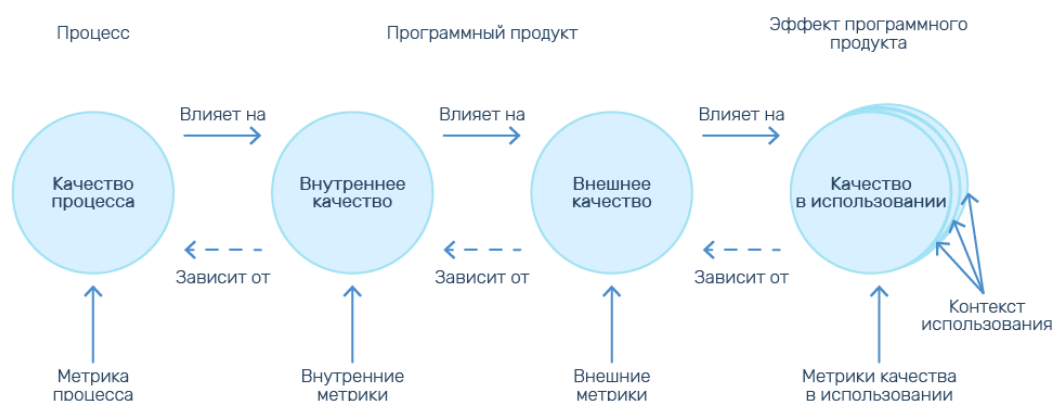


Рисунок 1 – Схема взаимосвязи видов качества программного продукта

Из рисунка 1 видно, что:

- улучшение качества процесса способствует улучшению качества продукта. *Под процессом понимается любой из процессов жизненного цикла, которые определены в стандарте ISO/IEC 12207.*
- улучшение качества продукта способствует улучшению качества в использовании компьютерной системы.

Соответственно, оценка и улучшение процесса являются средством улучшения качества продукта, а оценивание и улучшение качества продукта средство улучшения качества в использовании системы.

Аналогично:

- уже оцененное *качество в использовании системы* может обеспечить обратную связь, чтобы улучшить продукт;
- в свою очередь, оценка продукта может обеспечить обратную связь для улучшения *процесса*.

Обеспечение внутреннего качества программного продукта, предпосылка для достижения необходимого внешнего поведения, обеспечение внешнего качества, предпосылка для достижения качества в использовании.

Иллюстрация изменений и взаимосвязи различных видов качества в жизненном цикле программных средств в соответствии с международным стандартом **ISO/IEC 25020** представлено на рисунке 2.

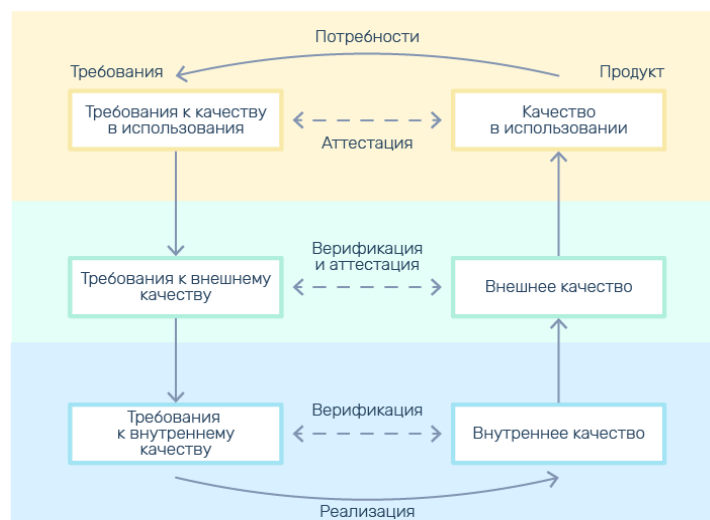


Рисунок 2 – Качество в жизненном цикле программного продукта

Требования к внутреннему качеству определяют требуемый уровень качества с внутренней точки зрения на программный продукт.

