**Ка́чество програ́ммного обеспечения** — способность [программного продукта](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5) при заданных условиях удовлетворять установленным или предполагаемым [потребностям](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D1%82%D1%80%D0%B5%D0%B1%D0%BD%D0%BE%D1%81%D1%82%D1%8C) (ISO/IEC 25000:2014)[[1]](https://ru.wikipedia.org/wiki/%D0%9A%D0%B0%D1%87%D0%B5%D1%81%D1%82%D0%B2%D0%BE_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D1%8F#cite_note-1).

Другие определения:

* весь объем признаков и характеристик программ, который относится к их способности удовлетворять установленным или предполагаемым потребностям (ГОСТ Р ИСО/МЭК 9126-93, ISO 8402:94)[[2]](https://ru.wikipedia.org/wiki/%D0%9A%D0%B0%D1%87%D0%B5%D1%81%D1%82%D0%B2%D0%BE_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D1%8F#cite_note-2)[[3]](https://ru.wikipedia.org/wiki/%D0%9A%D0%B0%D1%87%D0%B5%D1%81%D1%82%D0%B2%D0%BE_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D1%8F#cite_note-3);
* степень, в которой система, компонент или процесс удовлетворяют потребностям или ожиданиям заказчика или пользователя (IEEE Std 610.12-1990)[[4]](https://ru.wikipedia.org/wiki/%D0%9A%D0%B0%D1%87%D0%B5%D1%81%D1%82%D0%B2%D0%BE_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D1%8F#cite_note-4).

**Содержание**

  [скрыть]

* [1Характеристики качества](https://ru.wikipedia.org/wiki/%D0%9A%D0%B0%D1%87%D0%B5%D1%81%D1%82%D0%B2%D0%BE_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D1%8F#.D0.A5.D0.B0.D1.80.D0.B0.D0.BA.D1.82.D0.B5.D1.80.D0.B8.D1.81.D1.82.D0.B8.D0.BA.D0.B8_.D0.BA.D0.B0.D1.87.D0.B5.D1.81.D1.82.D0.B2.D0.B0)
* [2См. также](https://ru.wikipedia.org/wiki/%D0%9A%D0%B0%D1%87%D0%B5%D1%81%D1%82%D0%B2%D0%BE_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D1%8F#.D0.A1.D0.BC._.D1.82.D0.B0.D0.BA.D0.B6.D0.B5)
* [3Примечания](https://ru.wikipedia.org/wiki/%D0%9A%D0%B0%D1%87%D0%B5%D1%81%D1%82%D0%B2%D0%BE_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D1%8F#.D0.9F.D1.80.D0.B8.D0.BC.D0.B5.D1.87.D0.B0.D0.BD.D0.B8.D1.8F)
* [4Литература](https://ru.wikipedia.org/wiki/%D0%9A%D0%B0%D1%87%D0%B5%D1%81%D1%82%D0%B2%D0%BE_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D1%8F#.D0.9B.D0.B8.D1.82.D0.B5.D1.80.D0.B0.D1.82.D1.83.D1.80.D0.B0)
* [5Ссылки](https://ru.wikipedia.org/wiki/%D0%9A%D0%B0%D1%87%D0%B5%D1%81%D1%82%D0%B2%D0%BE_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D1%8F#.D0.A1.D1.81.D1.8B.D0.BB.D0.BA.D0.B8)

Характеристики качества[[править](https://ru.wikipedia.org/w/index.php?title=%D0%9A%D0%B0%D1%87%D0%B5%D1%81%D1%82%D0%B2%D0%BE_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D1%8F&veaction=edit&section=1) | [править вики-текст](https://ru.wikipedia.org/w/index.php?title=%D0%9A%D0%B0%D1%87%D0%B5%D1%81%D1%82%D0%B2%D0%BE_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D1%8F&action=edit&section=1)]

Стандарт ISO/IEC 25010:2011 (ГОСТ Р ИСО/МЭК 25010-2015)[[5]](https://ru.wikipedia.org/wiki/%D0%9A%D0%B0%D1%87%D0%B5%D1%81%D1%82%D0%B2%D0%BE_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D1%8F#cite_note-ISO25010-5) определяет *модель качества продукта*, которая включает восемь характеристик верхнего уровня:

* функциональная пригодность;
* уровень производительности;
* совместимость;
* [удобство пользования](https://ru.wikipedia.org/wiki/%D0%AE%D0%B7%D0%B0%D0%B1%D0%B8%D0%BB%D0%B8%D1%82%D0%B8);
* [надёжность](https://ru.wikipedia.org/wiki/%D0%9D%D0%B0%D0%B4%D1%91%D0%B6%D0%BD%D0%BE%D1%81%D1%82%D1%8C);
* защищённость;
* [сопровождаемость](https://ru.wikipedia.org/wiki/%D0%A1%D0%BE%D0%BF%D1%80%D0%BE%D0%B2%D0%BE%D0%B6%D0%B4%D0%B5%D0%BD%D0%B8%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D1%8F);
* [переносимость (мобильность)](https://ru.wikipedia.org/wiki/%D0%9F%D0%B5%D1%80%D0%B5%D0%BD%D0%BE%D1%81%D0%B8%D0%BC%D0%BE%D1%81%D1%82%D1%8C_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5)).

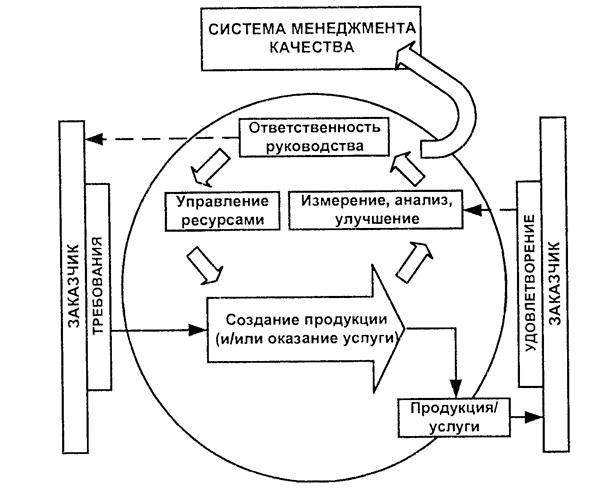
В этом стандарте модель качества продукта ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA)  *software product quality model*) рассматривается отдельно от субъективного *качества в использовании* ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *quality in use model*), которое может сильно отличаться для различных [стейкхолдеров](https://ru.wikipedia.org/wiki/%D0%A1%D1%82%D0%B5%D0%B9%D0%BA%D1%85%D0%BE%D0%BB%D0%B4%D0%B5%D1%80" \o "Стейкхолдер)[[6]](https://ru.wikipedia.org/wiki/%D0%9A%D0%B0%D1%87%D0%B5%D1%81%D1%82%D0%B2%D0%BE_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D1%8F#cite_note-Wijnholds.2C_et_al.E2.80.942016.E2.80.94.E2.80.94-6). Модель включает следующие характеристики верхнего уровня[[5]](https://ru.wikipedia.org/wiki/%D0%9A%D0%B0%D1%87%D0%B5%D1%81%D1%82%D0%B2%D0%BE_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D1%8F#cite_note-ISO25010-5):

* результативность;
* производительность;
* удовлетворенность;
* свобода от риска;
* покрытие контекста.

[Роберт Гласс](https://ru.wikipedia.org/w/index.php?title=%D0%93%D0%BB%D0%B0%D1%81%D1%81,_%D0%A0%D0%BE%D0%B1%D0%B5%D1%80%D1%82&action=edit&redlink=1)[[en]](https://en.wikipedia.org/wiki/Robert_L._Glass) в известной книге «Факты и заблуждения профессионального программирования» утверждает, что большинство профессиональных разработчиков согласны с выделением семи показателей качества как основных[[7]](https://ru.wikipedia.org/wiki/%D0%9A%D0%B0%D1%87%D0%B5%D1%81%D1%82%D0%B2%D0%BE_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D1%8F#cite_note-7):

* [переносимость](https://ru.wikipedia.org/wiki/%D0%9F%D0%B5%D1%80%D0%B5%D0%BD%D0%BE%D1%81%D0%B8%D0%BC%D0%BE%D1%81%D1%82%D1%8C_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5));
* [надёжность](https://ru.wikipedia.org/wiki/%D0%9D%D0%B0%D0%B4%D1%91%D0%B6%D0%BD%D0%BE%D1%81%D1%82%D1%8C);
* эффективность;
* [юзабилити](https://ru.wikipedia.org/wiki/%D0%AE%D0%B7%D0%B0%D0%B1%D0%B8%D0%BB%D0%B8%D1%82%D0%B8);
* [тестируемость](https://ru.wikipedia.org/w/index.php?title=%D0%A2%D0%B5%D1%81%D1%82%D0%B8%D1%80%D1%83%D0%B5%D0%BC%D0%BE%D1%81%D1%82%D1%8C&action=edit&redlink=1);
* понятность;
* модифицируемость

<https://ru.wikipedia.org/wiki/Качество_программного_обеспечения>



**Надёжность программных систем**

http://studopedia.ru/3\_115210\_pokazateli-kachestva-programmnih-produktov.html

Одним из основных показателей программного продукта является его надёжность. Надёжность определяется количеством ошибок, содержащихся в программном продукте.

