ГОСТ Р ИСО/МЭК 15026-2002

Группа П85

ГОСУДАРСТВЕННЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

Информационная технология

УРОВНИ ЦЕЛОСТНОСТИ СИСТЕМ И ПРОГРАММНЫХ СРЕДСТВ

Information technology.
System and software integrity levels

ОКС 35.080
ОКСТУ 5001

Дата введения 2003-07-01

Предисловие

1 РАЗРАБОТАН И ВНЕСЕН Всероссийским научно-исследовательским институтом стандартизации (ВНИИстандарт) Госстандарта России

2 ПРИНЯТ И ВВЕДЕН В ДЕЙСТВИЕ Постановлением Госстандарта России от 11 июня 2002 г. N 237-ст

3 Настоящий стандарт содержит полный аутентичный текст международного стандарта ИСО/МЭК 15026-98 "Информационная технология. Уровни целостности систем и программных средств"

4 ВВЕДЕН ВПЕРВЫЕ

Настоящий стандарт определяет концепцию уровней целостности программных средств и требования к целостности программных средств. Стандарт содержит основные положения, связанные с уровнями целостности, определяет процессы для установления уровней целостности и требований к целостности программных средств, а также устанавливает требования к каждому соответствующему процессу. Стандарт не предопределяет конкретный набор уровней целостности или требования к целостности конкретного программного средства. Данные наборы должны быть установлены либо для программного средства на уровне проектирования, либо для конкретного прикладного сектора и (или) страны. Стандарт применим только для программных средств. Уровень целостности системы и уровни целостности непрограммных компонентов в настоящем стандарте использованы только для определения уровней целостности компонентов программных средств.

Настоящий стандарт предназначен для применения разработчиками, пользователями, поставщиками и экспертами программных продуктов или систем, содержащих программные средства, а также для административной и технической поддержки данных продуктов и систем.

Уровень целостности программного средства не определяет диапазон значений конкретного свойства программного средства, необходимый для удержания системного риска в допустимых границах. Для программного средства, выполняющего функцию амортизации, подобным свойством является надежность, с которой программное средство должно выполнять функцию амортизации. Для программного средства, отказ которого может привести к угрозе для системы, таким свойством является ограничение частоты или вероятности подобного отказа.

Требованиями к целостности программного средства являются требования, удовлетворяемые в процессе программной инженерии, используемом для разработки программного средства, которым должны соответствовать программные продукты, и (или) требования, четко определяющие качество функционирования программного средства во времени, обеспечивающие степень его соответствия заданному уровню целостности.

Настоящий стандарт не устанавливает способ интеграции определения уровня целостности в общие Процессы жизненного цикла программной инженерии.

2 Нормативные ссылки

В настоящем стандарте использованы ссылки на следующие стандарты:

ГОСТ Р ИСО/МЭК 12207-99 Информационная технология. Процессы жизненного цикла программных средств

ИСО/МЭК 2382-1-93* Информационная технология. Словарь. Часть 1: Основополагающие термины

_____

* Оригиналы международных стандартов:
- ИСО (ИСО/МЭК) - во ВНИИКИ Госстандарта России;
- МЭК - во ВНИИстандарт Госстандарта России.

ИСО/МЭК 2382-20-95* Информационная технология. Словарь. Часть 20: Разработка систем

_____

* Оригиналы международных стандартов:
- ИСО (ИСО/МЭК) - во ВНИИКИ Госстандарта России;
- МЭК - во ВНИИстандарт Госстандарта России.

ИСО 8402-94* Управление качеством и обеспечение качества. Словарь

_____

* Оригиналы международных стандартов:
- ИСО (ИСО/МЭК) - во ВНИИКИ Госстандарта России;
- МЭК - во ВНИИстандарт Госстандарта России.

МЭК 50-191-93* Международный электротехнический словарь. Глава 191: Надежность и качество услуги

_____

* Оригиналы международных стандартов:
- ИСО (ИСО/МЭК) - во ВНИИКИ Госстандарта России;
- МЭК - во ВНИИстандарт Госстандарта России.

МЭК 300-3-9-95* Управление надежностью. Часть 3: Прикладное руководство. Раздел 9: Анализ риска технологических систем

_____

* Оригиналы международных стандартов:
- ИСО (ИСО/МЭК) - во ВНИИКИ Госстандарта России;
- МЭК - во ВНИИстандарт Госстандарта России.