Требования к внешнему качеству определяют требуемый уровень качества с внешней точки зрения.

Потребности пользователя в качестве (требования к качеству в использовании) определяются как требования к качеству, выраженные в терминах мер качества в использовании, внешних и иногда внутренних мер.

Внутреннее качество

Совокупность характеристик программного продукта с внутренней точки зрения; внутреннее качество измеряется с помощью внутренних метрик и оценивается по отношению к требованиям к внутреннему качеству.

Внешнее качество

Совокупность характеристик программного продукта с внешней точки зрения; это качество, измеряемое и оцениваемое на основе внешних метрик при выполнении программного продукта во время тестирования (испытаний) в моделируемой среде с моделируемыми данными или во время эксплуатации.

Качество в использовании

Качество программного продукта, применяемого в заданной среде и заданном контексте использования, с точки зрения пользователя; пользователь оценивает только те атрибуты программного продукта, которые он применяет в своих задачах.

Качество в жизненном цикле программного продукта можно рассматривать в трех основных стадиях:

- внутреннее качество программного обеспечения на стадии разработки;
- внешнее качество программного обеспечения на стадии тестирования;
- качество в использовании на стадии применения программного продукта.

Внешние оценки качества могут использоваться, чтобы предсказать системное качество в использовании.

2.3. ISO/IEC 25040 (Процесс оценки качества)

Международный стандарт **ISO/IEC 25040:2011** (Процесс оценки) серии SQuaRE предоставляет описание процесса для оценки качества программного продукта и устанавливает требования к применению этого процесса.

Стандарт предназначен для использования разработчиками, заказчиками и независимыми оценщиками систем и программных средств.

Стандарт ISO/IEC 25040:2011 заменяет собой стандарт ISO/IEC 14598.

В стандарте определена обобщенная эталонная модель процесса оценки качества программного продукта. На рисунке 1 представлена функциональная диаграмма данной модели, которая отражает входные и выходные данные, ограничения и ресурсы для процесса оценки качества программного продукта.

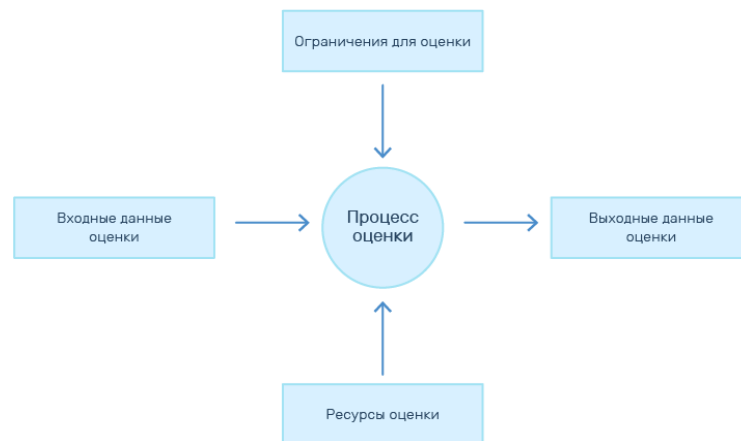


Рисунок 1 – Общий вид оценки качества программного продукта

- *Ограничения, которые накладываются на процесс оценки качества программного продукта*, могут включать в себя следующие моменты:

- a) конкретные потребности пользователя;
- b) ресурсы;
- c) расписание;
- d) стоимость;
- e) среду;
- f) инструменты и методологию;
- g) создание отчетов.

- *Ресурсы для процесса оценки качества* программного продукта могут включать в себя:

a) применимые измерительные инструменты и методологию, включая модули оценки;

b) применимые документы SQuaRE (ИСО/МЭК 25001, ИСО/МЭК 25010, ИСО/МЭК 2502n, ИСО/МЭК 25030, ИСО/МЭК 25041, ИСО/МЭК 25042);

- c) человеческие ресурсы для оценки качества программного продукта;
- d) экономические ресурсы для оценки качества программного продукта;
- e) информационную систему для оценки качества программного продукта;
- f) базу знаний для оценки качества программного продукта.

