

Качество программного обеспечения

2017

Тенденции индустрии разработки ПО

- ▶ Объем программ растет
- ▶ Время разработки новых версий ПО сокращается
- ▶ Все большее число задач решается программно
- ▶ ПО все больше используется при решении критически важных задач
- ▶ Существенная часть ПО является свободной и поставляется "as is"

Известные примеры программных ошибок

- ▶ **США, 1962 год.** Гибель несущего аппарата "Маринер-1".
 - Причина – ошибка в одном символе программы
 - $DO\ 100\ I = 1, 10$
 - $DO100I = 1.10$
- ▶ **США, 1987 год.** Ускоритель Theras-25. Переоблучение пациентов онкокlinik.
 - Причина – ошибка «race condition»
- ▶ **США, 1991 год.** Комплекс Patriot. Погибло 28 чел.
 - Причина – ошибка округления
- ▶ **Европа, 1996 год.** Ракета Ариан-5. Ущерб \$7 млрд.
 - Причина – использование унаследованного кода

Известные примеры программных ошибок

- ▶ **США, 2003 год.** Сбой в энергосистеме (Blackout). Ущерб 7-10 млрд.\$.
 - Причина – ошибка «race condition»
- ▶ **Израиль.** Сбой навигационной системы самолетов F16.
 - Высотомер выдавал значение ≤ 0 .
- ▶ **Голландия, 2000 год.** Остановка доменной печи 29 февраля. Гибель 6 человек.
 - Ошибка в процедуре расчета даты.
- ▶ ...

Что такое качественное ПО?

- ▶ Вариант 1: ПО, в котором отсутствуют ошибки
- ▶ Вариант 2: ПО, соответствующее требованиям

Стандарты качества ПО

- ▶ Международные стандарты
 - ISO/IEC 9126. Software engineering - Product quality
 - ISO/IEC 25000:2011. "Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models"
- ▶ Российские стандарты
 - ГОСТ 28195-89 «Оценка качества программных средств»
 - ГОСТ Р ИСО/МЭК 9126-93 «Оценка программной продукции»
 - ГОСТ Р ИСО/МЭК 25010-2015 Информационные технологии (ИТ). Системная и программная инженерия. Требования и оценка качества систем и программного обеспечения (SQuaRE). Модели качества систем и программных продуктов

Качество ПО

- ▶ ГОСТ Р ИСО/МЭК 9126 (ISO 9126):
 - **Качество ПО** – весь объем признаков и характеристик программной продукции, который относится к её способности удовлетворять установленным и предполагаемым свойствам
 - **Характеристики качества** – набор свойств программной продукции, по которым её качество описывается и оценивается

Характеристики качества по ISO 9126

- ▶ Функциональность (Functionality)
- ▶ Надежность (Reliability)
- ▶ Практичность (Usability)
- ▶ Эффективность (Efficiencies)
- ▶ Сопровождаемость (Maintainability)
- ▶ Мобильность (Portability)

Характеристики качества.

Функциональность

- ▶ **Функциональность** - набор атрибутов характеризующий, соответствие функциональных возможностей ПО набору требуемой пользователем функциональности.
- ▶ Подхарактеристики:
 - Пригодность (соответствие требуемому набору функций)
 - Корректность (правильность, точность)
 - Способность к взаимодействию (с другими компонентами и системами)
 - Согласованность (соответствие стандартам)
 - Защищенность

Характеристики качества.

Надежность

- ▶ **Надежность** - набор атрибутов, относящихся к способности ПО сохранять свой уровень качества функционирования в установленных условиях за определенный период времени.
- ▶ Подхарактеристики:
 - Стабильность (число отказов при ошибках)
 - Устойчивость к ошибкам
 - Восстанавливаемость
 - Доступность/Готовность

Характеристики качества.

Практичность

- ▶ **Практичность** (удобство) - набор атрибутов, относящихся к объему работ, требуемых для исполнения и индивидуальной оценки такого исполнения определенным или предполагаемым кругом пользователей.
- ▶ Подхарактеристики:
 - Понятность (организации)
 - Изучаемость
 - Простота использования
 - Привлекательность

Характеристики качества.