Можно выделить два класса ошибок:

1. Отказы программы (невозможность получить результат выполнения функции).

2. Неправильные с точки зрения пользователя результаты работы, даже если с точки зрения разработчика эти результаты верны

«В программном обеспечении имеется ошибка, если оно не выполняет того, что пользователю разумно от него ожидать», то есть ошибки не являются внутренним свойством программы, наличие ошибок зависит как от самого программного обеспечения, так и от ожиданий пользователя.

Надёжность программного обеспечения есть вероятность его работы без отказов в течение определённого периода времени, рассчитанная с учётом стоимости для пользователя каждого отказа.

Можно выделить две основные причины ошибок:

1. Недопонимание между участниками процесса разработки (как внутри коллектива разработчиков, так и между заказчиком и разработчиками). Сюда, например, можно отнести:

- чтение между строк — исполнитель может неоднозначные моменты понимать по-своему, а отсутствующие моменты додумывать;

- непонимание — невозможность понять, например, из-за недостатка знаний в некоторой области;

- нечёткое выражение мыслей — созданный документ изначально не соответствует замыслу составителя.

Сложность системы. При увеличении программы больше некоторого критического объема разработчик больше не в состоянии эффективно отслеживать её внутренние связи, что потенциально ведёт к ошибкам. Также сложность может увеличивать непонимание. Основным методом борьбы со сложностью является разбиение системы на компоненты, при этом, с одной стороны, компоненты должны быть достаточно простыми внутри, и, с другой стороны, их взаимодействие также должно быть не сложным. Для уменьшения сложности взаимодействия, компоненты организуют в иерархическую струткуру.

Четыре подхода к надёжности.

1. Предупреждение ошибок — количество ошибок минимизируют на этапе проектирования и реализации.

2. Обнаружение ошибок — программа во время выполнения способна обнаруживать возникающие ошибки.

3. Исправление ошибок — программа во время выполнения способна исправлять обнаруженные ошибки.

4. Устойчивость к ошибкам — несмотря на возникновение ошибок, программа способна продолжать работать, возможно, со сниженными показателями качества.

Для обнаружения ошибок рекомендуется использовать в подпрограммах проверку входных данных даже там, где в них нет прямой необходимости (например, спецификация вызывающего модуля гарантирует отсутствие вызовов с неверными аргументами).

Для обнаружения ошибок и устойчивости к ним может быть использовано резервирование — введение функциональной избыточности. Может осуществляться как информационное (внесение избыточной информации), так и структурное резервирование (добавление программных модулей). Резервирование целесообразно применять для высоконадёжных программных систем с высокой стоимостью отказов.

Реализацией информационного резервирования является использование помехозащитного кодирования.

Реализация структурного резервирования требует создания модулей, дублирующих, работу наиболее критичных по надёжности модулей. При этом резервный модуль должен быть спроектирован независимо от основного (например, другой группой разработчиков или с использованием другого метода), чтобы минимизировать вероятность появления одинаковой ошибки в обоих модулях. При одновременном использовании трех идентичных по функциональности модулей появляется возможность не только обнаруживать ошибки, но и продолжать правильно функционировать.

Многие современные языки программирования включают средства обработки ошибок. Например, в С++ для этого предназначена конструкция try — throw — catch.

Современные операционные системы персональных компьютеров изолируют процессы друг от друга, то есть не позволяют конкретному процессу занять всё процессорное время или получить доступ к данным других процессов. Благодаря этому, при отказе одного из процессов продолжение функционирования остальной системы оказывается достаточно вероятным.

Все методы повышения надёжности, основанные на обнаружении, исправлении и устойчивости к ошибкам усложняют программу, а, следовательно, одновременно и уменьшают её надёжность.

Предупреждение ошибок базируется на правильном выполнении технологии программирования. Оно не требует усложнения программы и, следовательно, является предпочтительным.