

Нефункциональные требования к программному обеспечению

Наталья Желнова

Вера Иванова

Содержание

- Классификация нефункциональных требований
- Шаблоны нефункциональных требований
- Численные значения нефункциональных требований
- Связи между нефункциональными и функциональными требованиями
- Влияние различных категорий нефункциональных требований друг на друга
- Атрибуты качества продукта и нефункциональные требования
- Роли в проекте, с которыми взаимодействует аналитик при выявлении и уточнении нефункциональных требований

Термины и определения

- **Определение требований**

- Условие или возможность, требуемая пользователем для решения задач или достижения целей
- Условие или возможность, которые должны удовлетворяться системой/компонентом системы или которыми система/компонент системы должна обладать для обеспечения условий контракта, стандартов, спецификаций или др. регулирующими документами
- Документальная репрезентация (зафиксированное представление, определение, описание) условий или возможностей, перечисленных в предыдущих пунктах

Термины и определения

- **Типы требований**

- *Потребности (needs)* – отражают проблемы бизнеса, персоналии или процесса, которые должны быть соотнесены с использованием системы
- *Функциональные требования*
- *Нефункциональные требования*
- *Системные требования*

Термины и определения

- **Функциональные требования**
 - Условие или возможность, требуемая пользователем для решения задач или достижения целей
 - Условие или возможность, которые должны удовлетворяться системой/компонентом системы или которыми система/компонент системы должна обладать для обеспечения условий контракта, стандартов, спецификаций и др. регулирующих документов

Термины и определения

- **Нефункциональные требования**

- *Бизнес-правила (Business Rules)* – основываются на корпоративных регламентах, политиках, стандартах, законодательных актах, алгоритмах, etc.
- *Внешние интерфейсы* – описание аспектов взаимодействия с другими системами, операционной средой
- *Атрибуты качества (Quality Attributes)* – дополнительные характеристики продукта, важные для пользователей и/или разработчиков (переносимость на другие платформы, оперативная совместимость, целостность, устойчивость, etc.)
- *Ограничения (Constraints)* – условия, ограничивающие выбор возможных решений по реализации отдельных требований или их наборов
- *Предложения по реализации* – предложения, оценивающие возможность использования определенных технологических и архитектурных решений
- *Предложения по тестированию разрабатываемого ПО* – дополнения к требованиям, указывающие, каким образом то или иное требование должно быть протестировано
- *Юридические требования* – требования к лицензированию, патентной чистоте, etc.

Термины и определения

- **Системные требования**
 - Высокоуровневые требования к программному обеспечению, содержащему несколько или много взаимосвязанных подсистем и приложений

Вигерс Карл. Разработка требований к программному обеспечению

[610.12-1990 - IEEE Standard Glossary of Software Engineering Terminology](#)