Эффективность

- ▶ **Эффективность** - набор атрибутов, относящихся к соотношению между уровнем качества функционирования ПО и объемом используемых ресурсов при установленных условиях.
- ▶ Подхарактеристики:
 - Временная эффективность
 - Используемость ресурсов

Характеристики качества. Сопровождаемость

- ▶ **Сопровождаемость** - набор атрибутов, относящихся к объему работ, требуемых для проведения конкретных изменений (модификаций).
- ▶ Подхарактеристики:
 - Анализируемость
 - Изменяемость
 - Стабильность
 - Тестируемость

Характеристики качества.

Мобильность

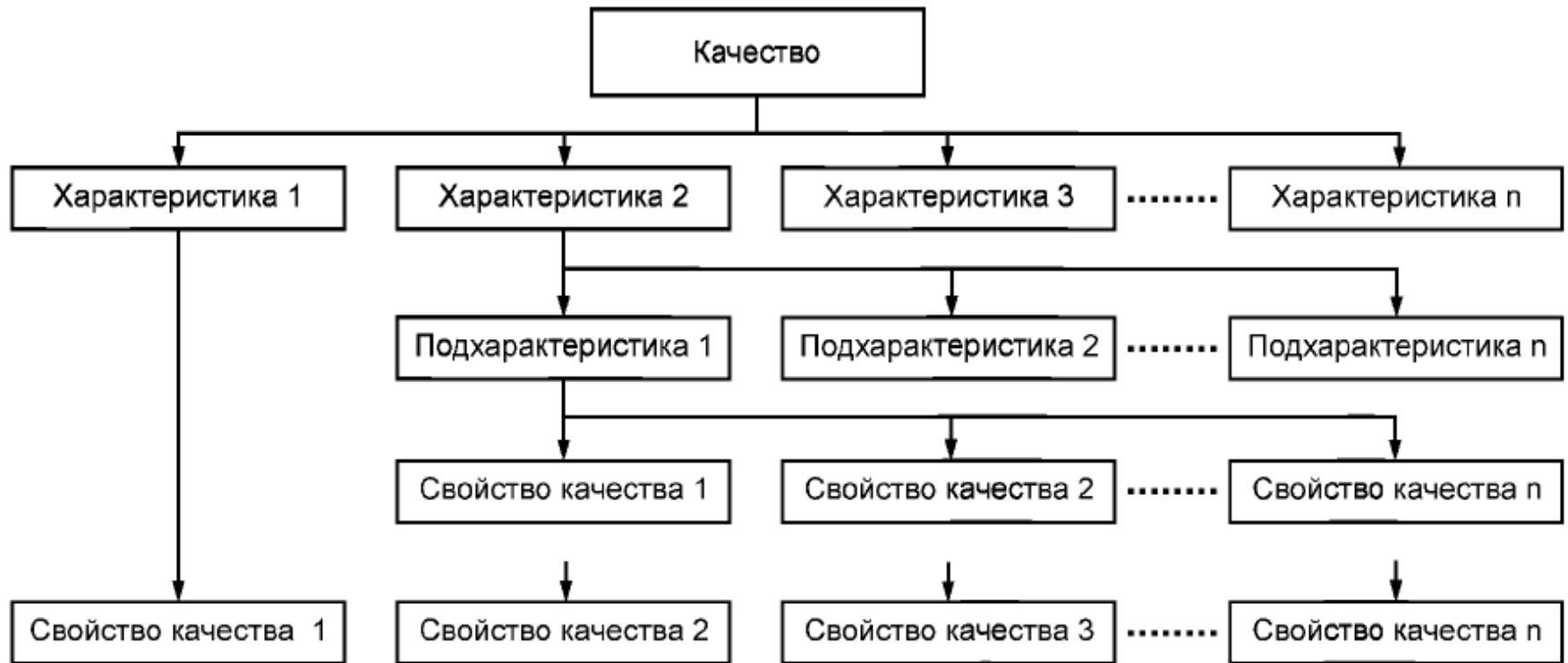
- ▶ **Мобильность** (переносимость)- набор атрибутов, относящихся к способности ПО быть перенесенным из одного окружения в другое.
- ▶ Подхарактеристики:
 - Адаптируемость
 - Простота установки (внедрения)
 - Соответствие стандартам (подчинение стандартам или соглашениям, относящимся к мобильности)
 - Взаимозаменяемость

