



Экспертная система анализа качества исходного кода программного обеспечения

Наталия Вегерина
Инженер-программист,
ООО «Инфострой»
nvegerina@infostroy.com.ua

Александр Липанов
к.т.н., директор ООО «Инфострой»,
доцент кафедры Информатики, ХНУРЭ
alex@infostroy.com.ua

Харьков, Украина

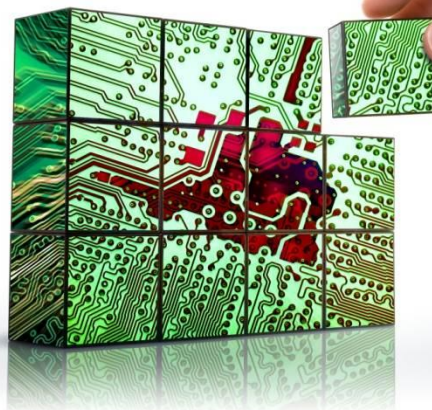


Custom Development Solutions



Содержание

1. Актуальность
2. Обзор существующих приложений
3. Концепция системы анализа и контроля качества исходного кода
4. Метрики
 - 4.1 Сцепление
 - 4.2 Связность
 - 4.3 Нестабильность
5. Экспертная система
6. Программная реализация





Использование метрик

Метрики программного кода являются важным инструментом и уже сегодня используются многими производителями программного обеспечения. Так, при сертификации на более высокие уровни по моделям ISO/IEC или CMM/CMMI использование метрик кода является обязательным, что позволяет в определенной степени достичь контролируемости процесса разработки.



Актуальность

Среди существующих программных продуктов, нацеленных на проведение в той или иной степени оценки качества исходного кода, не существует приложений, предоставляющих возможности интерпретации метрик, с целью предоставления словесных рекомендаций по улучшению исходного кода.



Средства проектирования

- Rational Software Architect;
- Borland Together;
- ArgoUML;
- PowerDesigner.



Приложения для подсчета метрик

- NDepend;
- OxyProject Metrics;
- Resource Standard Metrics;
- Visual Studio 2010;
- CodeMaid;
- CodeSnippets;
- Gmetrics;
- SourceAnalyzer;
- Software Index;
- Resource Standard Metrics.



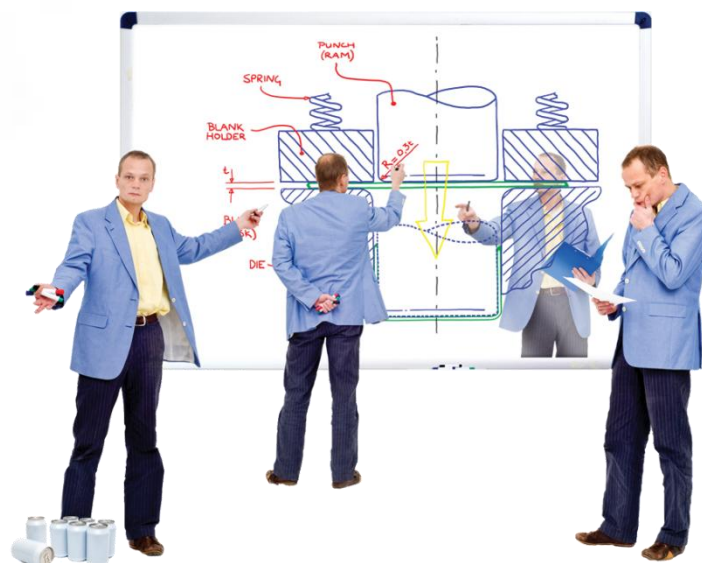
Задачи исследования

1. Провести исследование существующих метрик и выбрать наиболее объективные из них;
2. Нормировать выбранные метрики таким образом, чтобы их значения попадали в промежуток $[0, 1]$ для четкого определения уровня качества исходного кода;
3. Для каждой метрики разбить промежуток $[0, 1]$ на интервалы и поставить в соответствие каждому интервалу некоторую рекомендацию;
4. Реализовать алгоритм подсчета каждой отдельной метрики;
5. Произвести анализ примеров исходного кода экспертами с целью получения оценки каждого примера и рекомендаций по его улучшению.

