|  |
| --- |
|  |
| |  | | --- | |  | | МИНОБРНАУКИ РОССИИ | | Федеральное государственное бюджетное образовательное учреждение  высшего образования  **"МИРЭА - Российский технологический университет"**  **РТУ МИРЭА** | | Институт информационных технологий (ИТ) | | Кафедра инструментального и прикладного программного обеспечения (ИППО) | |
|  |
|  |

|  |
| --- |
| **ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №1** |
| **по дисциплине** |
| **Оценка качества ПО**  **на тему**  «Исследование стандартов в области качества программного обеспечения» |

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
| Выполнили студенты группы ИКБО-02-15 | *Апальков П.Ю.*  *Титов Д.А.* |
|  |  |
| Принял | *Чехарин Е.Е., старший преподаватель* |

|  |  |  |
| --- | --- | --- |
| Работа выполнена | «\_\_\_» \_\_\_\_\_\_\_\_ 201 г. |  |
|  |  |  |
| «Зачтено» | «\_\_\_»\_\_\_\_\_\_\_\_\_201 г. |  |

Москва 2018

# Содержание

[Введение 3](#_Toc531860617)

[1. Стандарт ISO/IEC 9126 3](#_Toc531860618)

[2. Модели качества 4](#_Toc531860619)

[3. Внутренние метрики 6](#_Toc531860620)

[4. Внешние метрики 8](#_Toc531860621)

[5. Стандарт ISO 25010 9](#_Toc531860622)

[6. Стандарт ISO/IEC 14598 10](#_Toc531860623)

[7. Модуль оценки 12](#_Toc531860624)

[8. Стандарт ISO 12207 13](#_Toc531860625)

[Заключение 14](#_Toc531860626)

[Список литературы 14](#_Toc531860627)

# 

# Введение

Важным источником требований являются стандарты, регламентирующие характеристики, функции и состав систем и программных средств, работающих в определенной предметной области. Такие стандарты создаются на основе опыта, накопленного в большом количестве проектов, в ходе которых такие системы создавались или развивались, и поэтому содержат ценную информацию о желательных характеристиках таких систем. Часто работа системы связана с выполнением каких-то действий, регулируемых существующим законодательством и нормами, действующими в данной области. Нормы и законы тоже являются разновидностью действующих стандартов, хотя обычно они формулируются менее четко и однозначно, чем технические стандарты и правила.

# 1. Стандарт ISO/IEC 9126

В течение десяти лет (с 1991 по 2001 г.) основой регламентирования характеристик качества ПС являлся международный стандарт ISO/IEC 9126:1991 – Информационная технология – Оценка программного продукта – Характеристики качества и руководства по их применению.

В дальнейшем стандарт ISO/IEC 9126:1991 заменен на две взаимосвязанные серии стандартов: ISO/IEC 9126–1–4 и ISO/IEC 14598–1–6.

Стандарт ISO/IEC 9126–1–4 регламентирует иерархическую модель качества программных средств. На верхнем уровне модели находятся характеристики.

Характеристики разделяются на подхарактеристики. Подхарактеристики определяются метриками. Метрики измеряют атрибуты (свойства) ПС.

**Пример**:

Характеристика – Функциональность.

Подхарактеристика – Пригодность.

Внутренняя метрика – Полнота функциональной реализации.

Внешняя метрика – Полнота функциональной реализации.

Данный стандарт состоит из четырех частей под общим названием Программная инженерия – Качество продукта

* 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.

# 2. Модели качества

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. Однако 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). В данной части стандарта регламентированы следующие виды метрик:

* внутренние метрики;
* внешние метрики;
* метрики качества в использовании.

Модель внутренних и внешних характеристик качества ПС состоит из шести групп базовых показателей, каждая из которых детализирована несколькими нормативными подхарактеристиками:

**1. Функциональность:**

1. пригодность;

2. корректность (правильность, точность);

3. способность к взаимодействию;

4. защищенность;

5. соответствие функциональности.