3 Определения

В настоящем стандарте применяют определения, приведенные в ИСО/МЭК 2382-1, ИСО/МЭК 2382-20, ИСО 8402 и МЭК 50-191, а также следующие термины с соответствующими определениями.

3.1 **компонент** (component): Элемент внутри системы, имеющий дискретную структуру (например, компоновочный или программный модуль), рассматриваемый на конкретном уровне анализа.

3.2 **степень достоверности** (degree of confidence): Степень уверенности в соответствии программного средства установленным требованиям.

3.3 **ответственный проектант** (design authority): Лицо или организация, которая отвечает за создание проекта системы.

3.4 **отказ** (failure): Прерывание способности объекта выполнять требуемую функцию или невозможность выполнения им заданной функции в заранее установленных границах.

3.5 **локализация отказа** (fault isolation): Способность подсистемы предупреждать отказ в рамках данной подсистемы без вызова последующих отказов в других подсистемах.

3.6 **функция** (function): Аспект определенного поведения системы.

3.7 **инициирующее событие** (initiating event): Событие, которое может привести к угрозе.

3.8 **ответственный за обеспечение целостности** (integrity assurance authority): Независимое лицо или организация, отвечающая за аттестацию на соответствие требованиям целостности.

3.9 **уровень целостности** (integrity level): Указание диапазона значений свойства объекта, необходимых для удержания системного риска в допустимых границах. Для объекта, выполняющего функции амортизации, подобным свойством является надежность, с которой объект должен выполнять функцию амортизации. Для объектов, отказ которых может привести к угрозе для системы, таким свойством является ограничение частоты подобного отказа.

3.10 **объект** (item): Элемент, такой как часть, компонент, подсистема, оборудование или система, который может быть рассмотрен отдельно. Элемент может содержать технические и программные средства или и то и другое.

3.11 **функция амортизации** (mitigating function): Функция, которая, в случае успешной реализации, должна предотвратить возникновение инициирующего события конкретной угрозы.

3.12 **риск** (risk): Функция вероятности возникновения заданной угрозы и потенциально неблагоприятных последствий возникновения этой угрозы.

3.13 **размер риска** (risk dimension): Перспектива, с точки зрения которой будет проведена оценка риска для системы (например, безопасность, экономика, защита).

3.14 **безопасность** (safety): Вероятность того, что система при определенных условиях не придет к состоянию, в котором жизнь человека, его здоровье, собственность или окружающая среда могут быть подвергнуты опасности.

3.15 **защита** (security): Предохранение объектов системы от случайного или преднамеренного доступа, использования, изменения, уничтожения или раскрытия.

3.16 **уровень целостности программного средства** (software integrity level): Уровень целостности программного объекта (элемента).

3.17 **подсистема** (subsystem): Любая система, входящая в другую (большую) систему.

3.18 **система** (system): Сложная структура, состоящая из одного или нескольких процессов, технических и программных средств, устройств и персонала, удовлетворяющая установленным потребностям или целям.

3.19 **систематический отказ** (systematic failure): Отказ, в ряде случаев связанный с конкретным способом его устранения путем изменения проекта или производственного процесса, процедур эксплуатации, документирования или других сопутствующих факторов.

3.20 **уровень целостности системы** (system integrity level): Уровень целостности, заданный для системы.

3.21 **угроза** (threat): Состояние системы или ее окружения, которое может привести к неблагоприятному эффекту при одном или нескольких размерах риска.

4 Символы и сокращения

Отсутствуют символы, используемые в настоящем стандарте.

5 Основы уровней целостности программных средств

5.1 Использование настоящего стандарта

Концепция независимого субъекта, отвечающего за обеспечение целостности, является основополагающей при применении настоящего стандарта. Ответственным за обеспечение целостности является лицо или организация, отвечающая за проведение сертификации на соответствие требованиям целостности. Решения, принятые по согласованию между ответственным проектантом и ответственным за обеспечение целостности, должны быть документально оформлены. Решения, подлежащие согласованию, должны охватывать определение: соответствующих размеров риска, используемых конкретных уровней целостности, конкретного критерия оценки каждого уровня, степени эффективности использования определенных архитектурных особенностей проекта и требований к программному средству, возникающих вследствие установления конкретного уровня целостности.