Концепция системы

В системе должно быть реализовано:

1. Синтаксический анализ исходного кода и преобразование его в структурный вид;
2. Подсчет значений каждой отдельной метрики;
3. Экспертная система, выдающая рекомендации на основе каждой метрики в зависимости от ее значения;
4. Прототип приложения.



Сцепление

Сцепление (coupling) или зависимость – это мера того, насколько программный модуль зависит от каждого из остальных модулей

$$Coupling = r_t \left(\sum_{i=0}^n r_i \right)^{-1}$$

где r_i – количество ссылок на класс, где ссылка – это поле, локальная переменная, возвращаемый тип или параметр метода,

r_t – количество ссылок на класс, участвующий в подсчете метрики,

n – количество классов.



Типы сцепления



Связность

Связность (cohesion) – это мера того, насколько сильно связаны между собой методы модуля приложения.

$$LCOM = 1 - \frac{1}{M * F} \sum_{i=0}^n MF$$

$$LCOM HS = \frac{1}{M - 1} \left(M - \frac{1}{F} \sum_{i=0}^n MF \right)$$

где ***M*** – количество методов в классе (статических и экземплярных конструкторов, геттеров и сеттеров свойств, методов добавления и удаления событий),

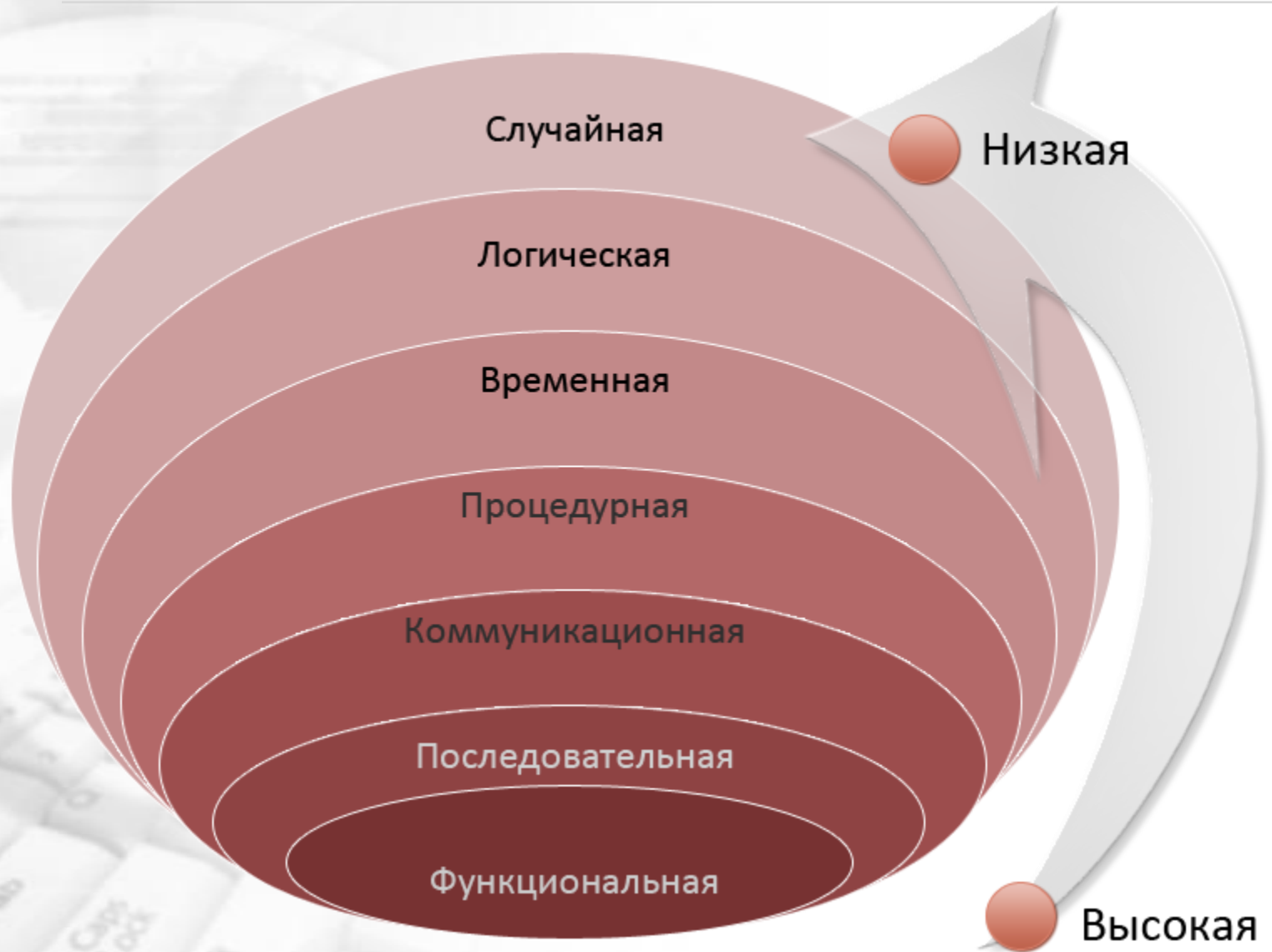
F – количество экземплярных полей класса,