**2. Надежность:**

1. завершенности (отсутствие ошибок);

2. устойчивость к ошибке;

3. восстанавливаемость;

4. соответствие надёжности.

**3. Применимость(практичность):**

1. понятность;

2. обучаемость;

3. простота использования;

4. привлекательность;

5. соответствие практичности.

**4. Эффективность:**

1. поведение во времени;

2. использование ресурсов;

3. соответствие эффективности.

**5. Сопровождаемость:**

1. анализируемость;

2. изменяемость;

3. стабильность;

4. тестируемость;

5. соответствие сопровождаемости.

**6. Переносимость(мобильность):**

1. адаптируемостью;

2. настраиваемость;

3. совместимость;

4. взаимозаменяемость;

5. соответствие мобильности.

Модель качества в использовании является иерархической. На ее верхнем уровне находятся четыре характеристики. В отличие от модели внешнего и внутреннего качества уровень подхарактеристик в модели качества в использовании отсутствует.

Характеристики:

1. Результативность – способность программного продукта, позволяющая пользователям достигать заданные цели с точностью и полнотой в заданном контексте использования.

2. Продуктивность – способность программного продукта, позволяющая пользователям расходовать количество ресурсов, соответствующее результативности, достигаемой в заданном контексте использования. Ресурсы могут включать время выполнения задачи, усилия пользователя, материалы, стоимость использования.

3. Безопасность – способность программного продукта достигать приемлемых уровней риска причинения вреда людям, бизнесу, программному обеспечению, имуществу или окружающей среде в заданном контексте использования. Обычно риски – это результат дефектов в функциональности (включая защищенность), надежности, практичности и сопровождаемости.

4. Удовлетворенность – способность программного продукта удовлетворять пользователя в заданном контексте использования. Удовлетворенность определяется реакцией пользователя на взаимодействие с программным продуктом и включает отношение к применению продукта.

Совокупности метрик, перечисленные в частях 2-4 стандарта, являются рекомендуемыми, их набор не является исчерпывающим.

Метрики могут модифицироваться. Возможно применение метрик, не включенных в данные части. В этих частях стандарта содержатся пояснения к применению метрик, к типам шкал метрик и типам измерений, примеры метрик для каждой подхарактеристики, примеры применения метрик на протяжении ЖЦ ПС.

Во второй части стандарта ISO/IEC TR 9126–2:2003 определяются метрики количественного измерения внешнего качества ПС. Внешние метрики – это метрики, предназначенные для измерения качества программного продукта путем измерения поведения системы, частью которой является данный продукт. Внешние метрики могут использоваться в процессе эксплуатации и на стадиях тестирования или испытаний в процессах разработки и сопровождения ПС, когда уже созданы исполнимые коды программного продукта.

В третьей части стандарта ISO/IEC TR 9126–3:2003 определяются метрики количественного измерения внутреннего качества ПС. Внутренние метрики – это метрики, измеряющие собственные свойства ПС. Они измеряются в процессе разработки ПС на основе спецификации требований, результатов проектирования, исходного кода или другой документации ПС. Внутренние метрики дают возможность оценить качество промежуточных программных продуктов разработки, предсказывая качество конечного программного средства.

В четвертой части стандарта ISO/IEC TR 9126 4:2004 определяются метрики количественного измерения качества в использовании. Метрики качества в использовании – это метрики, измеряющие соответствие продукта потребностям заданных пользователей в достижении заданных целей с результативностью, продуктивностью, безопасностью и удовлетворением в заданных контекстах использования. Очевидно, что данные метрики могут использоваться только в процессе эксплуатации ПС в реальной среде окружения. Метрики качества в использовании основаны на измерении поведения типичных пользователей и системы, содержащей данное программное средство.

# 3. Внутренние метрики

Внутренние метрики функциональностипредназначены для предсказания того, удовлетворяет ли разрабатываемый программный продукт требованиям к функциональности и предполагаемым потребностям пользователя. Примеры метрик:

1) Полнота функциональной реализации, оценивает пригодность ПС.

