

## Управление качеством ПО

Алексей Островский

Физико-технический учебно-научный центр НАН Украины

7 мая 2015 г.

# Качество ПО

## Определение

**Качество** — две взаимосвязанных характеристики разрабатываемого программного продукта:

- ▶ мера, в которой ПО соответствует *функциональным требованиям*, заданным планом разработки и прочими документами спецификации (функциональное качество);
- ▶ уровень поддержки *нефункциональных требований*: надежности, производительности, ... (структурное качество).

## Поддержание качества:

- ▶ функциональное качество — с помощью тестирования;
- ▶ структурное качество — с помощью соответствия стандартам и статического анализа кода.

## Оценка качества

- ▶ Были ли в процессе разработки учтены **стандарты кодирования и документирования** ПО?
- ▶ Были ли проведены достаточное **тестирование и / или верификация** ПО?
- ▶ Достаточно ли **надежен** программный продукт?
- ▶ Удовлетворительна ли **скорость работы и время отклика** ПО?
- ▶ Обладает ли ПО достаточно понятным **пользовательским интерфейсом**?
- ▶ Является ли программная система **модульной**? В какой степени она поддерживает **сопровождение**, в частности, внесение изменений?

# Управление качеством

## Определение

**Управление качеством ПО** (англ. *software quality management*) — деятельность, направленная на поддержание уровня качества программной системы.

### Составляющие процессы:

- ▶ **обеспечение качества** (англ. *quality assurance, QA*) — определение стандартов и регламентирующих документов для производства, проверки и оценки программных продуктов;
- ▶ **планирование качества** (англ. *software quality plan*) — применение стандартов и инструментов для поддержания качества в конкретном программном проекте;
- ▶ **контроль качества** (англ. *software quality control*) — соблюдение планирования качества и обеспечения качества во время разработки.

## Планирование качества

### План обеспечения качества:

- ▶ желаемые характеристики качества ПО;
- ▶ методы оценки и достижения характеристик качества.

### Примерное содержание:

- ▶ описание продукта, рынка сбыта, ожидаемых характеристик качества;
- ▶ планы разработки, крайние даты выпуска;
- ▶ описание процессов разработки и используемых стандартов;
- ▶ описание и обоснование важнейших характеристик качества;
- ▶ ключевые риски при разработке, методы их минимизации.

# Стандарты

## Определение

**Стандарты ПО** (англ. *software standard*) — документы, регламентирующие определенные аспекты производства программной системы.

### Категории:

- ▶ **стандарты продуктов** (англ. *product standards*) — стандарты, касающиеся конечных или промежуточных артефактов разработки: структура документов, стандарты оформления кода, содержание комментариев, ...
- ▶ **стандарты процессов** (англ. *process standards*) — описание процессов при разработке ПО: процессы спецификации, проектирования и тестирования, используемые инструменты, ...

## Примеры стандартов

### Стандарты продуктов:

- ▶ структура документа спецификации;
- ▶ формат документирования методов;
- ▶ стандарт оформления кода;
- ▶ формат плана проекта;
- ▶ форма запроса на изменение.

### Стандарты процессов:

- ▶ порядок проведения рецензии на дизайн;
- ▶ процесс выпуска версии ПО;
- ▶ процесс фиксации изменений;
- ▶ процесс внедрения плана проекта;
- ▶ процесс проведения тестирования.

## Стандарт ISO 9001

Стандарт ISO 9001 определяет **категории процессов производства** произвольных продуктов. Соответствие стандарту — определение отношений между конкретными процессами разработки ПО и категориями.

### Процессы поставки продукта:

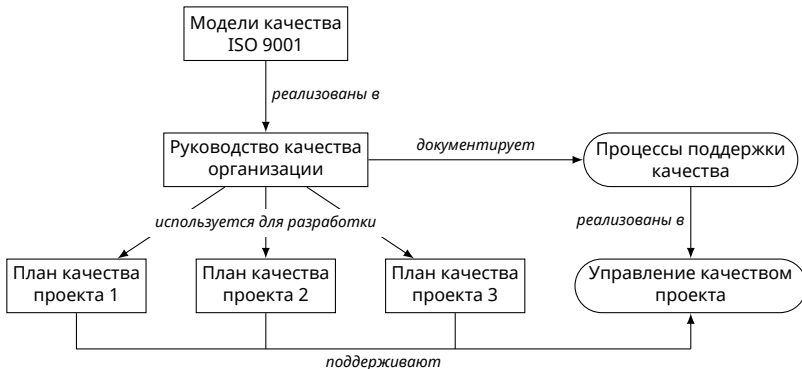
- ▶ приобретение бизнеса (англ. *business acquisition*);
- ▶ проектирование и разработка;
- ▶ тестирование;
- ▶ производство и поставка (англ. *production and delivery*);
- ▶ обслуживание и поддержка.