Современные стандарты качества

- ▶ ГОСТ Р ИСО/МЭК 25010-2015
Информационные технологии (ИТ).
Системная и программная инженерия.
Требования и оценка качества систем и программного обеспечения (SQuaRE).
Модели качества систем и программных продуктов.
- ▶ ISO/IEC 25010:2011 "Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models".

Качество системы

- **Качество системы** - это степень удовлетворения системой заявленных и подразумеваемых потребностей различных заинтересованных сторон



Характеристики качества

- ▶ **Функциональная пригодность** (functional suitability)
- ▶ **Уровень производительности** (performance efficiency)
- ▶ **Совместимость** (compatibility)
- ▶ **Удобство использования** (usability)
- ▶ **Надежность** (reliability)
- ▶ **Защищенность** (security)
- ▶ **Сопровождаемость** (maintainability)
- ▶ **Переносимость** (portability)

Функциональная пригодность

- ▶ Степень, в которой продукт или система обеспечивают выполнение функции в соответствии с заявленными и подразумеваемыми потребностями при использовании в указанных условиях.
- ▶ Подхарактеристики:
 - **функциональная полнота** (functional completeness): степень покрытия совокупностью функций всех определенных задач и целей пользователя
 - **функциональная корректность** (functional correctness): степень обеспечения продуктом или системой необходимой степени точности корректных результатов.
 - **функциональная целесообразность** (functional appropriateness): степень функционального упрощения выполнения определенных задач и достижения целей.

Уровень производительности

- ▶ Производительность относительно суммы использованных при определенных условиях ресурсов.
- ▶ Подхарактеристики:
 - **Временные характеристики** (time behaviour): степень соответствия требованиям по времени отклика, времени обработки и показателей пропускной способности продукта или системы.
 - **Использование ресурсов** (resource utilization): степень удовлетворения требований по потреблению объемов и видов ресурсов продуктом или системой при выполнении их функций.
 - **Потенциальные возможности** (capacity): степень соответствия требованиям предельных значений параметров продукта или системы.

Совместимость

- ▶ Способность обмениваться информацией с другими продуктами, системами или компонентами, и/или выполнять требуемые функции при совместном использовании одних и тех же аппаратных средств или программной среды.
- ▶ Подхарактеристики:
 - **Сосуществование (совместимость)** (co-existence): способность продукта совместно функционировать с другими независимыми продуктами в общей среде с разделением общих ресурсов и без отрицательного влияния на любой другой продукт.
 - **Функциональная совместимость** (интероперабельность) (interoperability): способность двух или более систем, продуктов или компонент обмениваться информацией и использовать такую информацию.

Удобство использования

- ▶ Степень, в которой продукт или система могут быть использованы определенными пользователями для достижения конкретных целей с эффективностью, результативностью и удовлетворенностью в заданном контексте использования.
- ▶ Подхарактеристики:
 - **определимость пригодности** (appropriateness recognizability): возможность пользователей понять, подходит ли продукт или система для их потребностей, сравним ли с функциональной целесообразностью
 - **изучаемость** (learnability): возможность использования продукта или системы определенными пользователями для достижения конкретных целей обучения для эксплуатации продукта или системы с эффективностью, результативностью, свободой от риска и в соответствии с требованиями в указанном контексте использования.
 - **управляемость** (operability): наличие в продукте или системе атрибутов, обеспечивающих простое управление и контроль.
 - **защищенность от ошибки пользователя** (user error protection): уровень системной защиты пользователей от ошибок.
 - **эстетика пользовательского интерфейса** (user interface aesthetics): Степень "приятности" и "удовлетворенности" пользователя интерфейсом взаимодействия с пользователем.
 - **доступность** (accessibility): возможность использования продукта или системы для достижения определенной цели в указанном контексте использования широким кругом людей с самыми разными возможностями.

