



SOFTWARE TESTING AND QUALITY ASSURANCE

**ТЕСТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И
ГАРАНТИЯ КАЧЕСТВА**

KSHIRASAGAR NAIK

Department of Electrical and Computer Engineering University of Waterloo, Waterloo

PRIYADARSHI TRIPATHY

NEC Laboratories America, Inc

Basic Concepts and Preliminaries

Основные понятия. Введение

Software is like entropy. It is difficult to grasp, weighs nothing, and obeys the second law of thermodynamics, i.e., it always increases.

Norman Ralph Augustine.

Компьютерная программа подобна энтропии. Она трудна в управлении, не имеет никакой массы и подчиняется второму закону термодинамики, т.е. размер программы постоянно растёт. - Норман Ральф Августин.

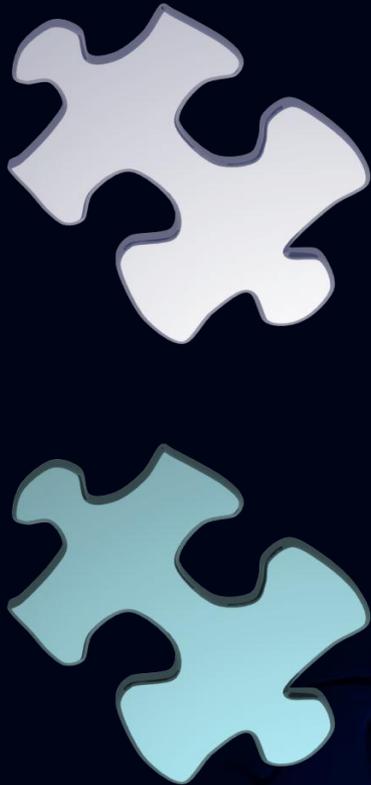


1.2 Software Quality

Качество программного обеспечения

The question "What is software quality?" evokes many different answers. Quality is a complex concept—it means different things to different people, and it is highly context dependent. Garvin [18] has analyzed how software quality is perceived in different ways in different domains, such as philosophy, economics, marketing, and management. Kitchenham and Pfleeger's article [60] on software quality gives a succinct exposition of software quality. They discuss five views of quality in a comprehensive manner as follows:

Вопрос "Что такое качество программного обеспечения?" порождает множество разнообразных ответов. Качество, это комплексное понятие – имеющее для каждого свой смысл и очень сильно зависящее от ситуации. Гарвин (в своей статье "What Does "Product Quality" Really Mean?") проанализировал насколько отличаются понятия качества применительно к разным сферам деятельности человека, таким как философия, экономика, маркетинг и менеджмент. Статья Китченхама и Пфлегера, на эту же тему ("Software Quality: The Elusive Target") даёт краткое описание качества программного обеспечения. Авторы всесторонне рассмотрели следующие пять подходов к его определению:



1. Transcendental
View

1. Transcendental View - Трансцендентный подход

2. User View

2. User View - Подход, ориентированный на пользователя

3. Manufacturing
View

3. Manufacturing View - Подход, ориентированный на производственный процесс

4. Product View

4. Product View - Подход, ориентированный на готовый продукт

5. Value-Based
View

5. Value-Based View - Подход, основанный на стоимости продукта



1. Transcendental View

1. **Transcendental View:** It envisages quality as something that can be recognized but is difficult to define. The transcendental view is not specific to software quality alone but has been applied in other complex areas of everyday life. For example, In 1964, Justice Potter Stewart of the **U.S. Supreme Court**, while ruling on the case *Jacobellis v. Ohio*, 378 U.S. 184 (1964), which involved the state of Ohio banning the French film *Les Amants* ("The Lovers") on the ground of pornography, wrote "I shall not today attempt further to define the kinds of material I understand to be embraced within that shorthand description; and perhaps I could never succeed in intelligibly doing so. But / *know it when I see it*, and the motion picture involved in this case is not that" (*emphasis added*).