- *Входные и выходные данные* зависят от целей оценки (различные роли исполнителя процесса имеют различные цели) и от этапа процесса (какие работы и задачи выполняются).

Эталонная модель оценки качества программного продукта предполагает, что оценка должна выполняться на основе спецификаций требований к качеству программного продукта с использованием стандарта ИСО/МЭК 25030. При этом в процессе оценки задействованы и другие стандарты SQuaRE – модели качества по ИСО/МЭК 25010 и меры качества по ИСО/МЭК 2502n.

Процесс оценки можно использовать для различных целей и подходов. Процесс можно использовать для оценки качества ранее разработанного программного обеспечения, готового к использованию программного обеспечения или разработанного по заказу, и можно использовать как в процессе, так и по завершении разработки.

Качество программного продукта может быть оценено в пределах определенной структуры качества на всех этапах жизненного цикла, относящихся к процессу разработки программного обеспечения и процессу приобретения и определенных в процессах жизненного цикла программного обеспечения ИСО/МЭК 12207 и процессах жизненного цикла систем ИСО/МЭК 15288.

Эталонная модель описывает процесс и детализирует действия и задачи, обеспечивающие их цели, а также предоставляет дополнительную методическую информацию, которую допускается использовать для проведения оценки качества программного продукта. Эталонная модель процесса оценки по ISO/IEC 25040:2011 состоит из пяти действий (работ), каждое из которых содержит набор задач (рисунок 2).

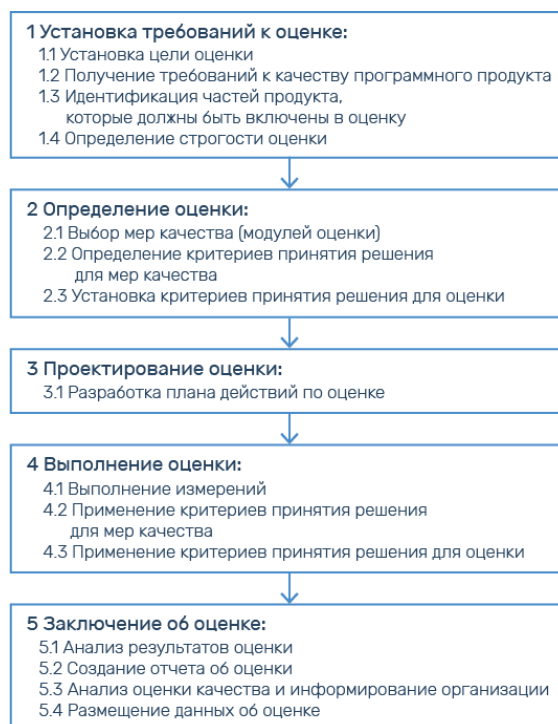


Рисунок 2 – Процесс оценки качества программного продукта по ISO/IEC 25040

Каждое действие в оценке качества программного продукта должно быть задокументировано. Протокол должен включать подробный отчет о действиях, выполняемых оценщиком при реализации плана оценки качества программного продукта, включая любые промежуточные данные, на которых базируется любая интерпретация. Протокол должен содержать исчерпывающую информацию и организован таким образом, чтобы обеспечить управление оценкой качества программного продукта и позволить повторно обрабатывать результаты оценки.

Обязательным результатом всего процесса оценки должен быть отчет об оценке качества программного продукта, в котором должны быть записаны мероприятия по оценке и результаты этой оценки.

На функциональных диаграммах (рисунки 3–7) представлены входная и выходная информация, ограничения и ресурсы для каждого этапа процесса оценки (для каждой работы).