### Вспомогательные процессы:

- ▶ управление бизнесом;
- ▶ управление ресурсами;
- ▶ управление поставками;
- ▶ управление конфигурацией.



# Стандарт ISO 9001



Использование стандарта ISO 9001 при разработке ПО

# Обзоры и инспекции

## Определение

**Обзоры** (англ. *review*) и **инспекции** (англ. *inspection*) — процессы обеспечения качества, предназначенные для оценки качества артефактов разработки ПО (программ, документации, диаграмм проектирования и т. п.).

**Обзор** — формальный подход: команда разработчиков, проверка на соответствие установленным стандартам.

**Инспекция** — менее формальный подход: кооперация между разработчиками, поиск ошибок и пропусков.

## Области проверки кода

### ▶ Дефекты данных:

- ▶ инициализация переменных перед использованием;
- ▶ отсутствие «магических» констант;
- ▶ индексация элементов массивов / списков;
- ▶ потенциальные случаи переполнения буфера.

### ▶ Дефекты выполнения:

- ▶ проверка условий ветвления;
- ▶ конечность циклов;
- ▶ корректность блоков операций;
- ▶ полнота вариантов и наличие **break** в операторе **switch / case**.

### ▶ Дефекты ввода/вывода:

- ▶ использование всех входных переменных;
- ▶ присвоение выходных переменных.

## Области проверки кода

### ▶ Дефекты интерфейсов:

- ▶ корректное количество и порядок аргументов при вызове функций / методов;
- ▶ соответствие ожидаемых и фактических типов аргументов;
- ▶ идентичность структуры разделяемой памяти.

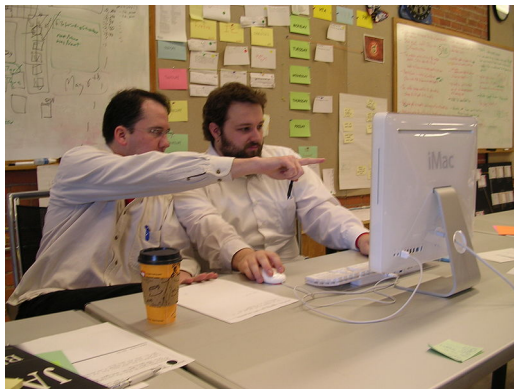
### ▶ Дефекты работы с памятью:

- ▶ корректность работы со связанными объектами (изменение всех требуемых ссылок);
- ▶ корректность выделения / освобождения памяти.

### ▶ Дефекты обработки исключений:

- ▶ полнота проверок возникновения исключительных ситуаций.

## Инспекции в гибкой методологии



В гибкой методологии разработки наряду или вместо обзоров и инспекций используется парное программирование (англ. *pair programming*).

# Парное программирование

## Определение

**Парное программирование** (англ. *pair programming*) — способ разработки ПО, при котором код, написанный первым программистом, немедленно проверяется вторым программистом; альтернатива формальным инспекциям в гибкой методологии разработки.

**Достоинства:** более глубокое понимание кода ⇒ обнаружение большего числа дефектов.

### Недостатки:

- ▶ повышенный шанс непонимания требований;
- ▶ пропуск ошибок из-за высокого темпа разработки;
- ▶ необъективность инспектора.

# Измерение качества

## Определение

**Измерение ПО** — описание аспектов программной системы, отдельных ее компонентов или процессов разработки с помощью математического аппарата.

### Цели измерения ПО:

- ▶ замена дорогостоящих процессов обеспечения качества (напр., обзоров);
- ▶ идентификация компонентов ПО, подлежащих изменению.

**NB.** На сегодняшний день не существует полностью автоматизированных инструментов оценки качества на основе измерений; результаты измерений анализируются человеком.

# Модели качества

## Определение

**Модель качества ПО** — математическая модель, позволяющая оценивать различные характеристики качества на основе анализа программы.

### Уровни модели качества ПО:

1. определение основных и вторичных характеристик качества ПО (надежность, удобство использования, ...);
2. определение измеряемых метрик для каждой характеристики (напр., число ошибок на строку кода);
3. измерение характеристик качества с помощью метрик в статике (анализ структуры программы) и динамике (выполнение программы);
4. агрегация метрик, выделение наиболее важных характеристик качества.