Надежность

- ▶ Степень выполнения системой, продуктом или компонентом определенных функций при указанных условиях в течение установленного периода времени.
- ▶ Подхарактеристики
 - **завершенность** (maturity): степень соответствия системы, продукта или компонента при нормальной работе требованиям надежности.
 - **готовность** (availability): степень работоспособности и доступности системы, продукта или компонента
 - **отказоустойчивость** (fault tolerance): способность системы, продукта или компонента работать как предназначено, несмотря на наличие дефектов программного обеспечения или аппаратных средств.
 - **восстанавливаемость** (recoverability): способность продукта или системы восстановить данные и требуемое состояние системы в случае прерывания или сбоя.

Защищенность

- ▶ Степень защищенности информации и данных, обеспечиваемая продуктом или системой путем ограничения доступа людей, других продуктов или систем к данным в соответствии с типами и уровнями авторизации.
- ▶ Подхарактеристики:
 - **Конфиденциальность** (confidentiality): обеспечение продуктом или системой ограничения доступа к данным только для тех, кому доступ разрешен.
 - **Целостность** (integrity): степень предотвращения системой, продуктом или компонентом несанкционированного доступа или модификации компьютерных программ или данных.
 - **Неподдельность** (non-repudiation): степень, с которой может быть доказан факт события или действия таким образом, что этот факт не может быть отвергнут когда-либо позже.
 - **Отслеживаемость** (accountability): степень, до которой действия объекта могут быть прослежены однозначно.
 - **Подлинность** (authenticity): степень достоверности тождественности объекта или ресурса требуемому объекту или ресурсу.

Сопровождаемость

- ▶ Результативность и эффективность, с которыми продукт или система могут быть модифицированы предполагаемыми специалистами по обслуживанию.
- ▶ Подхарактеристики:
 - **Модульность** (modularity): степень представления системы или компьютерной программы в виде отдельных блоков таким образом, чтобы изменение одного компонента оказывало минимальное воздействие на другие компоненты
 - **Возможность многократного использования** (reusability): степень, в которой актив может быть использован в нескольких системах или в создании других активов.
 - **Анализируемость** (analysability): степень простоты оценки влияния изменений одной или более частей на продукт или систему или простоты диагностики продукта для выявления недостатков и причин отказов, или простоты идентификации частей, подлежащих изменению.
 - **Модифицируемость** (modifiability): степень простоты эффективного и рационального изменения продукта или системы без добавления дефектов и снижения качества продукта.
 - **Тестируемость** (testability): степень простоты эффективного и рационального определения для системы, продукта или компонента критериев тестирования, а также простоты выполнения тестирования с целью определения соответствия этим критериям.

Переносимость

- ▶ Степень простоты эффективного и рационального переноса системы, продукта или компонента из одной среды (аппаратных средств, программного обеспечения, операционных условий или условий использования) в другую.
- ▶ Подхарактеристики:
 - **Адаптируемость** (adaptability): степень простоты эффективной и рациональной адаптации для отличающихся или усовершенствованных аппаратных средств, программного обеспечения, других операционных сред или условий использования.
 - **Устанавливаемость** (installability): степень простоты эффективной и рациональной, успешной установки и/или удаления продукта или системы в заданной среде.
 - **Взаимозаменяемость** (replaceability): способность продукта заменить другой конкретный программный продукт для достижения тех же целей в тех же условиях.

Качество ПО.