Рисунок 3 – Диаграмма определения требований к оценке по ISO/IEC 25040



Рисунок 4 – Диаграмма определения оценки по ISO/IEC 25040

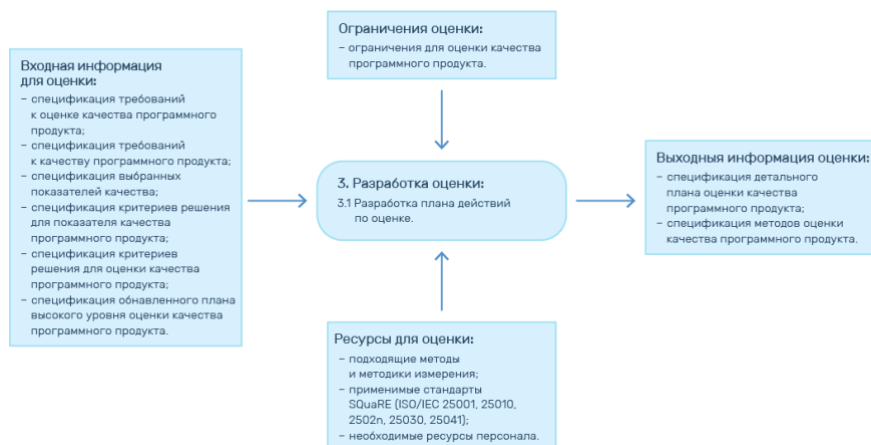


Рисунок 5 – Диаграмма проектирования оценки по ISO/IEC 25040



Рисунок 6 – Диаграмма выполнения оценки по ISO/IEC 25040

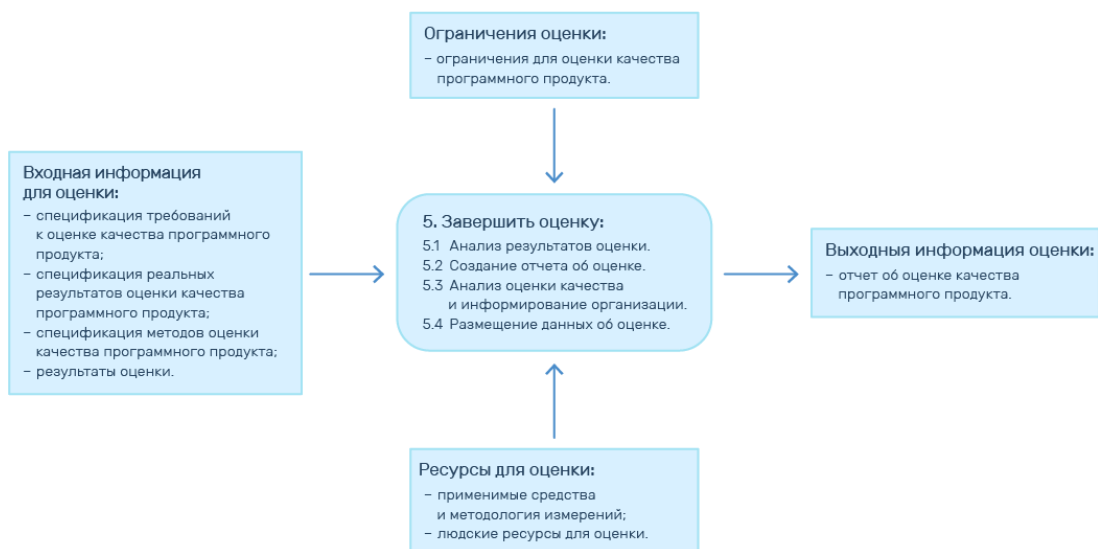


Рисунок 7 – Диаграмма заключения об оценке по ISO/IEC 25040

Таким образом, оценщик должен реализовать действия и задачи в соответствии с применимыми организационными политиками и процедурами, относящимися к процессу оценки качества программного продукта (рисунок 2).

Дополнительный материал. Стандарт ГОСТ Р ИСО/МЭК 9126-93

1. Стандарт ГОСТ Р ИСО/МЭК 9126-93

ГОСТ Р ИСО/МЭК 9126-93 — «Информационная технология. ОЦЕНКА ПРОГРАММНОЙ ПРОДУКЦИИ. Характеристики качества и руководства по их применению».