MF – количество методов класса, имеющих доступ к определенному экземплярному полю,

$$\sum_{i=0}^n MF$$

– сумма ***MF*** по всем экземплярным полям класса.

Типы связности



Нестабильность

Нестабильность (instability) – это мера того, насколько зависят классы в пределах модуля.

$$I = \frac{C_e}{C_a + C_e}$$

где C_e – центростремительное сцепление,

C_a – центробежное сцепление.





Система анализа качества исходного кода ПО

Infostroy NVMetrics



Экспертная система

Вход

Значение метрики

База знаний

Анализ значения метрики

Метрика 1 (интервалы)



Метрика N (интервалы)

Формирование рекомендации

Метрика 1 (рекомендации)



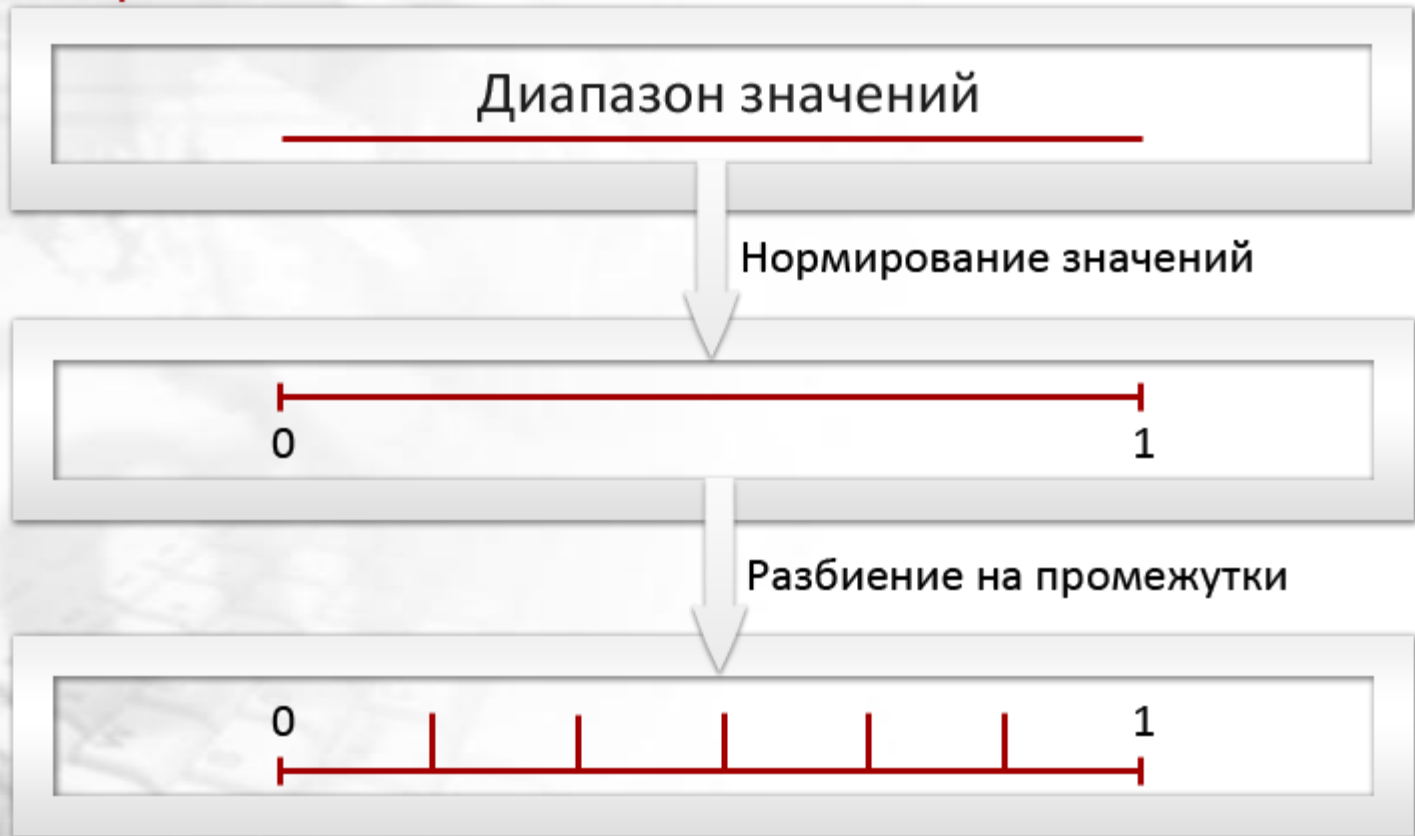
Метрика N (рекомендации)