1. **Трансцендентный подход:** При таком подходе, качество можно представить как нечто очевидное, но сложное для четкого определения. Трансцендентный подход характерен не только для определения качества программного обеспечения, но также применим в других трудноразрешимых вопросах нашей повседневной жизни. Например, в 1964 году, судья Верховного Суда США Поттер Стюарт, во время вынесения судебного решения по делу Якобеллиса против штата Огайо, 378 U.S. 184 (1964), которое касалось запрещения, наложенного штатом Огайо, французского фильма "Любовники" ("The Lovers") на основании его порнографического содержания, написал: "Я не буду сегодня и в будущем пытаться определить вид, к которому относится данный материал, который, как я понимаю, должен быть полностью охвачен этим условным определением; и, возможно никогда не смогу вразумительно сделать это. Но **я знаю это, когда я вижу это**, и кинокартина, вовлечённая в этот процесс, не относится к такой категории (запрещённой)" (*выделено мной*).



2. User View

2. *User View*: It perceives quality as fitness for purpose. According to this view, while evaluating the quality of a product, one must ask the key question: "Does the product satisfy user needs and expectations?"

2. *Подход, ориентированный на пользователя*: Качество понимают как соответствие продукта своему назначению. Согласно этому подходу, оценка качества заключается в ответе на ключевой вопрос: "Удовлетворяет ли продукт потребностям и ожиданиям пользователя"?

3. Manufacturing View

3. *Manufacturing View*: Here quality is understood as conformance to the specification. The quality level of a product is determined by the extent to which the product meets its specifications.

3. *Подход, ориентированный на производственный процесс*. В этом случае качество понимают как соответствие описанию спецификации. Уровень качества продукта определён степенью соответствия определённым для него в спецификации техническим требованиям.



4. Product View

4. *Product View*: In this case, quality is viewed as tied to the inherent characteristics of the product. A product's inherent characteristics, **that is**, internal qualities, determine its external qualities.

4. *Подход, ориентированный на готовый продукт*: В этом случае, качество видят как взаимосвязь неотъемлемых характеристик продукта. Эти характеристики, т.е. внутреннее его качество, определяют качество внешнее.

5. Value-Based View

5. *Value-Based View*: Quality, in this perspective, depends on the amount a customer is willing to pay for it.

5. *Подход, основанный на стоимости продукта*: Качество, с этой точки зрения, зависит от суммы, которую пользователь готов платить за него.



The concept of software quality and the efforts to understand it in terms of measurable quantities date back to the mid-1970s. McCall, Richards, and Walters [19] were the first to study the concept of software quality in terms of quality factors and quality criteria. A quality factor represents a behavioral characteristic of a system. Some examples of high-level quality factors are correctness, reliability, efficiency, testability, maintainability, and reusability.

A quality criterion is an attribute of a quality factor that is related to software development. For example, modularity is an attribute of the architecture of a software system. A highly modular software allows designers to put cohesive components in one module, thereby improving the maintainability of the system.



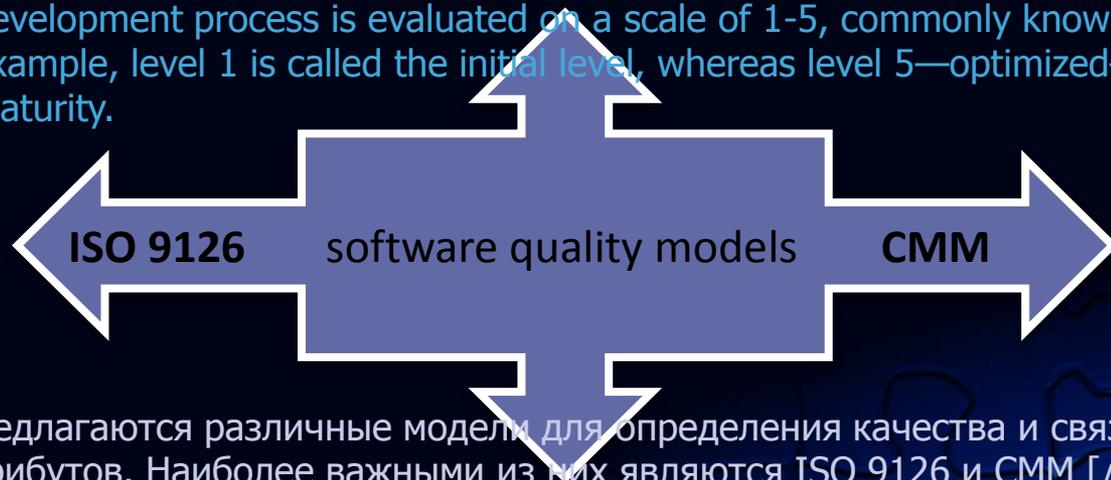
Концепция качества ПО и попытка осмыслить её в терминах, описывающих большое количество поддающихся измерению данных, возвращает нас в середину 1970-х годов. МакКолл, Ричардс и Уолтерс, изучавшие в то время концепцию качества, первыми ввели термины фактор качества и критерий качества. Фактор качества показывает характерные свойства системы. Вот, для примера, несколько факторов качества верхнего уровня: корректность, надёжность, эффективность, тестируемость, сопровождаемость и пригодность к повторному использованию.