Стандарт подготовлен на основе применения аутентичного текста международного стандарта ISO/IEC 9126-91 "*Information technology. Software product evaluation. Quality characteristics and guidelines for their use*" (переиздание – ноябрь 2004 г.).

2. Область применения

Настоящий стандарт определяет шесть характеристик, которые с минимальным дублированием описывают качество программного обеспечения (ПО). Данные характеристики образуют основу для дальнейшего уточнения и описания качества ПО. Руководства описывают использование характеристик качества для оценки качества ПО.

Настоящий стандарт не определяет подхарактеристики (комплексные показатели) и показатели, а также методы измерения, ранжирования и оценки. Данный стандарт придерживается определения качества по ИСО 8402.

Определения характеристик и соответствующая модель процесса оценки качества, приведенные в настоящем стандарте, применимы тогда, когда определены требования для программной продукции и оценивается ее качество в процессе ЖЦ.

Эти характеристики могут применяться к любому виду ПО, включая программы и данные, входящие в программно-технические средства (встроенные программы).

Настоящий стандарт предназначен для характеристик, связанных с приобретением, разработкой, эксплуатацией, поддержкой, сопровождением или проверкой ПО.

3. Применяемость стандарта

Настоящий стандарт применяется для установления требований к качеству ПО и оценивания (измерения, ранжирования и оценки) программных продуктов, включая:

- определение требований к качеству программной продукции;

- оценивание технических требований к ПО при контроле за тем, чтобы требования качества были удовлетворены в процессе разработки;
- описание признаков и свойств (атрибутов) внедренного ПО (например, в руководствах пользователя);
- оценивание разработанного ПО перед его поставкой;
- оценивание ПО перед приемкой.

Существуют только несколько общепринятых метрик для характеристик, описанных в настоящем стандарте. Организации и группы по стандартизации могут устанавливать свои собственные модели процесса оценивания и методы формирования и проверки метрик, связанных с этими характеристиками, для охвата различных областей применения и стадии ЖЦ. В тех случаях, когда соответствующие метрики отсутствуют и не могут быть разработаны, иногда пользуются словесными описаниями или "приблизительными методами".

При использовании шести характеристик качества в целях описания и оценивания также необходимо установить уровни ранжирования и критерии конкретно для данной организации или для данного применения, или для того и другого.

Должны быть установлены метрики, уровни ранжирования и критерии применительно к оценке качества, когда обмениваются результатами оценивания.

Хотя отсутствует общепринятая система классификации ПО, имеется несколько общепринятых классов ПО. Важность каждой характеристики качества меняется в зависимости от класса ПО.

***Например,** надежность наиболее важна для ПО боевых критичных систем, эффективность наиболее важна для ПО критичных по времени систем реального времени, а практичность наиболее важна для ПО диалога конечного пользователя.*

Важность каждой характеристики качества также меняется в зависимости от принятых точек зрения.

4. Представления о качестве ПО

Имеется несколько представлений о качестве:

• **Представление пользователя**

Определение качества по ИСО 8402 отражает представление пользователя так же, как и характеристики, определенные в настоящем стандарте. Пользователи в основном проявляют заинтересованность в применении ПО, его производительности и результатах использования. Пользователи оценивают ПО без изучения его внутренних аспектов или того, как ПО создавалось.

Пользователя могут интересовать следующие вопросы:

- Имеются ли требуемые функции в ПО?
- Насколько надежно ПО?
- Насколько эффективно ПО?
- Является ли ПО удобным для использования?
- Насколько просто переносится ПО в другую среду?

Представление пользователя должно также включать представление о характеристиках качества, требуемое тем, кто сопровождает ПО.

• **Представление разработчика**

Процесс создания требует от пользователя и разработчика использования одних и тех же характеристик качества ПО, так как они применяются для установления требований и приемки. Когда разрабатывается ПО для продажи, в требованиях качества должны быть отражены предполагаемые потребности.