Процессы, описанные в настоящем стандарте, представлены независимо от общих процессов системной инженерии, но настоящий стандарт не препятствует включению этих процессов в процессы системной инженерии. Кроме того, для соответствия настоящему стандарту при реализации этих процессов необходимо, чтобы все требования к ним были удовлетворены.

Подраздел 5.2 содержит обзор этих процессов с точки зрения установления уровней целостности и требований к целостности программного средства. Разделы 6, 7 и 8 описывают данные процессы более детально и определяют требования, налагаемые на эти процессы.

5.2 Обзор

На рисунке 1 представлен обзор процессов, необходимых для установления уровней целостности системы и программного средства и требований к целостности программного средства. В таблице 1 перечислены исходные данные и выходные результаты для каждого из трех основных процессов: установления уровня целостности системы, установления уровня целостности программного средства и установления требований к целостности программного средства.

Рисунок 1 - Процесс установления и применения уровня целостности программного средства

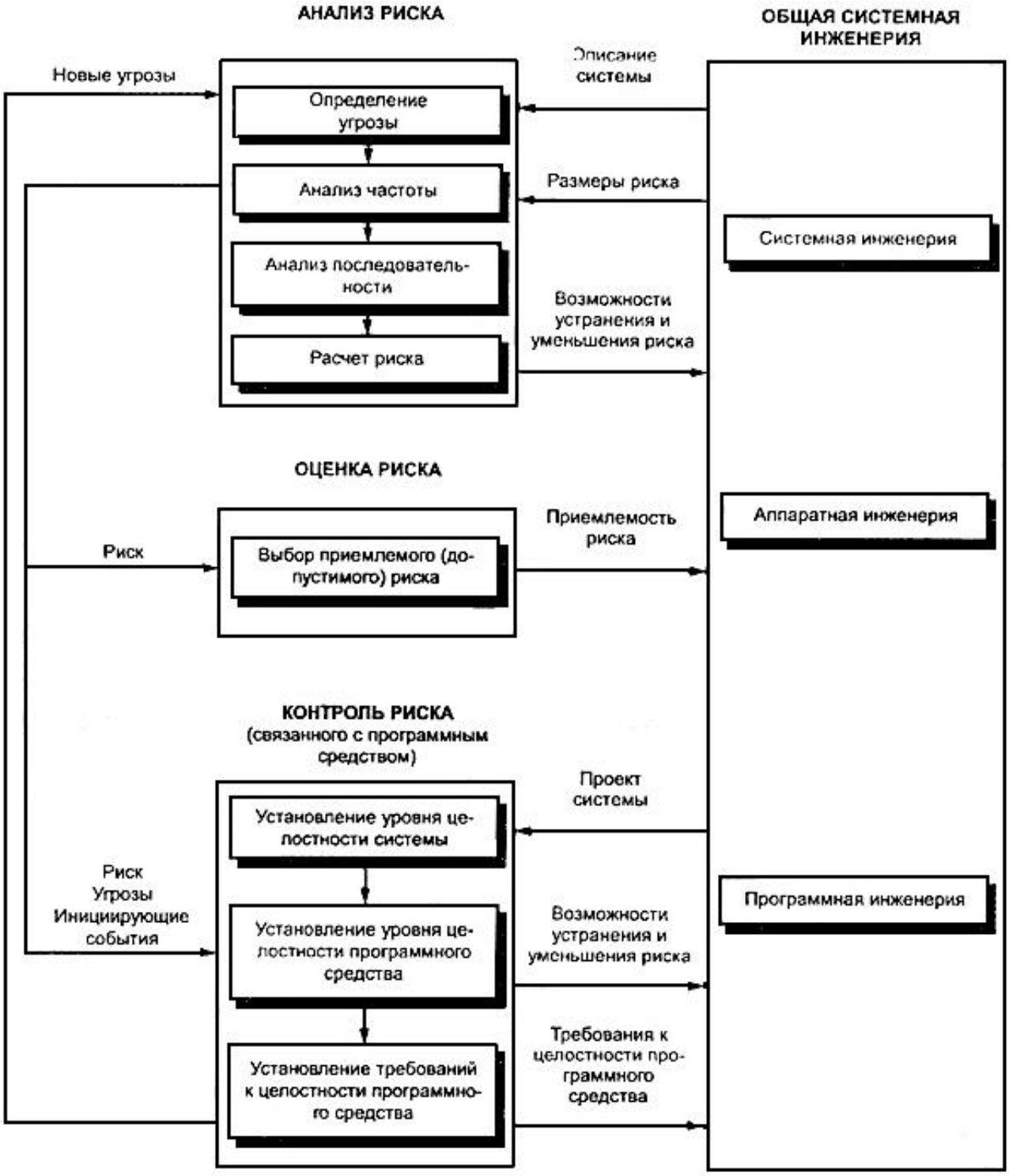

Рисунок 1 - Процесс установления и применения уровня целостности программного средства