Типы нефункциональных требований

Нефункциональные
требования

Требования к
продукту

Организационные
требования

Внешние
требования

Требования к
эксплуатации

Требования к
надежности

Требования к
переносимости

Требования к
взаимодействи
ю

Требования к
эффективности

Требования к
реализации

Стандарты

Юридические
требования

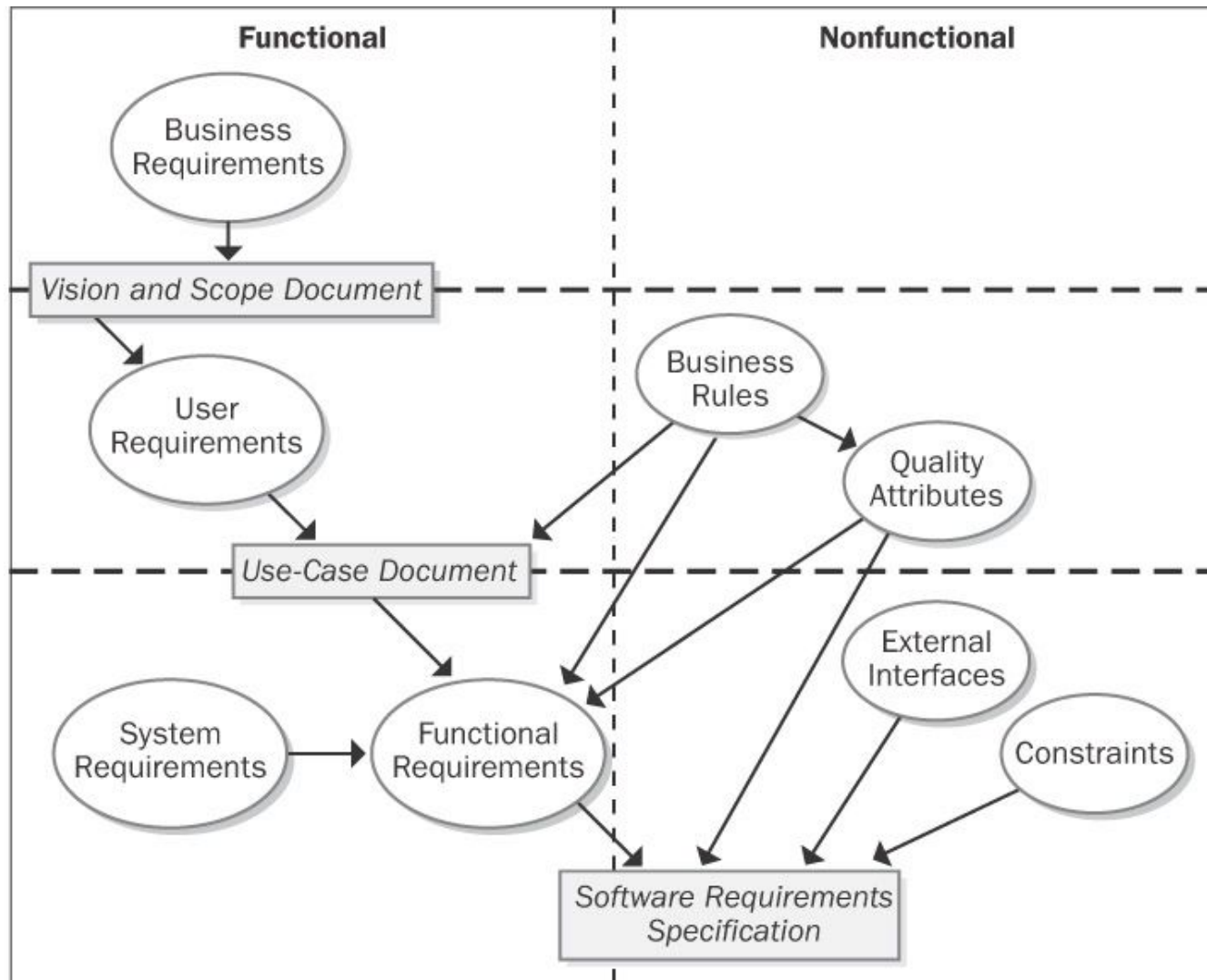
Требования к
производительности

Требования к
памяти

Требования о
сохранении
конфиденциальности

Требования по
технике
безопасности

Состав спецификаций



Модель качества ПО

Характеристики качественного ПО

- Легко использовать
- Хорошая производительность
- Нет ошибок
- Не портит пользовательские данные при сбоях
- Можно использовать на разных платформах
- Может работать 24 часа в сутки и 7 дней в неделю
- Легко добавлять новые возможности
- Удовлетворяет потребности пользователей
- Хорошо документировано
- etc.