Выход

Рекомендация

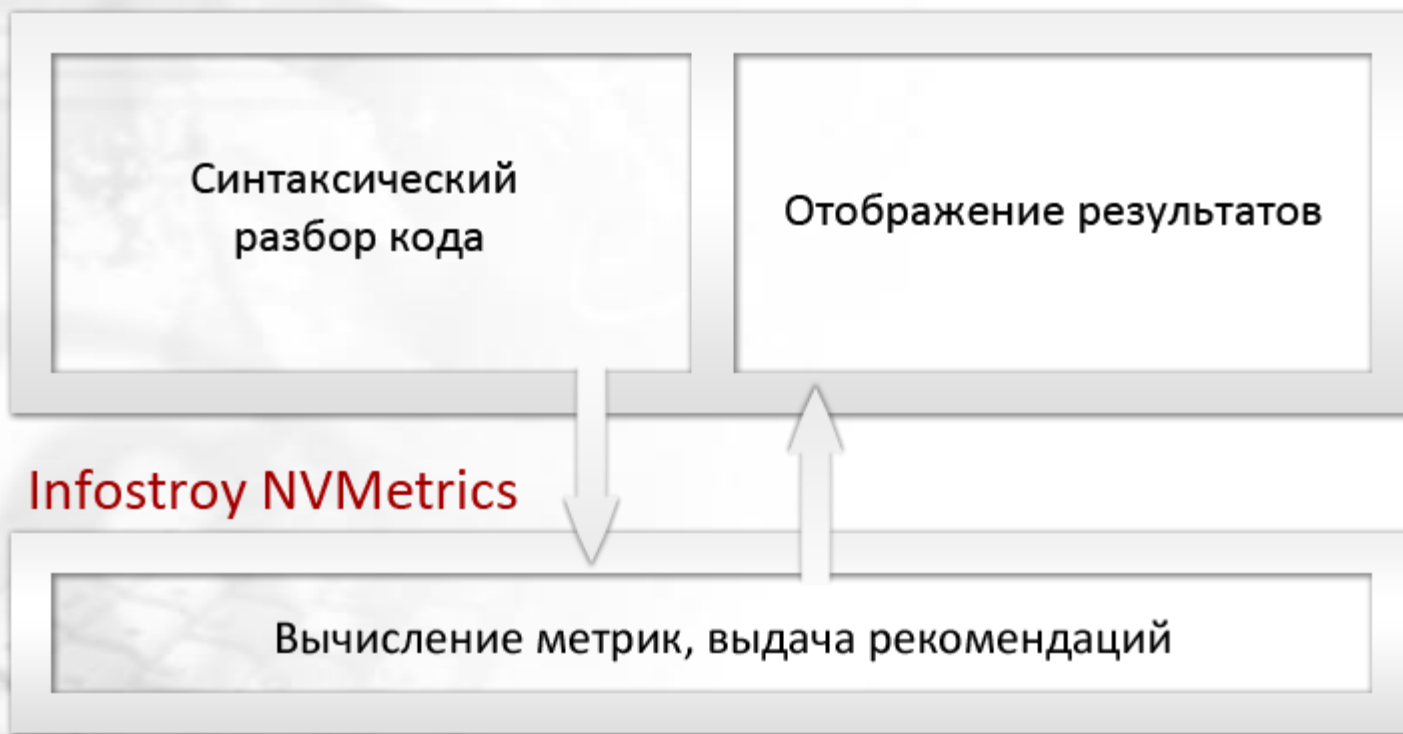
Нормирование метрики

Метрика