Так как разработчики отвечают за создание ПО, которое должно удовлетворять требованиям качества, они заинтересованы в качестве промежуточной продукции так же, как и в качестве конечной продукции. Для того, чтобы оценить качество промежуточной продукции на каждой фазе цикла разработки, разработчики должны использовать различные метрики для одних и тех же характеристик, потому что одни и те же метрики неприменимы для всех фаз ЖЦ.

***Например,** пользователь понимает эффективность в терминах времени реакции, тогда как разработчик использует в проектной спецификации термины длины маршрута и времени ожидания и доступа.*

Метрики, применяемые для внешнего интерфейса продукции, заменимы метриками, применяемыми для ее структуры.

• **Представление руководителя**

Руководитель может быть более заинтересован в общем качестве, чем в конкретной характеристике качества, и по этой причине будет нуждаться в определении важности значений, отражающих коммерческие требования для индивидуальных характеристик.

Руководителю может также потребоваться сопоставление повышения качества с критериями управляемости, такими как плановая задержка или перерасход стоимости, потому что он желает оптимизировать качество в пределах ограниченной стоимости, трудовых ресурсов и установленного времени.

5. Модель процесса оценивания

Схема 1 отражает основные этапы, требуемые для оценивания качества ПО, начиная с характеристик качества, определенных в настоящем стандарте. Ряд детальных процедур, таких как анализ и проверка метрик, на схеме 1 не показаны.



Схема 1 – Процесс оценивания качества ПО

Данный процесс может применяться в любой подходящей фазе ЖЦ для каждого компонента программной продукции.

Процесс состоит из трех стадий: установление (определение) требований к качеству, подготовка к оцениванию и процедура оценивания.

Целью начальной стадии является установление требований в терминах характеристик качества и возможных комплексных показателей (подхарактеристик). Требования выражают потребности внешнего окружения для рассматриваемой программной продукции и должны быть определены до начала разработки. Так как программная продукция разделяется на основные компоненты, требования для продукции в целом могут отличаться от требований для отдельных компонентов.

Целью второй стадии является подготовка основы для оценивания.

Последняя стадия модели процесса оценивания уточняется по трем этапам, называемым "измерение", "ранжирование" и "оценка".

Для измерения выбранные метрики применяются к программной продукции. Результатом являются значения в масштабах метрик.

На этапе ранжирования устанавливается уровень ранжирования для измеренного значения.

Оценка является последним этапом процесса оценивания ПО, на котором обобщается множество установленных уровней. Результатом является заключение о качестве программной продукции. Затем обобщенное качество сравнивается с другими факторами, такими, как время и стоимость. Окончательное решение руководства принимается на основе критерия управляемости. Результатом является решение руководства по приемке или отбраковке, или по выпуску или невыпуску программной продукции.

6. Приложение А. Комплексные показатели (подхарактеристики) качества

Данное приложение представляет иллюстративную качественную модель, которая определяет характеристики из настоящего стандарта в терминах комплексных показателей (подхарактеристик). Это является необходимым этапом в определении качества с использованием модели процесса оценивания качества, описанной в настоящем стандарте. Последующие соответствующие документы будут посвящены определению комплексных показателей.

Существует ряд подобных моделей качества, описанных в литературе и применяемых на практике. Степень завершенности этих моделей, терминов в определении пока еще не позволяет включить их в стандарт. Однако они публикуются для поощрения их практического использования и накопления опыта для их дальнейшего уточнения. Ключевым моментом в данном вопросе должна быть модель качества, по крайней мере, на уровне комплексных показателей (подхарактеристик) программной продукции, необязательно в точном соответствии с формой, описанной в данном приложении.

Определение комплексных показателей качества

А.2.1 Функциональные возможности (Functionality)

А.2.1.1 Пригодность (Suitability)

Атрибут ПО, относящийся к наличию и соответствию набора функций конкретным задачам.