Заинтересованные лица

- ▶ Пользователь/заказчик
 - Функциональная пригодность
 - Надежность
 - Совместимость
 - Удобство использования
 - Уровень производительности
 - Защищенность
- ▶ Разработчик/руководитель
 - Сопровождаемость
 - Переносимость

Оценка качества ПО.

Программометрика

- ▶ Программометрика - наука о количественном оценивании свойств программного обеспечения
- ▶ Программная метрика - мера, позволяющая получить численное значение некоторого свойства программного обеспечения или его спецификаций

Использование метрик для оценки качества

- ▶ Функциональность – метрики тестирования
- ▶ Практичность – метрики эргономики
- ▶ Сопровождаемость – **метрики кода**
- ▶ Мобильность – **метрики кода**
- ▶ Надежность – метрики тестирования, динамические методы
- ▶ Эффективность – только динамические методы

Метрики программного обеспечения

- ▶ Различные системы метрик позволяют оценивать различные характеристики ПО:
 - Локализация
 - Инкапсуляция
 - Информационная закрытость
 - Наследование
 - Абстракция
 - Связность объектов программы
 - Сложность
 - Размер
 - И т.п.

Метрики программного обеспечения

- ▶ Существует множество систем метрик:
 - Метрики Холстеда
 - Метрики Л. Константейна и Э. Йордана
 - Метрики Л. Отта и Б. Мехра
 - Метрики Д. Биемена и Б. Кенга
 - Метрики С. Чидамбера и К. Кемерера
 - Метрики М. Лоренца и Д. Кидда
 - Метрики Ф. Абреу
 - Метрики Р. Байндера
 - И т.п.

Метрики Ф. Абреу (MOOD)

- ▶ Набор метрик MOOD (Metrics of Object-Oriented Design)
- ▶ Разработаны Фернандо Абреу в 1994 г.
- ▶ Цели:
 - Описание ОО-механизмов: инкапсуляция, наследование, полиморфизм, обмен сообщений
 - Формализованность метрик
 - Независимость от размера ПО
 - Независимость от ЯП

Метрики MOOD

- ▶ Фактор закрытости метода (MHF)
- ▶ Фактор закрытости свойства (AHF)
- ▶ Фактор наследования метода (MIF)
- ▶ Фактор наследования свойства (AIF)
- ▶ Фактор полиморфизма (POF)
- ▶ Фактор сцепления (COF)

MOOD. Фактор закрытости метода

- ▶ MHF – Method Hiding Factor
- ▶ Показывает долю скрытых методов в программе
- ▶ $MHF = \sum_{1..N}(Mh_i) / \sum_{1..N}(Mh_i + Mv_i)$
 - Mh_i – число скрытых неунаследованных методов класса i
 - Mv_i – число видимых неунаследованных методов класса i

MOOD. Фактор закрытости свойства

- ▶ AHF – Attribute Hiding Factor
- ▶ Показывает долю скрытых свойств в программе
- ▶ $AHF = \sum_{1..N}(Ah_i) / \sum_{1..N}(Ah_i + Av_i)$
 - Ah_i – число скрытых неунаследованных свойств класса i
 - Av_i – число видимых неунаследованных свойств класса i

MOOD. Фактор наследования метода

- ▶ MIF – Method Inheritance Factor
- ▶ Показывает долю унаследованных непереопределенных методов в программе
- ▶ $MIF = \sum_{1..N}(MI_i) / \sum_{1..N}(MN_i + MI_i + MO_i)$
 - MI_i – число унаследованных непереопределенных методов класса i
 - MN_i – число новых методов класса i
 - MO_i – число унаследованных переопределенных методов класса i

MOOD. Фактор наследования свойств

- ▶ AIF – Attribute Inheritance Factor
- ▶ Показывает долю унаследованных непереопределенных свойств в программе
- ▶ $AIF = \sum_{1..N}(AI_i) / \sum_{1..N}(AN_i + AI_i + AO_i)$
 - AI_i – число унаследованных непереопределенных свойств класса i
 - AN_i – число новых свойств класса i
 - AO_i – число унаследованных переопределенных свойств класса i