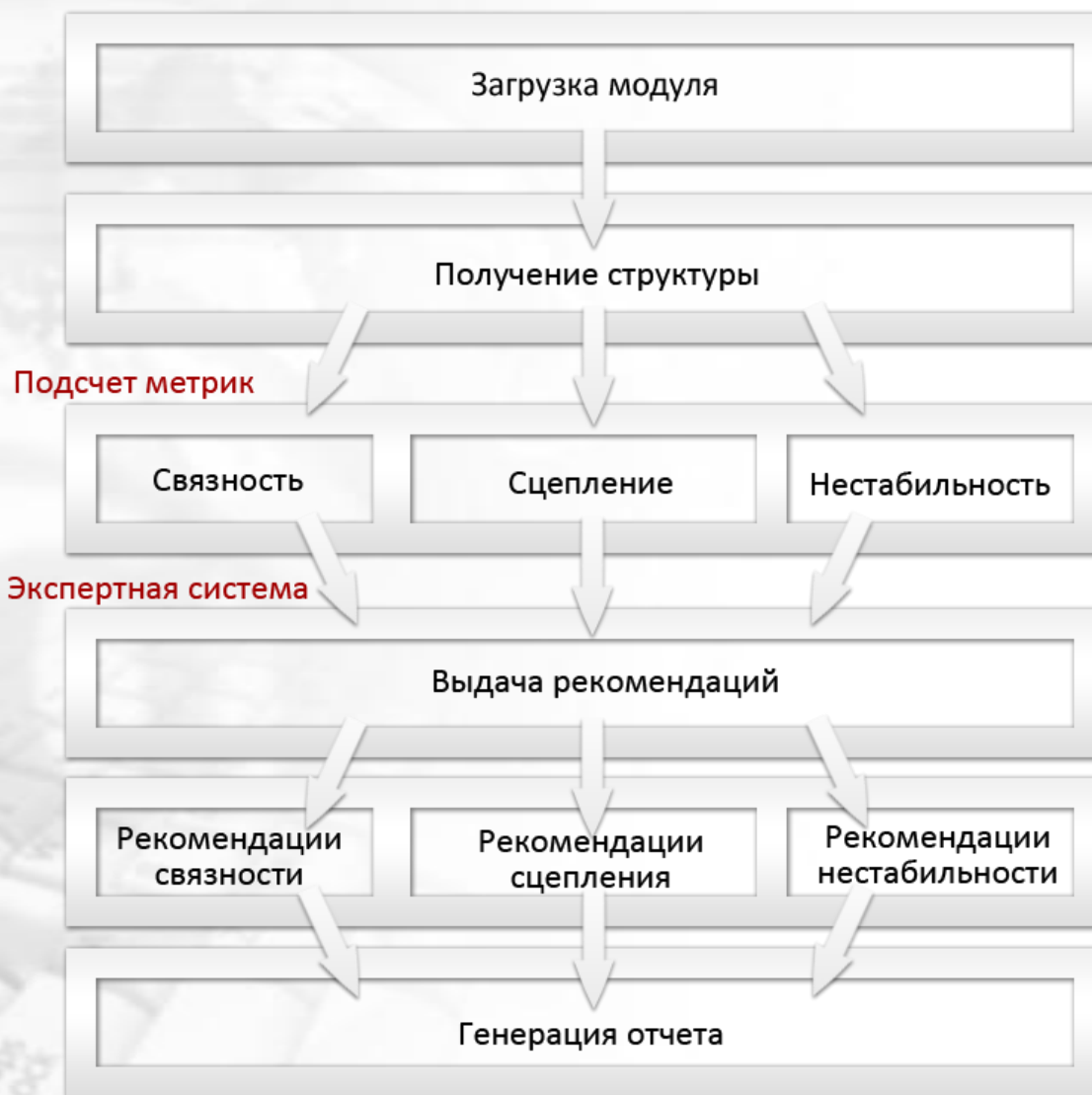


Структура программы

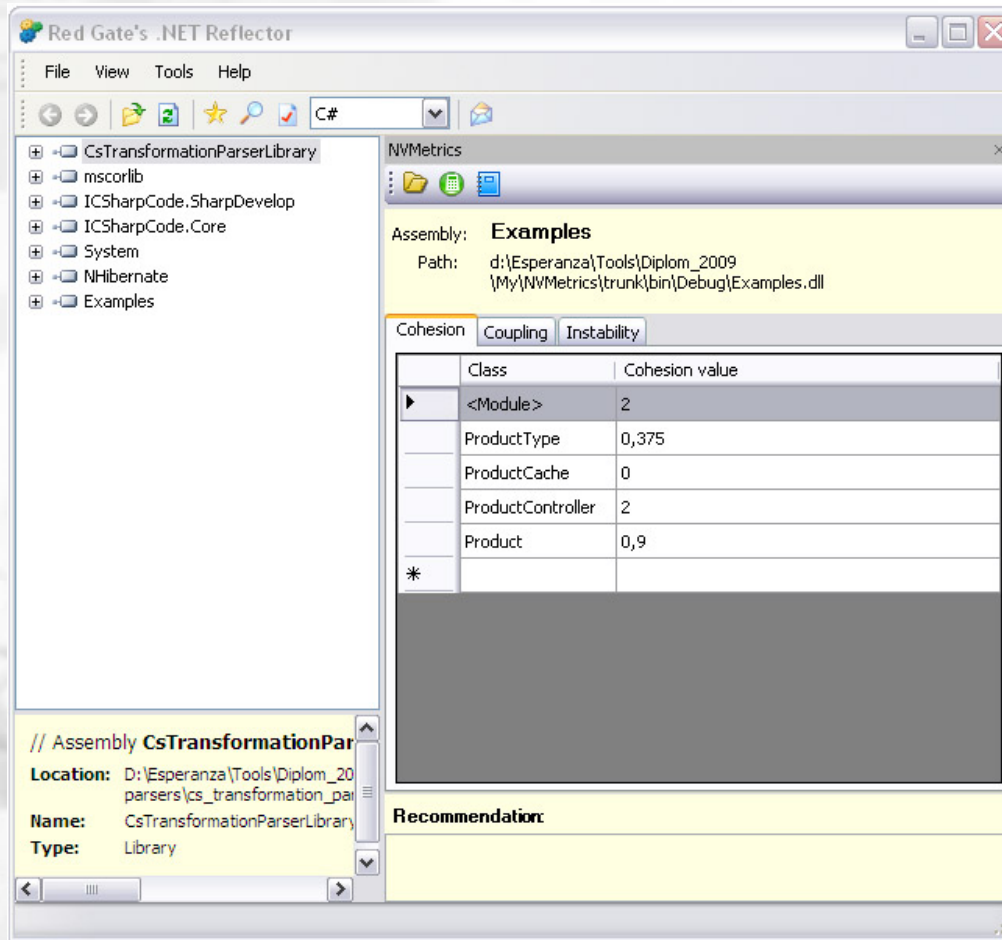
Reflector



Алгоритм работы программы



Главное окно приложения “Infostroy NVMetrics”