Критерий качества является атрибутом фактора качества, связанным в свою очередь с разработкой программного обеспечения. Например, модульность – это атрибут архитектуры программной системы.

Программа, состоящая из большого количества модулей, позволяет проектировщикам собрать образующие единое целое компоненты в один модуль, таким образом, улучшая сопровождаемость системы.



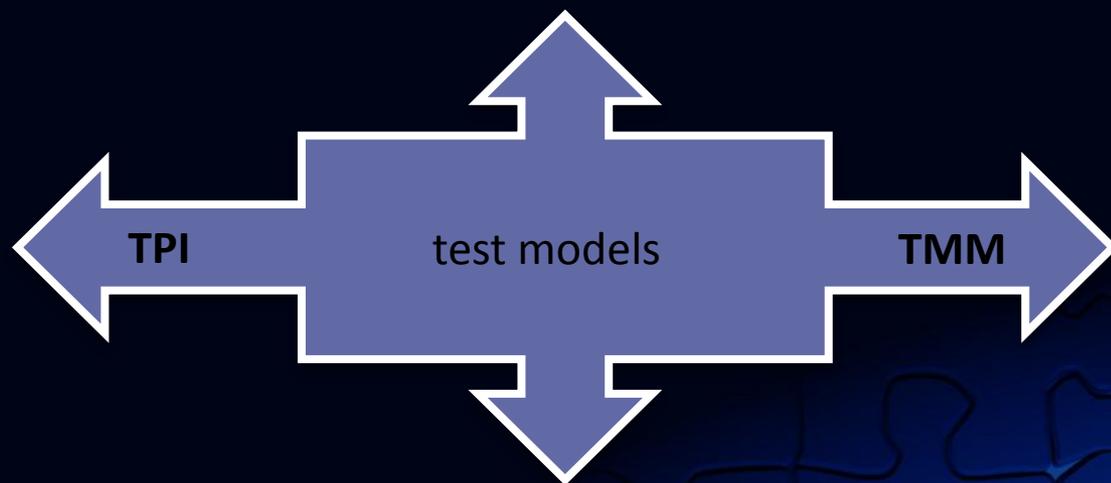
Various software quality models have been proposed to define quality and its related attributes. The most influential ones are the ISO 9126 [20-22] and the CMM [14]. The ISO 9126 quality model was developed by an expert group under the aegis of the International Organization for Standardization (ISO). The document ISO 9126 defines six broad, independent categories of quality characteristics: functionality, reliability, usability, efficiency, maintainability, and portability. The CMM was developed by the Software Engineering Institute (SEI) at Carnegie Mellon University. In the CMM framework, a development process is evaluated on a scale of 1-5, commonly known as level 1 through level 5. For example, level 1 is called the initial level, whereas level 5—optimized—is the highest level of process maturity.



Предлагаются различные модели для определения качества и связанных с ними атрибутов. Наиболее важными из них являются ISO 9126 и CMM [модель зрелости процессов разработки ПО]. Модель определения качества ISO 9126 была разработана группой экспертов под эгидой Международной Организации по Стандартизации (ISO). Документ ISO 9126 даёт определение общим, независимым категориям характеристик качества: функциональность, надёжность, простота использования, эффективность, сопровождаемость и переносимость.