2) Точность, оценивает правильность ПС.

3) Соответствие интерфейсов, оценивает способность к взаимодействию.

4) Предотвращение разрушения данных, оценивает защищенность.

5) Соответствие функциональности, оценивает соответствие функциональности.

Внутренние метрики надежностииспользуются во время разработки программного продукта для предсказания того, удовлетворяет ли программный продукт заявленным потребностям в надежности. Примеры метрик:

1) Полнота тестирования, оценивает завершенность.

2) Предотвращение некорректных действий, оценивает устойчивость к ошибке.

3) Способность к восстановлению, оценивает восстанавливаемость.

4) Соответствие надежности, оценивает соответствие надежности.

Внутренние метрики практичности используются во время разработки программного продукта для предсказания степени, в которой программный продукт может быть понят, изучен, управляем, привлекателен и соответствует договоренностям и руководствам по практичности

1) Способность к демонстрации, оценивает понятность.

2) Полнота документации пользователя, оценивает обучаемость.

3) Отменяемость действий пользователя, оценивает простоту использования.

4) Настраиваемость вида интерфейса пользователя, оценивает привлекательность.

5) Соответствие практичности, оценивает соответствие практичности.

Внутренние метрики эффективностииспользуются во время разработки программного продукта для предсказания эффективности поведения программного продукта во время тестирования или эксплуатации. Примеры метрик:

1) Пропускная способность, оценивает поведение во времени.

2) Использование памяти, оценивает использование ресурсов.

3) Соответствие эффективности, оценивает соответствие эффективности.

Внутренние метрики сопровождаемостииспользуются для предсказания уровня усилий, необходимых для модификации программного продукта. Примеры метрик:

1) Готовность диагностических функций, оценивает анализируемость.

2) Регистрируемость изменений, оценивает изменяемость.

3) Влияние изменений, оценивает стабильность.

4) Полнота встроенных функций тестирования, оценивает тестируемость.

5) Соответствие сопровождаемости, оценивает соответствие сопровождаемости.

Внутренние метрики мобильностииспользуются для предсказания воздействия программного продукта на поведение исполнителя или системы при проведении работ по переносу. Примеры метрик:

1) Адаптируемость структур данных, оценивает адаптируемость.

2) Объем работ по установке, оценивает настраиваемость.

3) Доступная совместимость, оценивает совместимость.

4) Преемственность данных, оценивает взаимозаменяемость.

5) Соответствие мобильности, оценивает соответствие мобильности.

# 

# 

# 4. Внешние метрики

Внешние метрики функциональностидолжны измерять свойства (атрибуты) функционального поведения системы, содержащей программное средство. Примеры:

1) Полнота функциональной реализации, оценивает пригодность ПС.

2) Точность, оценивает правильность ПС.

3) Способность к обмену данными (основанная на успешных попытках пользователя), оценивает способность к взаимодействию.

4) Предотвращение разрушения данных, оценивает защищенность.

5) Соответствие функциональности, оценивает соответствие функциональности.

Внешние метрики надежностидолжны измерять свойства, связанные с поведением системы, содержащей программное средство, во время тестирования, чтобы показать степень надежности программного средства в системе в процессе эксплуатации.

1) Плотность ошибок, оценивает завершенность.

2) Предотвращение некорректных действий, оценивает устойчивость к ошибке.

3) Способность к восстановлению, оценивает восстанавливаемость.

4) Соответствие надежности, оценивает соответствие надежности.

Внешние метрики практичностипоказывают, в какой мере программное средство может быть понято, изучено, управляемо, привлекательно и соответствует договоренностям и руководствам по практичности.

1) Полнота описания, оценивает понятность.

2) Эффективность документации пользователя, оценивает обучаемость.

3) Исправление ошибок при использовании, оценивает простоту использования.

4) Изменяемость вида интерфейса, оценивает привлекательность.