Создание модели качества ПО

- **Определение заинтересованных лиц**
- **Определение критериев качества**
- **Нахождение решения,
удовлетворяющего критериям**

Модели качества программного обеспечения

- [ISO 9126](#)
- [ГОСТ 34](#)
- McCall's Quality Model (1977)
- Boehm's Quality Model (1978)

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

ISO 8402:1994 Quality management and quality assurance

Модель качества ISO 9126

Оценка качества ПО основана на трехуровневом рассмотрении:

- **Цели** (goals) — то, что мы хотим видеть в ПО
- **Атрибуты** (attributes) — свойства ПО, показывающие приближение к целям
- **Метрики** (metrics) — количественные характеристики степени наличия атрибутов

Выделено 6 целей:

- функциональность (functionality)
- надежность (reliability)
- практичность или удобство использования (usability)
- эффективность (efficiency)
- сопровождаемость (maintainability)
- переносимость или мобильность (portability).

Характеристики качества ПО (ISO 9126)

Наименование характеристики	Набор атрибутов
Функциональность	Пригодность к определенной работе(suitability) Точность, правильность (accuracy) Способность к взаимодействию (interoperability) Соответствие стандартам и правилам (compliance) Защищенность (security)
Надёжность	Зрелость, завершенность (обратна к частоте отказов) (maturity) Устойчивость к отказам (fault tolerance) Способность к восстановлению работоспособности при отказах (recoverability) Доступность Готовность Соответствие стандартам надежности (reliability compliance, добавлен в 2001)

Характеристики качества ПО (ISO 9126)

Наименование характеристики	Набор атрибутов
Практичность, удобство использования	Понятность (understandability) Удобство обучения (learnability) Работоспособность (operability) Привлекательность (attractiveness, добавлен в 2001) Соответствие стандартам практичности (usability compliance, добавлен в 2001)
Эффективность	Временные характеристики (time behaviour) Использование ресурсов (resource utilisation) Соответствие стандартам эффективности (efficiency compliance, добавлен в 2001)

Характеристики качества ПО (ISO 9126)

Наименование характеристики	Набор атрибутов
Сопровождаемость	Анализируемость (analyzability) Изменяемость, удобство внесения изменений (changeability) Риск возникновения неожиданных эффектов при внесении изменений (stability) Контролируемость, удобство проверки (testability) Соответствие стандартам сопровождаемости (maintainability compliance, добавлен в 2001)
Переносимость	Адаптируемость (adaptability) Устанавливаемость, удобство установки (installability) Способность к сосуществованию с другим ПО (coexistence) Удобство замены другого ПО данным (replaceability) Соответствие стандартам переносимости (portability compliance, добавлен в 2001)

Характеристики качества

- **Функциональность** – способность решать задачи, которые соответствуют зафиксированным и предполагаемым потребностям пользователя, при заданных условиях использования
- **Надежность** – способность выполнять требуемые задачи в обозначенных условиях на протяжении заданного промежутка времени или указанное количество операций
- **Удобство использования** – возможность легкого понимания, изучения, использования и привлекательности ПО для пользователя

Характеристики качества

- **Эффективность** – способность обеспечивать требуемый уровень производительности в соответствии с выделенными ресурсами, временем и другими обозначенными условиями
- **Удобство сопровождения** – легкость, с которой ПО может анализироваться, тестироваться, изменяться для исправления дефектов, для реализации новых требований, для облегчения дальнейшего обслуживания и адаптироваться к изменяющемуся окружению
- **Переносимость** – характеризует ПО с точки зрения легкости его переноса из одного окружения (software/hardware) в другое

Основные характеристики качества ПО (ISO 9126)

- **Системная эффективность** — применение программного продукта по назначению
- **Продуктивность** — производительность при решении основных задач системы, достигаемая при реально ограниченных ресурсах в конкретной внешней среде применения
- **Безопасность** — надежность функционирования комплекса программ и возможный риск от его применения для людей, бизнеса и внешней среды
- **Удовлетворение требований и затрат пользователей** в соответствии с целями применения системы

Модель качества ПО по МакКолу

Характеристики качества:

- **Факторы** (factors), описывающие ПО с позиций пользователя и задаваемые требованиями
- **Критерии** (criteria), описывающие ПО с позиций разработчика и задаваемые как цели
- **Метрики** (metrics), используемые для количественного описания и измерения качества

Критерии качества ПО по МакКолу

Критерии качества — числовые уровни факторов, поставленные в качестве целей при разработке:

- Удобство проверки на соответствие стандартам (auditability)
- Точность управления и вычислений (accuracy)
- Степень стандартности интерфейсов (communication commonality)
- Функциональная полнота (completeness)
- Однородность используемых правил проектирования и документации (consistency)
- Степень стандартности форматов данных (data commonality)
- Устойчивость к ошибкам (error tolerance)
- Эффективность работы (execution efficiency)
- Расширяемость (expandability)

Критерии качества ПО по МакКолу

- Широта области потенциального использования (generality)
- Независимость от аппаратной платформы (hardware independence)
- Полнота протоколирования ошибок и других событий (instrumentation)
- Модульность (modularity)
- Удобство работы (operability)
- Защищенность (security)
- Самодокументированность (selfdocumentation)
- Простота работы (simplicity)
- Независимость от программной платформы (software system independence)
- Возможность соотнесения проекта с требованиями (traceability)
- Удобство обучения (training)

Критерии качества ПО по Боэму

Дополнительные атрибуты качества по Боэму:

- ясность (clarity),
- удобство внесения изменений (modifiability),
- документированность (documentation),
- способность к восстановлению функций (resilience),
- понятность (understandability),
- адекватность (validity),
- функциональность (functionality),
- универсальность (generality),
- экономическая эффективность (economy)

ГОСТ 34 (с дополнениями)

Runtime (атрибуты, относящиеся ко времени работы приложения или системы):

- Доступность
- Надежность
- Требования к времени хранения данных
- Масштабируемость
- Требования к удобству использования
- Требования к безопасности
- Требования к конфигурируемости
- Требования к производительности
- Ограничения

ГОСТ 34 (с дополнениями)

Design time (атрибуты, определяющие ключевые аспекты проектирования приложения или системы):

- Требования к повторному использованию *реализации* или *компонентов* приложения/системы
- Требования к расширяемости
- Требования к переносимости
- Требования к взаимодействию
- Требования к поддержке
- Требования к модульности
- Требования к возможности тестирования
- Требования к возможности и простоте локализации
- Требования к совместимости между версиями приложений

См. [Нефункциональные требования к ПО](#)

Численные характеристики

Доступность (отказоустойчивость)

- Частота недоступности системы в пределах временного интервала, который используется для определения доступности
- Продолжительность недоступности системы
- Доступность по расписанию
 - 5 x 8 (рабочие дни, рабочие часы)
 - 7 x 24 (все дни недели, 24 часа)
 - 365 x 24 (все дни года, 24 часа)

Доступность пять «9 » или 99,999% - стремление индустрии

Например, производители серверов:

Достигнутый результат – 99,998% для кластеров (10 минут недоступности в течение года)

ПК – 97,5% доступности в среднем (219 часов в год)

Численные характеристики

Классификация по уровню требуемой непрерывности обслуживания и важности для бизнеса

- Mission Critical- системы, работающие в режиме «боевого дежурства».
- Business Critical– системы, критические для управления, с режимом работы 24x7x365. Рекомендованное время восстановления подобных систем после отказа менее 2 часов.
- Business Operational - обычные бизнес-приложения - системы, не требующие работы в реальном времени, с режимом работы 8x5. Рекомендованное время восстановления подобных систем после отказа 4-6 часов.
- Office Production - не критические для управления приложения, персональные данные. Рекомендованное время восстановления подобных систем после отказа 1-2 рабочих дня.