Таблица 1 - Исходные данные и выходные результаты

Процесс	Исходные данные	Выходные результаты
Установление уровня целостности системы	Соответствующие размеры риска Описание системы Описание среды Архитектура системы (при ее наличии)	Риск Угрозы Допустимая частота или вероятность появления угрозы Инициирующие события Частота или вероятность инициирующего события Уровень целостности системы
Установление уровня целостности программного средства	Уровень целостности системы Архитектура подсистемы или программного средства Перечень угроз и для каждой угрозы: - допустимая частота или вероятность появления угрозы; - инициирующие события, которые могут привести к угрозе; - ожидаемая частота или вероятность появления каждого инициирующего события	Уровень целостности подсистемы или программного средства Архитектурные особенности, связанные с понижением уровня целостности
Установление требований к целостности программного средства	Уровень целостности подсистемы или программного средства	Требования к целостности программного средства

В настоящем стандарте использован метод установления уровня целостности на основе понятия риска. Поэтому первый шаг при установлении соответствующего уровня целостности системы охватывает выполнение анализа риска. МЭК 300-3-9 содержит руководство по проведению анализа риска. Для того чтобы выполнить анализ риска, должна быть получена достаточная информация о системе, ее среде и допустимых для системы размерах риска. Результаты анализа риска должны охватывать все соответствующие размеры риска в части безопасности, экономики и защиты и быть согласованными ответственным проектантом и ответственным за обеспечение целостности.

Должна быть оценена допустимость любого риска, выявленного в результате анализа. Сразу же после проведения анализа и оценки проекта системы на наличие допустимого риска системе должен быть присвоен уровень целостности. Уровень целостности системы должен отражать наихудший случай риска, связанный с системой.

Уровень целостности программного средства в системе первоначально назначают таким же, как уровень целостности системы. Проект системы может быть проанализирован с целью определить наличие в нем архитектурных особенностей, позволяющих присвоить программному средству пониженный уровень целостности по сравнению с уровнем целостности системы.

Система является интегрированным комплектом, состоящим из одного или нескольких компонентов. Компонентом может быть только программное средство, только техническое средство или подсистема, состоящая из компонентов более низкого уровня. Изначально любому программному компоненту системы должен быть присвоен уровень целостности системы. Установление уровня целостности системы охватывает анализ системной архитектуры для определения возможностей понизить уровни целостности подсистем по сравнению с уровнем целостности системы. Данный процесс может быть применен рекурсивно, до окончательного установления уровня целостности подсистемы, состоящей только из программного средства, или до тех пор, пока уровень целостности подсистемы, содержащей программное средство, не станет приемлемым для ответственного проектанта и ответственного за обеспечение целостности при присвоении его любому программному компоненту в подсистеме.

Возможно, что при проведении анализа архитектуры будут определены новые угрозы и риск, не выявленные при предшествующем анализе. При этом должен быть повторно проведен анализ риска с учетом новой информации.

Уровень целостности программного средства является показателем, отражающим степень надежности функции амортизации или требуемое ограничение частоты отказа, могущее появиться в результате угрозы. Так как отказы программного средства являются строго систематическими, то уровень целостности программного средства является показателем степени достоверности, необходимой для надежного обеспечения функции амортизации или отсутствия отказа, приводящего к угрозе.

Установление и применение уровней целостности программного средства являются частью общего процесса управления риском. Управление риском следует проводить на всем протяжении срока службы системы или программных продуктов и можно проводить итерационно, по мере принятия различных уточнений проекта или по мере развития проекта. На рисунке 1 показаны взаимосвязи между общими процессами системной инженерии и процессами управления риском при его анализе, оценке и контроле.