5) Соответствие практичности, оценивает соответствие практичности.

Эффективность - совокупность свойств программного средства, характеризующая те аспекты его уровня пригодности, которые связаны с характером и временем использования ресурсов, необходимых при заданных условиях функционирования.

1) Использование устройств ввода/вывода.

2) Соответствие эффективности.

3) Время отклика.

Сопровождаемость - совокупность свойств программного средства, характеризующая усилия, которые необходимы для его модификации.

1) Поддержка диагностическими функциями.

2) Возможность управления изменением ПС.

3) Название метрики, формула для её оценки.

4) Возникновение отказа после изменения.

5) Доступность встроенных функций тестирования.

6) Соответствие сопровождаемости.

Мобильность - совокупность свойств программного средства, характеризующая приспособленность для переноса из одной среды функционирования в другие.

1) Адаптируемость структурных данных.

2) Простота установки.

3) Доступная совместимость.

4) Переемственность данных.

5) Соответствие мобильности.

6) Название метрики, формула для её оценки.

**5. Стандарт ISO 25010**

В 2011 году принят стандарт ISO 25010, заменяющий ISO 9126-1 и несколько изменяющий набор характеристик и атрибутов внутреннего качества ПО. В его рамках имеются следующие характеристики и подхарактеристики.

Функциональное соответствие - степень обеспечения программным продуктом или системой функций, отвечающих заданным или подразумеваемым потребностям при использовании в заданных условиях.

Подхарактеристики: функциональная полнота, функциональная правильность, функциональная пригодность.

Эффективность функционирования - зависимость функционирования от кол-ва ресурсов, используемых в заданных условиях. Ресурсы могут включать другие программные продукты, программную и аппаратную конфигурацию системы, материалы.

Подхарактеристики: поведение во времени, использование ресурсов, емкость.

Совместимость - степень возможностей программного продукта, системы или компонента обмениваться информацией с другими продуктами, системами или компонентами и/или выполнять свои заданные функции при совместном использовании одной и той же аппаратной или программной среды.

Подхарактеристики: сосуществование, способность к взаимодействию.

Практичность - степень применимости программного продукта или системы заданными пользователями для достижения заданных целей с результативностью, эффективностью и удовлетворенностью в заданном контексте использования.

Подхарактеристики: распознаваемость пригодности, обучаемость, простота использования, защита от ошибок пользователей, эстетичность пользовательского интерфейса, доступность.

Надежность - степень выполнения системой, программным продуктом или компонентом заданных функций в заданных условиях в течении заданного периода времени.

Подхарактеристики: завершенность, готовность, устойчивость к ошибке, восстанавливаемость.

Защищенность - степень защиты программным продуктом или системой информации и данных так, чтобы люди, другие продукты или системы имели степень доступа к данным, соответствующую типам и уровням их авторизации.

Подхарактеристики: конфиденциальность, целостность, неопровержимость, идентифицируемость, аутентичность.

Сопровождаемость - степень результативности и продуктивности модификации программного продукта или системы запланированным персоналом сопровождения. Модификации включают: исправления, улучшения, адаптация к новым требованиям.

Подхарактеристики: модульность, повторная используемость, анализируемость, модифицируемость, тестируемость.

Мобильность - степень результативности и эффективности переноса системы, программного продукта или компонента из одной аппаратной, программной или иной эксплуатационной или используемой среды в другую.

Подхарактеристики: адаптируемость, простота установки, взаимозаменяемость.

**6. Стандарт ISO/IEC 14598**

Стандарт ISO/IEC 14598–1–6 определяет процессы оценки качества программного продукта, содержит руководство и требования к оценке. Стандарт может применяться при разработке, приобретении и независимой оценке программного средства. Данный стандарт состоит из шести частей:

* 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. В данной части содержатся общие требования к спецификации и оценке качества, разъясняются концепции оценки. Устанавливаются требования к методам измерений и оценки программных продуктов. Определяется общий процесс оценки качества программного продукта.