MOOD. Фактор полиморфизма

- ▶ POF – Polymorphism factor
- ▶ $POF = \sum_{1..N}(MO_i) / \sum_{1..N}(MN_i * D_i)$
 - MN_i – число новых методов класса i
 - MO_i – число унаследованных переопределенных методов класса i
 - D_i – количество потомков класса i

MOOD. Фактор сцепления

- ▶ COF – Coupling Factor
- ▶ Определяет долю пар классов, связанных отношением «клиент-поставщик»
- ▶ $COF = \sum_{i \in 1..N} \sum_{j \in 1..N} (C_{ij}) / (N \cdot (N-1))$
 - $C_{ij} = 1$, если класс i имеет собственную ссылку на класс j

Метрики сложности

- ▶ Метрики размера программных систем
- ▶ Классические метрики сложности
 - Метрики Холстеда
- ▶ Метрики сложности потока управления
- ▶ Метрики сложности потока данных

Метрики размера

- ▶ LOC – Lines Of Code. Число строк кода. Может вычисляться
 - как общее число всех строк
 - как число операторов программы
- ▶ Среднее число строк в модуле
- ▶ Среднее число строк в функции
- ▶ Степень комментирования программы (LOCC/LOC)
- ▶ Плотность комментирования

Метрики Холстеда

- ▶ Характеристики программы
 - NUOprtr (Number of Unique Operators) — число уникальных операторов программы
 - NUOprnd (Number of Unique Operands) — число уникальных операндов программы (словарь операндов)
 - Noprtr (Number of Operators) — общее число операторов в программе
 - Noprnd (Number of Operands) — общее число операндов в программе
- ▶ Оценки
 - **Словарь программы** (Halstead Program Vocabulary, HPVoc):
 - $HPVoc = NUOprtr + NUOprnd$;
 - **Длина программы** (Halstead Program Length, HPLen):
 - $HPLen = Noprtr + Noprnd$;
 - **Объем программы** (Halstead Program Volume, HPVol):
 - $HPVol = HPLen * \log_2 HPVoc$;
 - **Сложность программы** (Halstead Difficulty, HDiff):
 - $HDiff = (NUOprtr/2) \times (Noprnd / NUOprnd)$;
 - **Усилия программиста при разработке:** (Halstead Effort):
 - $HEff = HDiff \times HPVol$.

Метрики сложности потока управления. Метрика Мак-Кейба

- ▶ Цикломатическая сложность
 - $C = A - V + 2$
 - A – число ребер в ГПУ
 - V- число вершин в ГПУ
 - В общем случае показывает число тестовых прогонов программы, для прохода всех ветвей

Метрики сложности потока данных. Метрики Чепина

- ▶ **Множество «Р»** – входные переменные программы
- ▶ **Множество «М»** – модифицируемые или создаваемые внутри программы переменные
- ▶ **Множество «С»** – переменные, участвующие в управлении работой программного модуля (управляющие переменные)
- ▶ **Множество «Т»** – не используемые в программе переменные
- ▶ **$Q = P + 2M + 3C + 0.5T$**