Итерации возможны, если проект системы изменяют с целью

предусмотреть устранение или уменьшение риска, обнаруживают новые угрозы по сравнению с присвоенными уровнями целостности системы и программного средства или проект системы не обеспечивает допустимого риска.

Уровень целостности программного средства используют при установлении контроля за риском для программных компонентов путем определения требований к программному средству и процессу программной инженерии, обеспечивающих заданную степень достоверности программного средства, связанную с ролью данных требований по отношению к системному риску. Эти требования называют "требованиями к целостности программного средства".

6 Установление уровня целостности системы

Уровень целостности системы должен соответствовать допустимому уровню риска, связанному с ней. Реализация системы может быть связана с риском, потому что ее отказ потенциально приводит к угрозе или ее функциональные возможности предусматривают амортизацию последовательностей инициирующих событий в системной среде, приводящих к угрозе. Установление уровня целостности системы включает в себя следующие этапы:

a) установление уровня риска при анализе риска системы. Анализ риска является процессом использования доступной информации для определения угроз и оценки риска, связанного с этими угрозами;

b) выполнение оценки риска, обеспечивающее подготовку заключений о допустимости риска. Выходными результатами анализа и оценки риска могут быть изменения проекта системы, уменьшающие или устраняющие риск и могущие потребовать повторного проведения процессов анализа и оценки;

c) присвоение уровня целостности системы, выполняемое сразу же после определения допустимого риска для проекта системы. Точное число уровней целостности, из которых выбирают необходимый уровень, и критерии различий между уровнями должны быть согласованы ответственным проектантом и ответственным за обеспечение целостности.

6.1 Анализ риска

Анализ риска должен быть выполнен для ответа на три основных вопроса:

- что может привести к неисправности? (путем определения угрозы);
- какова вероятность того, что это случится? (путем анализа частоты инициирующих событий);
- каковы последовательности этого? (путем анализа последовательности инициирующих событий).

В результате этого анализа должен быть подсчитан каждый риск. Выходными результатами анализа конкретного риска являются:

a) описание риска в соответствующих терминах;

b) угрозы, связанные с данным риском;

c) конкретные инициирующие события, связанные с каждой угрозой;

d) предполагаемая частота возникновения инициирующего события.

Описание риска, полученное в результате анализа, используется процессом оценки риска для установления его допустимости. Все выходные результаты анализа риска используются процессами установления уровней целостности системы и программного средства для определения требуемого уровня целостности компонентов программных средств в системе.

Соответствующим руководством при анализе риска является МЭК 300-3-9. Специфичные термины в части безопасности, использованные в МЭК 300-3-9, а именно "опасность (hazard)" и "ущерб (harm)" должны быть истолкованы соответственно как "угроза (threat)" и "вредное воздействие (adverse effect)".

Анализ риска может охватывать множество размеров риска, например по безопасности, экономике и защите. Конкретные размеры определенного риска должны быть установлены ответственным проектантом и ответственным за сохранение целостности.

6.1.1 Определение угрозы

Угрозы, связанные с системой, должны быть определены вместе с инициирующими событиями, могущими привести к этим угрозам. Угрозы связаны с системой, если отказ системы может привести к конкретной угрозе, эксплуатация системы в установленном режиме может привести к конкретной угрозе или выполнение системой функции амортизации связано с инициирующим событием в среде эксплуатации, которое может привести к угрозе. Угрозы не могут быть указаны заранее, для этого должны быть использованы и документально оформлены методы их определения, охватывающие конкретную ситуацию (см. МЭК 300-3-9). При наличии архитектуры системы эта ситуация должна быть учтена при определении угрозы с целью установить виды отказов, связанных с конкретными технологиями, использованными в системе.

6.1.2 Анализ частоты

Анализ частоты используют для оценки вероятности каждого инициирующего события, выявленного при определении угрозы. Когда инициирующим событием является отказ системы, анализом не оценивается вероятность отказа, но устанавливается ограничение частоты отказа, которое должно находиться в границах допустимого риска и достижимо практически.

Каждая частота должна быть оценена с использованием статистических опытных данных, обобщенных на основе соответствующих методов, например анализа дерева отказов или дерева событий (см. МЭК 300-3-9), или оценена на основе экспертных заключений.