В 2001–2004 гг. стандарт **ISO/IEC 9126:1991** был заменен на 2 взаимосвязанные серии стандартов: **ISO/IEC 9126–1–4:2001–2004** и **ISO/IEC 14598–1–6:1998–2001**. Стандарт **ISO/IEC 14598** регламентирует метод оценки качества программных средств, который основан на иерархической модели качества, определённой в **ISO/IEC 9126-1:2001.**

Процесс оценки состоит из четырех стадий: установка требований к оценке, определение оценки, проектирование оценки и выполнение оценки. Данный процесс может применяться после любой подходящей работы жизненного цикла для промежуточных или конечного продуктов разработки (сопровождения).

Стандарт **ISO/IEC 12207**, используя устоявшуюся терминологию, устанавливает общую структуру процессов жизненного цикла программных средств, на которую можно ориентироваться в программной индустрии. Стандарт определяет [процессы](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D1%85%D0%BD%D0%BE%D0%BB%D0%BE%D0%B3%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81), виды деятельности и задачи, которые используются при приобретении программного продукта или услуги, а также при поставке, разработке, применении по назначению, сопровождении и прекращении применения программных продуктов.

Стандарт группирует различные виды деятельности, которые могут выполняться в течение жизненного цикла программных систем, в семь групп процессов. Каждый из процессов жизненного цикла в пределах этих групп описывается в терминах цели и желаемых выходов, списков действий и задач, которые необходимо выполнять для достижения этих результатов.

Вторая часть стандарта ISO/IEC 14598–2:2000 содержит требования и руководство по поддержке оценки. В данной части приводятся концепции планирования и управления процессом оценки качества программного продукта, рассматривается содержание плана количественной оценки качества. 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. В ней приводятся концепции оценки и требования к процессу оценки. Данная часть ориентирована на выполнение оценки ПП, используя собственный технический персонал. Внимание в данной части сконцентрировано на оценках, которые могут предсказать качество конечного ПП на основе измерений промежуточных ПП жизненного цикла разработки.

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/IEC12207–1995с работами, выполняемыми при оценке ПП. Рассмотрены особенности выполнения оценки для имеющегося в наличии готового ПП, для приемки заказного ПП и для выбора из альтернативного числа готовых ПП.

Пятая часть стандартаISO/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.

# 7. Модуль оценки

Модуль оценкипредставляет собой полностью укомплектованную информацию, необходимую для проведения процесса оценки некоторой характеристики или подхарактеристики качества. Модуль содержит спецификацию соответствующей модели качества (характеристика, подхарактеристики, внутренние или внешние метрики качества), методики и процедуры оценки, входные данные, связанные с оценкой, информацию о запланированном применении модели и о ее фактическом применении, структуру типового отчета о результатах выполненной оценки. В *шестой части стандарта****ISO/IEC 14598–6:2001*** можно найти примеры трех конкретных модулей оценивания – для метрики «плотность дефектов в ПО», характеристики качества «функциональность», а также обобщенной характеристики «качество при использовании».

Для каждого процесса оценки должны выбираться соответствующие модули оценки. Данная часть стандарта может быть использована организациями, разрабатывающими новые модули оценки и производящими оценку ПС.

**8. Стандарт ISO 12207**

Стандарт ISO 12207 (ГОСТ Р ИСО/МЭК 12207) «Информационная технология. Процессы жизненного цикла программных средств» наиболее полно на уровне международных стандартов отражает жизненный цикл, технологию разработки и обеспечения качества сложных программных средств. Жизненный цикл ПС представлен набором этапов, частных работ и операций в последовательности их выполнения и взаимосвязи, регламентирующих ведение разработки на всех стадиях от подготовки технического задания до завершения испытаний ряда версий и окончания эксплуатации ПС. В жизненный цикл включаются описания исходной информации, способов выполнения операций и работ, устанавливаются требования к результатам и правилам их контроля, а также к содержанию технологических и эксплуатационных документов.