Численные характеристики

Надежность и доступность

- Операционная мера надежности – MTTF (Mean Time To Failure – среднее время до отказа или наработка на отказ).
Измеряется в часах
- Частота отказов: $(1 / \text{MTTF})$
- Среднее время на устранение отказа – MTTR (Mean Time To Repair)

Численные характеристики

Связь между уровнем дефектов и значениями МТТФ

Дефектов на KLOC	МТТФ
Больше 30	Меньше 2 минут
20-30	4-15 минут
10-20	5-60 минут
5-10	1-4 часа
2-5	4-24 часа
1-2	24-160 часов
Меньше 1	Не определено

Численные характеристики

Надежность и доступность (промышленные средние)

- Среднее число ошибок в бизнес - системах, найденное в течение первого года эксплуатации:
 - США 4,44/KLOC
 - Япония 1,96/KLOC
- Среднее число ошибок в ПО
 - CMMI уровень 1 7,38/KLOC
 - CMMI уровень 3 1,30/KLOC
- Число ошибок в системах высокой доступности (99,9%+)
 - Должно быть ниже 0,01/KLOC

Численные характеристики

Производительность

Transaction Processing Performance Council

- Число операций в секунду:
 - Ед. измерения: MIPS – миллионы инструкций в секунду
- Число транзакций в секунду
 - TPC-App для серверов приложений и веб-сервисов
 - TPC-C для операций многих пользователей с базой данных
 - TPC-E – новый OLTP тест. Эмулирует брокерскую компанию с клиентами, которые генерируют торговые транзакции. Компания взаимодействует с финансовыми рынками
 - TPC-H для поддержки принятия решений. Набор произвольных бизнес-запросов и параллельная модификация данных

Численные характеристики

Производительность

- При заданных параметрах системы
 - Число серверов
 - Процессоры
 - Память
 - Дисковая подсистема
 - Сеть
- При заданном объеме базы данных
 - Число записей того или иного сорта, например, число позиций на складе или число счетов в банковской системе или число полисов в страховой системе
- При меняющемся числе параллельно работающих пользователей
 - Например, 1 – 10 – 100 – 1000 – 10000
- Время отклика системы на воздействие
 - Он-лайн запросы
 - Пакетные запросы (отчеты)

Численные характеристики

Производительность

- Необходимо учитывать разные архитектуры
 - Клиент – сервер
 - Клиент – сервер приложений – сервер базы данных
 - Клиент – сервер интерфейса – сервер приложений – сервер базы данных
- Как осуществляется балансировка загрузки
 - Автоматически, средствами сервера приложений, операционной системы, базы данных
 - Алгоритмами приложения

Численные характеристики

Безопасность

- Внешние метрики безопасности:

Метрика	Формула
протоколирование доступа	$X = A / B;$ A = число «фактов доступа пользователя к системе и данным», зафиксированных в протоколе системы; B = число «фактов доступа пользователя к системе и данным», которые были произведены во время оценки;
контролируемость доступа	$X = A / B;$ A = число обнаруженных видов несанкционированного доступа; B = число видов несанкционированного доступа в спецификации;

Численные характеристики

Безопасность

- Внешние метрики безопасности:

Метрика	Формула
предотвращение повреждения данных;	a) $X = 1 - A / N$; A = число фактов существенного повреждения данных; N = число видов тестов, при помощи которых пытались спровоцировать факт повреждения данных; b) $Y = 1 - B / N$; B = число фактов незначительного повреждения данных; c) $X = A / T$ или B / T; T = время выполнения операции;

Численные характеристики

Безопасность

- Внутренние метрики безопасности:

Метрика	Формула
протоколирование доступа	$X = A / B;$ A = число типов доступа, которые были зарегистрированы корректно, как определено в спецификации; B = число типов доступа, которые должны регистрироваться по спецификации;
контроль доступа	$X = A / B;$ A = число требований контроля доступа, реализованных корректно, в соответствии со спецификацией; B = число требований контроля доступа в спецификации;

Численные характеристики

Безопасность

- Внутренние метрики безопасности:

Метрика	Формула
предотвращение повреждения данных	$X = A / B;$ A = число реализованных механизмов защиты от повреждения данных; B = число механизмов, требуемых по спецификации;
криптографическая защита данных	$X = A / B;$ A = число реализованных механизмов; B = число требуемых механизмов по спецификации;