При анализе частоты, проводимом в процессе установления уровня целостности системы, систему следует трактовать как "черный ящик", а архитектуру системы при этом не учитывают. Архитектура системы должна быть учтена при установлении уровня целостности программного средства. Частота конкретного инициирующего события, определенная в данном анализе, может быть выражена в количественных или в качественных терминах, указывающих диапазоны частот, таких как обычная, возможная, случайная, маловероятная, невероятная или неправдоподобная.

Анализ частоты должен учитывать, что некоторые инициирующие события приведут к угрозе только в сочетании с другими событиями или как часть последовательности этих событий.

6.1.3 Анализ последовательности

Анализ последовательности используют для оценки опасности угрозы, вызванной инициирующим(и) событием(ями). При этом должны быть определены любые меры, могущие амортизировать последовательность инициирующих событий. Каждая угроза должна быть связана с категорией последовательности из набора категорий, описывающего различные степени опасности данной последовательности, например, катастрофическая, главная, опасная или незначительная.

6.1.4 Расчет риска

Риск, связанный с каждой угрозой, рассчитывают с использованием матрицы риска, связывающей частоту появления инициирующих событий с опасностью их последовательности. Матрица риска устанавливает классы риска, такие как высокий, средний, низкий или обычный, для каждой комбинации частоты и опасности. В таблице 2 приведен пример матрицы риска, взятый из МЭК 300-3-9.

Таблица 2 - Пример матрицы риска

Частота появления	Характерная частота (в год)	Опасность последовательности			
		Катастрофическая	Главная	Опасная	Незначительная
Обычная	> 1	Высокий	Высокий	Высокий	Средний
Вероятная	$1\text{-}10^{-1}$	Высокий	Высокий	Средний	Низкий
Случайная	$10^{-1}\text{-}10^{-2}$	Высокий	Высокий	Низкий	Низкий
Маловероятная	$10^{-2}\text{-}10^{-4}$	Высокий	Высокий	Низкий	Низкий
Невероятная	$10^{-4}\text{-}10^{-6}$	Высокий	Средний	Низкий	Обычный
Неправдоподобная	$< 10^{-6}$	Средний	Средний	Обычный	Обычный

Ответственный проектант и ответственный за обеспечение целостности должны согласовать любую матрицу, используемую при расчете риска. Для согласования матрицы риска необходимы соглашения по соответствующим диапазонам частот инициирующих событий, конкретным категориям их последовательностей и определений и классам риска, связанным с каждой парой диапазона частоты и категории последовательности.

6.2 Оценка риска

Ответственным проектантом и ответственным за обеспечение целостности должно быть выдано заключение о допустимости каждого риска, выявленного в процессе его анализа. В некоторых областях подобное заключение должно быть сделано по отношению к заданным ограничениям, установленным соответствующим регулирующим органом в конкретной области.

6.3 Присвоение уровня целостности системы

Уровень целостности системы изначально назначают в соответствии с наивысшим классом риска, присвоенным любым угрозам, связанным с системой. Число уровней целостности должно соответствовать числу установленных классов риска. В таблице 3 приведен пример распределения классов риска по уровням целостности системы.
Таблица 3 - Соотношение класса риска с уровнем целостности системы

Класс риска	Уровень целостности системы
Высокий	A
Средний	B
Низкий	C
Обычный	D

Настоящий стандарт не устанавливает число уровней риска или используемые для них обозначения.

7 Установление уровня целостности

программного средства

Уровень целостности программного средства представляет собой распределение уровня целостности системы по подсистемам, содержащим программные средства в качестве компонента или состоящим только из программного средства. Первоначально уровень целостности программного средства должен быть установлен на уровне целостности системы.

Данное условие остается в силе до тех пор, пока уровень целостности программного средства не будет понижен с учетом следующих факторов:

a) архитектурных характеристик системы, амортизирующих последовательности отказов подсистем, могущих в противном случае привести к возникновению системной угрозы;

b) архитектурных характеристик системы, обеспечивающих избыточность функций, амортизирующих последствия единичных или множественных отказов в подсистемах, для которых предназначены данные функции амортизации;

c) роли подсистемы, связанной с определением инициирующих событий или функций амортизации.