Red Gate's .NET Reflector

File View Tools Help

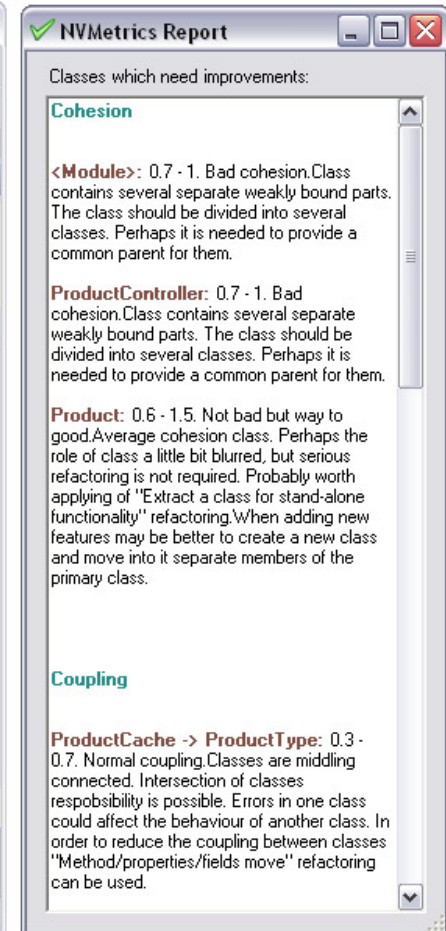
Assembly: **Examples**
Path: d:\Esperanza\Tools\Diplom_2009\My\NVMetrics\trunk\bin\Debug\Examples.dll

Cohesion Coupling Instability

| Class | Cohesion value |
|-------------------|----------------|
| > <Module> | 2 |
| ProductType | 0,375 |
| ProductCache | 0 |
| ProductController | 2 |
| Product | 0,9 |
| * | |

Recommendation:

// Assembly CsTransformationPar
Location: D:\Esperanza\Tools\Diplom_2009\parsers\cs_transformation_pa
Name: CsTransformationParserLibrary
Type: Library



✓ NVMetrics Report

Classes which need improvements:

Cohesion

<Module>: 0.7 - 1. Bad cohesion. Class contains several separate weakly bound parts. The class should be divided into several classes. Perhaps it is needed to provide a common parent for them.

ProductController: 0.7 - 1. Bad cohesion. Class contains several separate weakly bound parts. The class should be divided into several classes. Perhaps it is needed to provide a common parent for them.

Product: 0.6 - 1.5. Not bad but way to good. Average cohesion class. Perhaps the role of class a little bit blurred, but serious refactoring is not required. Probably worth applying of "Extract a class for stand-alone functionality" refactoring. When adding new features may be better to create a new class and move into it separate members of the primary class.

Coupling

ProductCache -> ProductType: 0.3 - 0.7. Normal coupling. Classes are middling connected. Intersection of classes responsibility is possible. Errors in one class could affect the behaviour of another class. In order to reduce the coupling between classes "Method/properties/fields move" refactoring can be used.

Заключение

В результате исследований в рамках разработки системы анализа и контроля качества исходного кода были выполнены следующие задачи:

- рассмотрены метрики качества исходного кода, а именно связность, сцепление и нестабильность;
- разработан алгоритм подсчета значений метрик;
- разработана экспертная система по выдаче текстовых рекомендаций на основе полученных значений метрик;
- выработаны рекомендации для метрик связности, сцепления и нестабильности;
- реализован прототип приложения для анализа исходного кода с использованием метрик.