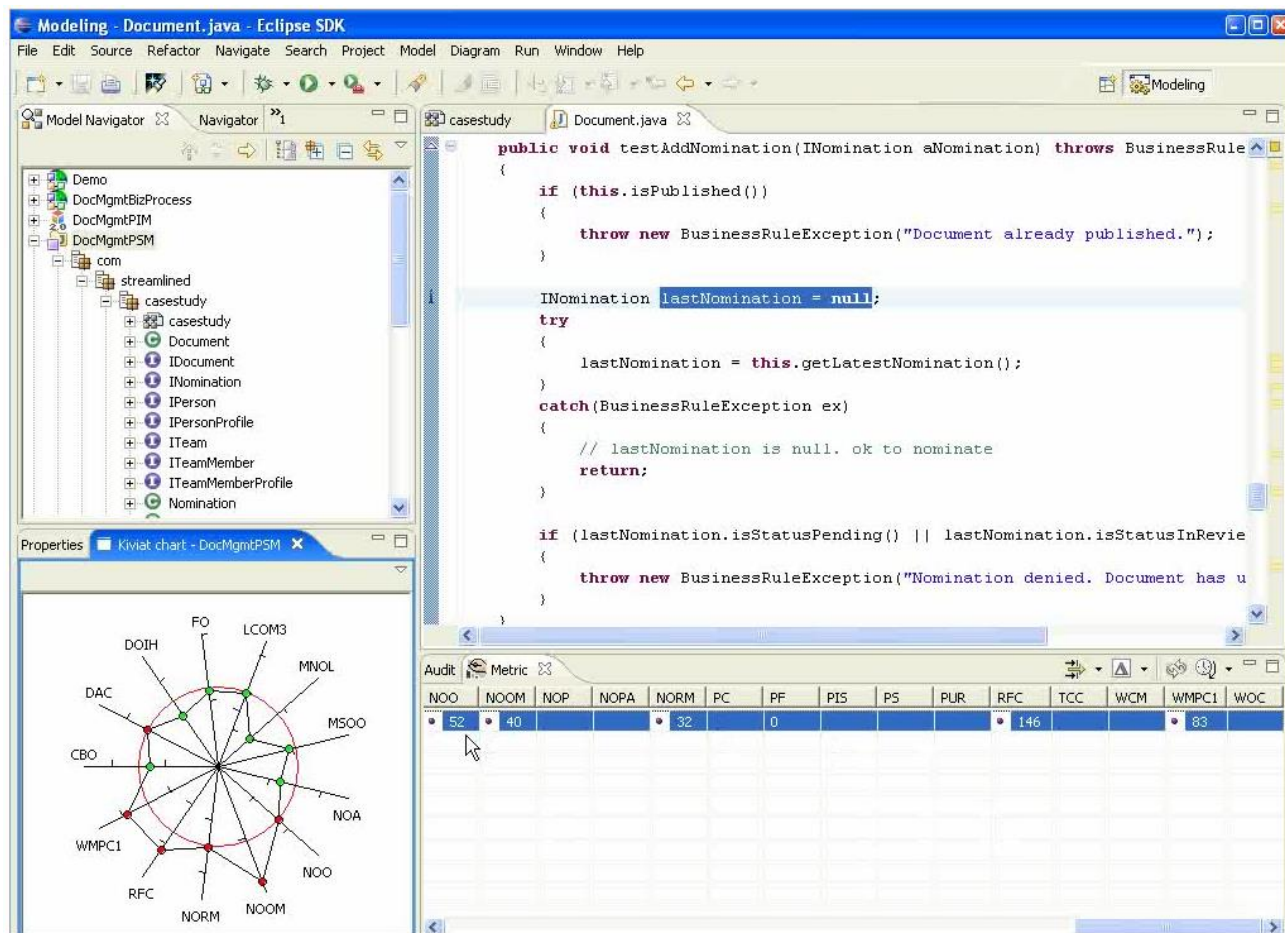
Аудит программного кода

- ▶ Аудит (review) – процесс контроля и оценки программного кода в процессе его эволюции
- ▶ Ручной аудит проводится экспертами в области программирования
- ▶ Автоматизированный аудит проводится на основе программных метрик
- ▶ Автоматизированный аудит является частью многих сред разработки

Использование метрик

- ▶ Для проекта выбирается набор(ы) метрик
- ▶ Для каждой метрики:
 - Если метрика неоднозначна – доопределяется
 - Формируются эталонные значения
 - Значение метрик - нормируются
- ▶ Осуществляется непрерывный аудит программного кода
- ▶ Метрики, значения которых неудовлетворительны, выбираются для анализа
- ▶ Программный код модифицируется с целью улучшения метрик
- ▶ Накапливается статистика
- ▶ * Возможно корректируются эталонные значения метрик

Отображение метрик



Отображение метрик