Численные характеристики

Безопасность

- Метрики безопасности качества в

Метрика	Формула
безопасность пользователей и их здоровья	$X = 1 - A / B;$ A = число пользователей, сообщивших о наличии проблем; B = число пользователей;
безопасность людей, задействованных в использовании системы	$X = 1 - A / B;$ A = число людей, подверженных риску; B = число людей, задействованных в использовании продукта;

Численные характеристики

Безопасность

- Метрики безопасности качества в

Метрика	Формула
экономический ущерб	$X = 1 - A / B;$ A = число событий экономического ущерба; B = общее число использования системы;
повреждение прочего ПО	$X = 1 - A / B;$ A = число событий повреждения прочего ПО; B = общее число использования системы;

Численные характеристики

Если точное значение определить невозможно

- Используйте оценочные значения (границы интервалов, за которые нельзя выходить)
- Оценки по порядку величины
 - Например, 1 – 10 – 100 – 1000 – 10000
- Уточняйте требования бизнес-уровня
- Пользуйтесь экспертизой ведущих производителей ПО
 - Benchmark tests
 - Техническая документация (MSDN)

Атрибуты качества: проблемы

Общие проблемы:

- Клиентам трудно их определить, и потому они обычно не упоминаются
 - У разных классов пользователей свои предпочтения.
- Подразумеваются заказчиками
- Существенны и значимы при выборе архитектурного решения
- Должны быть исследованы во время процесса выявления требований при участии всех заинтересованных сторон (а не только пользователей)
- Должны быть измеряемы и проверяемы

Атрибуты качества: рабочие группы

Рабочая группа по определению атрибутов качества:

Создание такой группы – это дополнительный способ выявления основных характеристик систем ПО, смысл которого в привлечении заинтересованных лиц (владельцы проекта).

- Основные особенности рабочей группы по характеристикам качества:
 - системно-ориентирована
 - сфокусирована на заинтересованных лицах
 - созывается до того, как завершено проектирование архитектуры ПО
- Результат включает в себя:
 - исходные сценарии
 - сценарии характеристик качества с приоритетами
 - уточненные сценарии
- Может использоваться для:
 - уточнения требований
 - планирования прототипов для уменьшения рисков
 - проектирования архитектуры

Атрибуты качества: рабочие группы

- Определить список заинтересованных лиц (ЗЛ), с которыми можно провести техническое интервью:
 - Представители бизнес-пользователей
 - Архитекторы ПО
 - Ведущие специалисты по тестированию
 - Менеджеры и аналитики зависимых систем
- Составить опросник с архитектурными требованиями:
 - Несколько вопросов для каждого требования
 - Указать приоритет
- Собрать ответы ЗЛ, проанализировать их на непротиворечивость.

Атрибуты качества: рабочие группы

Сценарий работы группы по определению атрибутов качества:

1. Знакомство и представление рабочей группы
2. Представление бизнес-целей и задач
3. Представление плана архитектуры ПО
4. Определение ведущих элементов архитектуры
5. Сценарий «мозговой штурм»
6. Сценарий «консолидация результатов»
7. Сценарий «задание приоритетов»
8. Сценарий «уточнение»

Атрибуты качества: КОМПРОМИССЫ

	Availability	Efficiency	Flexibility	Integrity	Interoperability	Maintainability	Portability	Reliability	Reusability	Robustness	Testability	Usability
Availability							+		+			
Efficiency			-		-	-	-	-		-	-	-
Flexibility		-		-		+	+	+			+	
Integrity		-			-				-		-	-
Interoperability		-	+	-			+					
Maintainability	+	-	+					+			+	
Portability		-	+		+	-			+		+	-
Reliability	+	-	+			+				+	+	+
Reusability		-	+	-	+	+	+	-			+	
Robustness	+	-						+				+
Testability	+	-	+			+		+				+
Usability		-								+	-	

Связь между функциональными и нефункциональными требованиями

Работа над функциональными требованиями:

- Определение вариантов использования и их декомпозиция
- Определение функциональных областей в системе (общие цели, общие действующие лица)
- Определение критических факторов, влияющих на выполнение сценариев
- Определение действующих ограничений