Примечание

Примерами соответствия является состав функций, ориентированных на задачу, из входящих в него подфункций и объемы таблиц.

А.2.1.2 Правильность (Accuracy)

Атрибуты ПО, относящиеся к обеспечению правильности или соответствия результатов или эффектов.

Примечание

Например, она включает необходимую степень точности вычисленных значений.

А.2.1.3 Способность к взаимодействию (Interoperability)

Атрибуты ПО, относящиеся к способности его взаимодействовать с конкретными системами.

Примечание

Способность к взаимодействию используется вместо совместимости для того, чтобы избежать возможной путаницы с взаимозаменяемостью (см. А.2.6.4).

А.2.1.4 Согласованность (Compliance)

Атрибуты ПО, которые заставляют программу придерживаться соответствующих стандартов или соглашений, или положений законов, или подобных рекомендаций.

А.2.1.5 Защищенность (Security)

Атрибуты ПО, относящиеся к его способности предотвращать несанкционированный доступ, случайный или преднамеренный, к программам и данным.

А.2.2 Надежность (Refiability)

А.2.2.1 Стабильность (Maturity)

Атрибуты ПО, относящиеся к частоте отказов при ошибках в ПО.

А.2.2.2 Устойчивость к ошибке (Fault tolerance)

Атрибуты ПО, относящиеся к его способности поддерживать определенный уровень качества функционирования в случаях программных ошибок или нарушения определенного интерфейса.

Примечание

Определенный уровень качества функционирования включает возможность отказобезопасности.

A.2.2.3 Восстанавливаемость (Recoverability)

Атрибуты ПО, относящиеся к его возможности восстанавливать уровень качества функционирования и восстанавливать данные, непосредственно поврежденные в случае отказа, а также к времени и усилиям, необходимым для этого.

A.2.3 Практичность (Usability)

A.2.3.1 Понятность (Understandability)

Атрибуты ПО, относящиеся к усилиям пользователя по пониманию общей логической концепции и ее применимости.

A.2.3.2 Обучаемость (Learnability)

Атрибуты ПО, относящиеся к усилиям пользователя по обучению его применению (например, оперативному управлению, вводу, выводу).

A.2.3.3 Простота использования (Operability)

Атрибуты ПО, относящиеся к усилиям пользователя по эксплуатации и оперативному управлению.

A.2.4 Эффективность (Efficiency)

A.2.4.1 Характер изменения во времени (Time behavior)

Атрибуты ПО, относящиеся к временам отклика и обработки и к скоростям выполнения его функций.

A.2.4.2 Характер изменения ресурсов (Resource behavior)

Атрибуты ПО, относящиеся к объему используемых ресурсов и продолжительности такого использования при выполнении функции.

A.2.5 Сопровождаемость (Maintainability)

A.2.5.1 Анализируемость (Analysability)

Атрибуты ПО, относящиеся к усилиям, необходимым для диагностики недостатков или случаев отказов или определения составных частей для модернизации.

A.2.5.2 Изменяемость (Changeability)

Атрибуты ПО, относящиеся к усилиям, необходимым для модификации, устранению отказа или для изменения условий эксплуатации.

А.2.5.3 Устойчивость (Stability)

Атрибуты ПО, относящиеся к риску от непредвиденных эффектов модификации.

А.2.5.4 Тестируемость (Testability)

Атрибуты ПО, относящиеся к усилиям, необходимым для проверки модифицированного ПО.

Примечание

Значения этой подхарактеристики могут быть изменены рассматриваемыми модификациями.

Примечание

1 Взаимозаменяемость используется вместо совместимости для того, чтобы избежать возможной путаницы со способностью к взаимодействию (см. А.2.1.3).

2 Взаимозаменяемость с конкретным ПС не предполагает, что данное средство заменимо рассматриваемым ПС.

3 Взаимозаменяемость может включать атрибуты простоты внедрения и адаптируемости. Понятие было введено в качестве отдельной подхарактеристики из-за его важности.