7.1 Предпосылки при установлении уровня целостности программного средства

При установлении уровня целостности программного средства учитывают следующие факторы:

a) для системы существует присвоенный ей уровень целостности;

b) архитектурные характеристики системы должны быть определены подробно, чтобы позволить установить роли подсистем и их интерфейсы;

c) исходные данные охватывают:

1) уровень целостности системы;

2) перечень угроз и следующую информацию для каждой угрозы об:
- инициирующих событиях, могущих привести к угрозе,
- ожидаемой частоте или вероятности совпадения каждого инициирующего события;

3) описание архитектуры системы с точностью, достаточной для определения роли каждой подсистемы и любых амортизирующих характеристик архитектуры;

d) выходным результатом является уровень целостности программного средства.

7.2 Понижение уровня целостности программного средства

При определении возможности снизить уровень целостности системы должны быть выполнены следующие этапы:

a) определен набор подсистем, образующих систему;

b) определена возможность возникновения угрозы при отказе любой подсистемы (единичном или в сочетании с состояниями других подсистем). Если единичный отказ подсистемы приводит к угрозе, то уровень целостности подсистемы назначают таким же, как уровень целостности системы. Если отказ подсистемы может привести к угрозе только в сочетании с состояниями других подсистем, то уровень целостности подсистемы может быть понижен в зависимости от результата оценки, определенной в 7.3. Оценка в соответствии с 7.3 не является обязательной, ее выполняют, если предполагают понижение уровня целостности подсистемы;

c) определена возможность невыполнения функции амортизации при отказе какой-либо подсистемы (единичном или в сочетании с состояниями других подсистем). Если единичный отказ подсистемы приводит к невыполнению функции амортизации, то уровень целостности подсистемы назначают таким же, как уровень целостности системы. Если отказ подсистемы может привести к невыполнению функции амортизации только в сочетании с состояниями других подсистем, то уровень целостности подсистемы может быть понижен в зависимости от результата оценки, определенной в 7.4. Оценка в соответствии с 7.4 не является обязательной, ее выполняют, если предполагают понижение уровня целостности подсистемы;

d) определено наличие подсистем с программными компонентами, отказ которых не может привести к угрозе, а их функциональные возможности не связаны с амортизацией любых системных событий. Любому такому программному средству должен быть присвоен пониженный уровень целостности. Для обеспечения того, чтобы отказ программного средства не смог привести к угрозе, необходима локализация неисправностей. Ответственный проектант и ответственный за обеспечение целостности должны согласовать соответствующие архитектурные характеристики, позволяющие локализовать соответствующие неисправности. Если локализация неисправности обеспечивается механизмом обработки отказа, данному механизму должен быть присвоен уровень целостности программного средства такой же, как для системы.

Ответственным за обеспечение целостности и ответственным проектантом должны быть определены и согласованы преимущества, реализуемые за счет архитектурных характеристик при понижении уровней целостности программного средства по сравнению с присвоенным уровнем целостности системы.

Описанный пошаговый процесс применяют рекурсивно до тех пор, пока не будут установлены уровни целостности всех подсистем, содержащих только программные средства, или уровни целостности всех таких подсистем не станут приемлемыми для ответственного проектанта и ответственного за обеспечение целостности при присвоении их любому программному компоненту данных подсистем.

7.3 Понижение уровня целостности для

программных средств, отказ которых может привести к угрозе

Для систем, отказ компонентов которых (например, программного средства) может привести к угрозе, уровень целостности системы частично основан на ограничении частоты отказа системы, согласованном с границами допустимого риска. Если отказ программного средства может привести к угрозе только в сочетании с другими подсистемами, находящимися в конкретном состоянии, возможно назначение менее строгого ограничения на частоту отказа данного средства, используемого в системе. Анализ частоты может быть использован для установления менее строгого ограничения на частоту отказа программного средства, соответствующего границам допустимого риска, и должен учитывать зависимости между подсистемами. Установленное верхнее ограничение частоты отказа программного средства может быть использовано при повторном расчете риска для определения наличия какого-либо изменения в классе риска и соответствующего снижения уровня целостности программного средства.

Механизмы обработки отказа могут быть использованы для обнаружения отказов программного средства и выполнения действий по их предотвращению при возникновении инициирующих событий. В этих случаях отказ программного средства может стать инициирующим событием при его возникновении и несрабатывании механизма обработки отказа. Примерами механизмов обработки отказов являются:

- проверки целостности данных (программный механизм);
- аппаратные контрольные реле времени (аппаратный механизм);
- ручное восстановление (механизм, реализуемый человеком).