Жизненный цикл (ЖЦ) программного средства – это совокупность процессов, работ и задач, включающая в себя разработку, эксплуатацию и сопровождение ПС и охватывающая жизнь ПС от подготовки технического задания на его разработку до прекращения его использования.

ЖЦ включает:

1. описание исходной информации;
2. описание способов и методов выполнения операций и работ;
3. установку требований к результатам и правилам их контроля;
4. определение содержания технологических и эксплуатационных документов;
5. определение организационной структуры коллектива, распределение и планирование работ;
6. определение видов контроля за ходом разработки.

Повышение эффективности разработки и ЖЦ ПС в целом достигается за счет следующих факторов:

1. регламентации порядка проведения работ;
2. автоматизации этапов и операций;
3. рационального разделения труда между специалистами разной квалификации и проблемной ориентации.

В стандартах ЖЦ ПС рекомендуются наиболее современные и эффективные методы и процессы, в том числе технологические процессы и приемы разработки, методическая база для их автоматизации. Это способствует повышению качества ПС и снижению затрат на их создание.

В соответствии со стандартом ЖЦ ПС и систем имеет трехуровневую иерархическую структуру.

Определяется организационная структура коллективов, распределение и планирование работ, а также контроль за реализацией жизненного цикла ПС.

Стандарт может использоваться как непосредственный директивный, руководящий или рекомендательный документ, а также как организационная база при создании средств автоматизации соответствующих технологических этапов или процессов. Для реализации положений стандарта должны быть выбраны инструментальные средства, совместно образующие взаимосвязанный комплекс технологической поддержки и автоматизации ЖЦ и не противоречащие предварительно скомпонованному набору нормативных документов. Имеющиеся в стандарте пробелы следует заполнять спецификациями или нормативными документами, регламентирующими применение выбранных или созданных инструментальных средств автоматизации разработки и документирования ПС.

# Заключение

Необходимость стандартизации разработки программного обеспечения наиболее удачно описана во введении в стандарт ISO/IEC 12207, где говорится о важности программного обеспечения в различных системах; о сложностях в разработке ПО из-за разнообразия процедур, методов, инструментов, использующихся в объединении и управлении программными продуктами и сервисными программами. Стратегия разработки программного обеспечения требует перехода от этого множества к общему порядку, который позволит специалистам, практикующимся в программном обеспечении, "говорить на одном языке" при разработке и управлении программным обеспечением. Этот международный стандарт обеспечивает такой общий порядок".

Стандарты занимают все более значительное место в направлении развития индустрии информационных технологий. Стандарты обеспечивают возможность разработчикам программного обеспечения использовать данные и программы других разработчиков, осуществлять экспорт/импорт данных. Стандарты регламентируют взаимодействие между различными программами. Все компании-разработчики должны обеспечить приемлемый уровень качества выпускаемого программного обеспечения (ПО). Для этих целей предназначены стандарты качества программного обеспечения или отдельные разделы в стандартах разработки программного обеспечения, посвященные требованиям к качеству программного обеспечения.

# Список литературы

1. Лозинин А.И., Шубинский И.Б «Характеристики качества программного обеспечения и методы их оценки»
2. В.В. Липаев «Стандартизация характеристик и оценивания качества программных средств»
3. Кулямин В. В. «Технологии программирования. Компонентный подход. Лекция 5. Качество ПО и методы его контроля»
4. Глухова Л.А. «Стандартизация и сертификация программного обеспечения. Учебное пособие» – Минск, 2007 – 158 с.
5. ISO 9126 (ГОСТ Р ИСО / МЭК 9126-93) — «Информационная технология. Оценка программного продукта. Характеристики качества и руководство по их применению».
6. ISO 12207 (ГОСТ Р ИСО/МЭК 12207-2010) — «Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств».
7. ISO/IEC 14598 (Части 1-6). — «Информационные технологии. Оценка программного продукта».