## Характеристики качества



Основные и вторичные характеристики качества ПО согласно стандарту ISO 9126

# Функциональность

## Определение

**Функциональность** (англ. *functionality*) — группа характеристик, оценивающих способность программной системы выполнять функции, заданные с помощью документов спецификации или неявно.

### Составляющие:

- ▶ **функциональная полнота** (англ. *suitability*) — достаточность основных функций ПО для решения задач, определяемых функциональными требованиями;
- ▶ **точность** (англ. *accuracy*) — достижение корректных результатов для всех наборов входных данных;
- ▶ **интероперабельность** (англ. *interoperability*) — возможность взаимодействия с системой посредством внешних интерфейсов (напр., средствами ОС или с помощью сети);
- ▶ **защищенность** (англ. *security*) — способность предотвращать несанкционированный доступ к функциям программы и данным.

# Надежность

## Определение

**Надежность** (англ. *reliability*) — группа характеристик, определяющих способность программной системы выполнять свои функции в течение определенного периода времени.

**NB.** Надежность определяется для программной системы и не зависит от оборудования или мнения пользователей.

### Составляющие:

- ▶ **зрелость** (англ. *maturity*) — способность ПО функционировать без отказов в нормальных условиях;
- ▶ **отказоустойчивость** (англ. *fault tolerance*) — способность ПО функционировать в аномальных условиях (сбой аппаратуры, ошибки в данных и интерфейсах, некорректные действия пользователя, ...);
- ▶ **возобновляемость** (англ. *recoverability*) — способность перезапуска для повторного использования и восстановления данных после отказов ПО.

# Удобство использования

## Определение

**Удобство использования** (англ. *usability*) — группа характеристик, связанных с эргономичностью программного продукта; характеристики, влияющие на выбор ПО определенной группой пользователей для решения конкретных задач.

### Составляющие:

- ▶ **понимаемость** (англ. *understandability*) — оценка усилий, затраченных на выяснение логических концепций и условий применения ПО;
- ▶ **обучаемость** (англ. *learnability*) — оценка усилий, затраченных на определение принципов функционирования ПО с помощью документации, правил, диагностики и т. п.;
- ▶ **работоспособность** (англ. *operability*) — простота работы и управления системой;
- ▶ **привлекательность** (англ. *attractiveness*) — соответствие пользовательского интерфейса программы ожиданиям пользователей.

# Эффективность

## Определение

**Эффективность** (англ. *efficiency*) — группа характеристик, определяющих производительность системы относительно потребляемых ресурсов при заданных условиях.

### Составляющие:

- ▶ **реактивность** (англ. *time behaviour*) — оценка времени отклика, обработки и выполнения функций ПО;
- ▶ **использование ресурсов** (англ. *resource utilization*) — соответствие использования различных видов ресурсов заданным требованиям;
- ▶ **емкость** (англ. *capacity*) — степень соответствия ограничений системы (напр., максимальное количество пользователей в произвольный момент времени) требованиям.

## Удобство сопровождения

### Определение

**Удобство сопровождения** (англ. *maintainability*) — множество характеристик, оценивающих усилия, затрачиваемые на эволюцию ПО: устранение дефектов, совершенствование и адаптацию.

### Составляющие:

- ▶ **простота анализа** (англ. *analyzability*) — оценка усилий для диагностики отказов ПО и идентификации компонентов для изменения;
- ▶ **простота изменений** (англ. *changeability*) — степень, в которой программная система допускает модификацию без внесения дефектов;
- ▶ **стабильность** (англ. *stability*) — степень, в которой программная система допускает внесение изменений без снижения качества существующих компонентов;
- ▶ **простота тестирования** (англ. *testability*) — оценка усилий для разработки критериев тестирования и набора тестов, с помощью которых можно подтвердить выполнение этих критериев.

# Переносимость

## Определение

**Переносимость** (англ. *portability*) — множество характеристик, оценивающих способность программной системы адаптироваться к работе в новом окружении.

### Составляющие:

- ▶ **адаптивность** (англ. *adaptability*) — оценка усилий на адаптацию к другому оборудованию, ПО или другим компонентам окружения;
- ▶ **простота установки** (англ. *installability*) — эффективность установки и удаления программной системы в заданной среде;
- ▶ **сосуществование** (англ. *co-existence*) — эффективность работы программной системы при разделении среды выполнения и ее ресурсов с другим ПО;
- ▶ **заменяемость** (англ. *replaceability*) — эффективность замены аналогичного программного продукта в заданной среде выполнения.

# Метрики качества

## Определение

**Метрика ПО** (англ. *software metric*) — характеристика программной системы, артефактов разработки (напр., системной документации) или процессов разработки, которую можно объективно измерить.