Для предотвращения отказов, приводящих к угрозам, может быть использовано резервирование, при этом следует предусмотреть предотвращение общих отказов резервируемых подсистем. При резервировании программных компонентов для предотвращения их общих отказов должна быть использована соответствующая стратегия, например диверсификация (разные версии) программного средства. При использовании диверсификации программного средства для понижения строгости ограничений на частоту его отказов преимущества, обеспечиваемые диверсификацией, и описания ее сути должны быть согласованы между ответственным проектировщиком и ответственным за обеспечение целостности.

7.4 Понижение уровня целостности для программных средств, отказ которых может привести к снижению возможностей функций амортизации

Если отказ подсистемы может привести к невыполнению функции амортизации только в сочетании с состояниями других подсистем, надежность программного средства может быть снижена по сравнению с надежностью, требуемой системой от функции амортизации. Как и в 7.3, в данном случае могут быть использованы методы анализа частоты отказа программного средства с целью установить степень снижения его надежности по сравнению с надежностью системы.

8 Установление требований к целостности программного средства

8.1 Степень достоверности

Уровень целостности программного средства определяет либо требуемую степень надежности выполнения функции амортизации, либо требуемое ограничение частоты отказа, могущего привести к угрозе. Так как отказы программного средства являются строго систематическими, то уровень его целостности определяют показателем степени достоверности, необходимой для надежного обеспечения функции амортизации или предотвращения возникновения отказа, приводящего к угрозе. Требования, которым должно соответствовать программное средство и процессы его жизненного цикла, должны, в случае их удовлетворения, обеспечить необходимую степень достоверности, их называют "требованиями к целостности программного средства".

Некоторыми способами ("стратегиями") обеспечения достоверности программного средства являются:

a) применение методов, минимизирующих внесение ошибок;

b) применение методов верификации и аттестации, позволяющих с максимальной вероятностью обнаружить ошибки;

c) демонстрация, на конкретных примерах статистики эксплуатации программного средства, надежности обеспечения им функции амортизации или непревышения установленной частоты отказов данного средства.

8.2 Методы обеспечения степени достоверности программного средства

В таблице 4 приведены схематические примеры некоторых работ программной инженерии, атрибутов выходных результатов этих работ и методов, которые могут быть использованы для достижения различных степеней достоверности реализации данных атрибутов. Настоящий стандарт не предназначен для определения жизненного цикла конкретного программного средства. При определении степени достоверности может быть использован ряд процессов жизненного цикла программных средств, описанных в ГОСТ Р ИСО/МЭК 12207.

Таблица 4 - Примеры работ процесса, атрибутов и степени уверенности

Работа	Атрибуты	Методы достижения степени уверенности в программном средстве для каждого уровня целостности
Анализ требований к программному средству	Точность, полнота, правильность, абстрагирование, непротиворечивость, проверяемость	1 Структурированный подход 2 (1) плюс полуформальные нотации 3 (1) плюс формальные нотации 4 (3) плюс формальная проверка
Проектирование программного средства	Анализ трассируемости требований, проверяемость, модульность, абстрагирование	1 Структурированный подход 2 (1) плюс полуформальные нотации 3 (1) плюс формальные нотации 4 (3) плюс формальная проверка
Программирование программного средства	Трассируемость проекта, однозначность программных конструктивов, стандартизация языка программирования, сопровождаемость	1 Структурное программирование 2 (1) плюс строго типизированный язык 3 (2) плюс ограничение на подмножество языка 4 (3) плюс формальная проверка
Примечание - В скобках указаны позиции текущего перечисления, дополняемые последующим текстом.		

Достоверность программного средства также может быть определена при оценке статистики эксплуатации, если данное средство уже создано и его эксплуатируют некоторое время. Степень достоверности на основе статистики эксплуатации может зависеть от таких факторов, как сложность программного средства, положительные результаты статистики его эксплуатации и соответствие режимов эксплуатации целевому использованию данного средства в системе.

8.3 Связь степени достоверности программного средства с уровнем его целостности

Настоящий стандарт не определяет конкретных требований, подлежащих удовлетворению для достижения степени достоверности, соответствующей заданному уровню целостности программного средства.

Ответственный проектант и ответственный за обеспечение целостности должны согласовать требования, подлежащие удовлетворению для достижения необходимой степени достоверности программного средства на каждом уровне его целостности.

Текст документа сверен по:
официальное издание
М.: ИПК Издательство стандартов, 2002