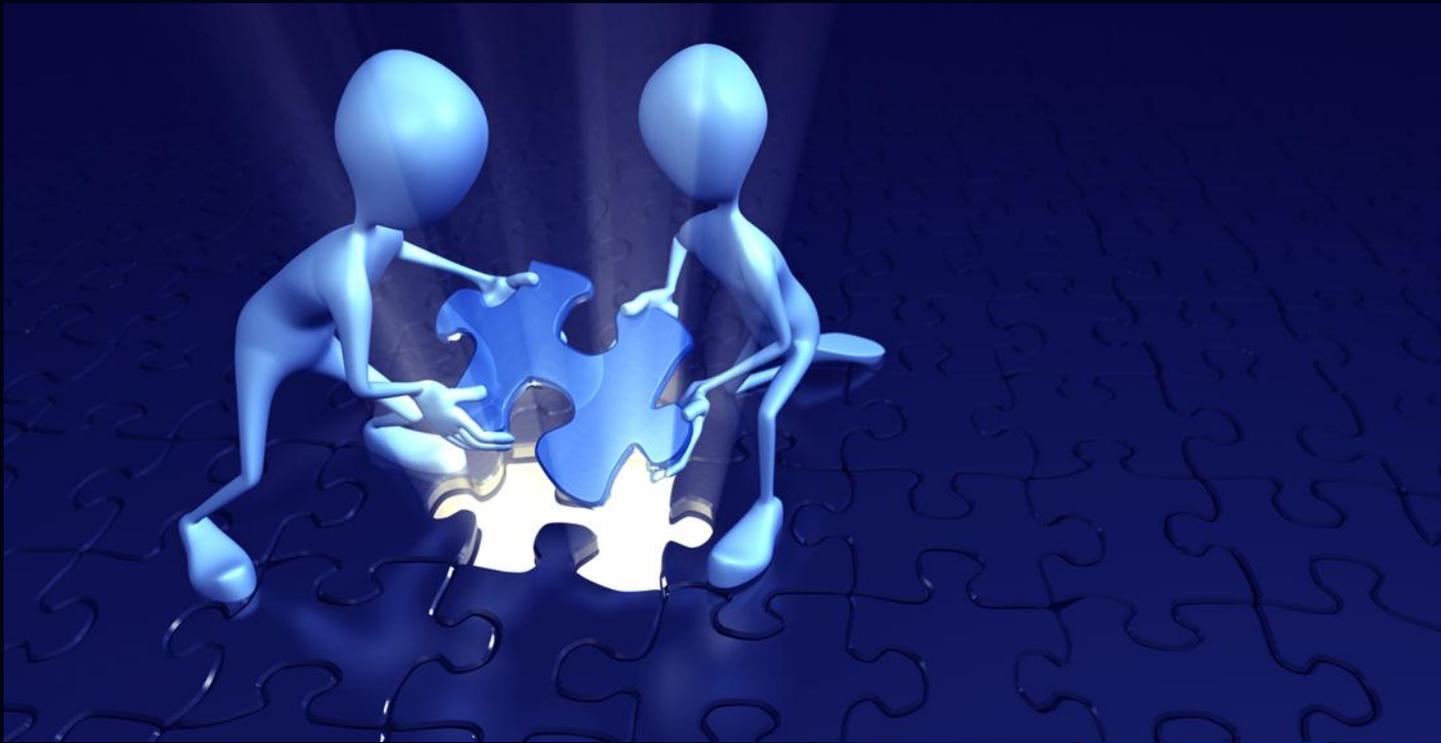
CMM была разработана Институтом Программной Инженерии (SEI) при Университете Карнеги Меллон. Согласно CMM структуре, процесс разработки определяется по шкале от 1 до 5, известной как иерархия уровней зрелости от 1 до 5. Например, уровень 1 называется начальным уровнем, тогда как уровень 5 – оптимизированный – самый верхний уровень зрелости процесса разработки программного обеспечения.

In the field of software testing, there are two well-known process models, namely, the test process improvement (TPI) model [23] and the test maturity Model (TMM) [24]. These two models allow an organization to assess the current state of their software testing processes, identify the next logical area for improvement, and recommend an action plan for test process improvement.



В области тестирования ПО существуют две широко известные модели, модель улучшения процесса тестирования (TPI) и модель зрелости процессов тестирования (TMM). Эти две модели позволяют оценить текущее состояние процесса, рекомендовать план действий для улучшения тестирования и обозначить следующую логическую область для работы.





Thank You