### Примеры метрик:

- ▶ количество строк кода;
- ▶ количество дефектов на строку кода;
- ▶ время выполнения программы;
- ▶ покрытие кода тестами (англ. *code coverage*);
- ▶ цикломатическая сложность (англ. *cyclomatic complexity*).



## Категории метрик ПО

### По объекту измерения:

- ▶ метрики продукта, оценивающие характеристики ПО или артефактов производства (напр., документации);
- ▶ метрики процессов, оценивающие характеристики процессов разработки;
- ▶ метрики использования, оценивающие свойства ПО при работе с конечным пользователем.

### По видимости:

- ▶ внешние метрики, описывающие свойства ПО, доступные для конечного пользователя;
- ▶ внутренние метрики, описывающие свойства, доступные только разработчикам.

### По способу измерения:

- ▶ динамические (требующие выполнения программы);
- ▶ статические, получаемые анализом проекта ПО, кода или документации.

## Метрики продукта

### Примеры общих метрик продукта:

- ▶ **коэффициенты разветвления** (англ. *fan-in / fan-out*) — количество функций, вызывающих определенную функцию (*fan-in*); количество функций, которые вызывает определенная функция (*fan-out*);
- ▶ **размер кода** для отдельных функций / методов, классов, компонентов;
- ▶ **цикломатическая сложность** (англ. *cyclomatic complexity*) — количество линейно независимых маршрутов через граф потока управления программы;
- ▶ **длина идентификаторов** — средняя длина идентификаторов в исходном коде (локальные переменные, названия классов, методов, ...);
- ▶ **глубина вложенных конструкций** — максимальное количество вложенных конструкций ветвления / цикла.

## Метрики продукта

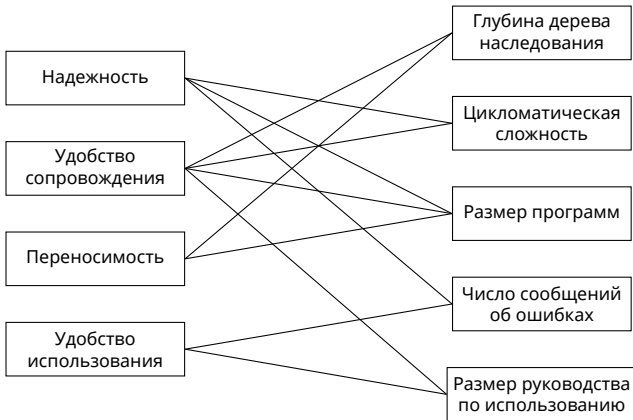
### Примеры метрик продукта в ООП [Chidamber, Kemerer, 1994]:

- ▶ **взвешенное количество методов** (англ. *weighted methods per class*) — сумма весов методов класса (вес метода определяется его сложностью);
- ▶ **глубина дерева наследования** (англ. *depth of inheritance tree*) — количество уровней в графе, представляющем отношение наследования;
- ▶ **число потомков** (англ. *number of children*) — среднее число непосредственных потомков для класса;
- ▶ **связь между классами** (англ. *coupling between object classes*) — мера, в которой методы одного класса используют методы другого класса;
- ▶ **реагирование** (англ. *response for a class*) — количество методов, которые могут быть вызваны при обработке определенного сообщения.

# Метрики и характеристики качества

## Характеристики качества

## Метрики



Связь метрик ПО с характеристиками качества

# Процесс измерения

## Этапы измерения ПО:

1. Выбор необходимых измерений в соответствии с заданной целью.
2. Выбор компонентов, для которых производится измерение (напр., репрезентативная выборка или ключевые компоненты).
3. Измерение характеристик компонентов с помощью автоматических утилит.
4. Идентификация аномальных измерений путем сравнения данных для различных компонентов и с результатами предыдущих измерений.
5. Анализ аномальных компонентов в соответствии с замеренными данными.

## Выводы

1. Качество программного продукта означает его соответствие функциональным и нефункциональным требованиям.
2. Управление качеством — отдельный аспект производства ПО, включающий подбор стандартов, разработку плана поддержки качества и контролирование качества продуктов производства.
3. Контроль качества осуществляется с помощью тестирования и инспекций программ. В гибкой методологии инспекциям соответствуют неформальные методы контроля, такие как парное программирование.
4. Альтернативный способ оценки качества — измерение ПО. Измеримые метрики связаны с характеристиками качества.

# Материалы



**Sommerville, Ian**

**Software Engineering.**

Pearson, 2011. — 790 p.



**Лавріщева К. М.**

**Програмна інженерія (підручник).**

К., 2008. — 319 с.

Спасибо за внимание!