

# Анализ показателей метрик при миграции и реинжиниринге программного комплекса

З.И. Фабарисов

Кафедра Вычислительной математики и  
кибернетики  
Уфимский государственный авиационный  
технический университет  
Уфа, Россия  
e-mail: zagirf@gmail.com

Д.А. Ризванов

Кафедра Вычислительной математики и  
кибернетики  
Уфимский государственный авиационный  
технический университет  
Уфа, Россия  
e-mail: ridmi@mail.ru

## Аннотация<sup>1</sup>

В статье исследуется задача миграции программного комплекса для работы с торговым оборудованием с языка C# на C++. Сформулирована математическая модель анализа миграции ПО на основе формализованных требований к миграции и анализа кодовой базы. Анализ метрик для оценки эффективности миграции показал преимущества объектно-ориентированных метрик для решения поставленной в работе задачи.

## 1. Введение

Реинжиниринг программного обеспечения - это модификация программной системы, которая происходит после того, как она была запрограммирована с обратной стороны. Данный процесс наиболее широко используется, когда задействована унаследованная система. Программные системы развиваются высокими темпами, так как проводится много исследований, для их улучшения, поэтому очень часто получается, что устаревшим системам приходится работать на новой вычислительной платформе. Существуют различные формы реинжиниринга программного обеспечения, преобразование исходного кода - это простейшая форма реинжиниринга программного обеспечения, где исходный код одного языка программирования преобразуется в исходный код какого-либо другого языка программирования.

Миграция исходного кода программного комплекса или, как это часто называют, реинжиниринг (re-engineering) используется для поддержки эволюционного технического обслуживания и наиболее широко используется в современной

разработке программного обеспечения. Термин «техническое обслуживание» должен быть заменен реконструкцией, поскольку он не только восстанавливает информацию из существующей системы, но и использует эту информацию для изменения или воссоздания существующей системы, чтобы улучшить ее общее качество.

Миграция программного комплекса на другой язык программирования - долгий и дорогостоящий проект, состоящий из множества этапов.

Работа посвящена анализу миграции программного комплекса для работы с торговым оборудованием с языка C# на C++.

## 2. Постановка цели и задач

Цель работы – Анализ миграции программного комплекса для работы с торговым оборудованием с языка C# на C++.

Задачи:

1. Анализ текущей кодовой базы;
2. Формализация требований к миграции;
3. Разработка математической модели анализа миграции ПО;
4. Выбор метрик для оценки миграции;

## 3. Анализ предметной области

Миграция кода — актуальная задача в современной индустрии разработки ПО. Кратко этот процесс можно описать как переход от использования одной рабочей среды к другой, которая в большинстве случаев считается лучшей. Миграция может включать переход на новое оборудование, новое программное обеспечение или и то, и другое. Миграция может быть небольшой, например, перенос одной системы или крупномасштабной, с участием многих систем, новых приложений или переработанной сети.

Также возможен перенос данных из одного типа базы данных в другой. Для этого требуется, чтобы данные

---

Труды Шестой всероссийской научной конференции "Информационные технологии интеллектуальной поддержки принятия решений", 28-31 мая, Уфа-Ставрополь, Россия, 2018

содержались в едином формате, по которому можно произвести перенос из старой базы данных в новую. Поскольку новая база данных может быть организована по-другому, может возникнуть необходимость написать программу, которая может обрабатывать переносимые файлы.

Когда миграция программного обеспечения достигает функциональной эквивалентности, перенесенное приложение может быть более точно согласовано с текущими и будущими бизнес-потребностями посредством добавления новых функций к преобразованному приложению.

Перенос установленного программного обеспечения со старого ПК на новый может осуществляться с помощью средства миграции программного обеспечения. Термин миграция также может быть применен к описанию процесса перемещения данных с одного устройства хранения на другое.

#### 4. Подходы к миграции программного обеспечения

Миграция приложения должна учитывать множество разных вещей. Необходимо глубокое понимание объектно-ориентированных понятий.

Основные подходы к миграции [1]:

- создание программ-обёрток (эмуляция API);
- синтаксический подход;
- семантический подход.

#### 5. Анализ кодовой базы

Опишем преимущества использования языка C++ для ПО работы с торговым оборудованием:

1. Широкая поддержка операционных систем и архитектур. Основные операционные системы – Windows, MacOS и Linux. Написание на языке C++ дает возможность разворачивать приложение на практически любых современных операционных системах.
2. Производительность и оптимизация кода. Код написанный на языке программирования C++ исполняется непосредственно процессором, а основным плюсом языка является возможность прямой работы с памятью. Конечно, такая свобода может создать больше проблем, но в ряде случаев это лучше, чем иметь некий предел в производительности.
3. Самодостаточность приложения. Для выполнения программы на языке C# необходима установка дополнительных зависимостей (например, .Net Framework). Работая с языком C++ возможно написать автономное приложение, которое может запускаться также и в составе других приложений.
4. Большое наличие крупных библиотек. Отличие набора библиотек у C++ и C# в том, что

библиотеки на C++ имеют большую историю, за которую стали неплохо отлажены и оптимизированы. Эти библиотеки часто кроссплатформенны и имеют открытый код. Также есть возможность использовать огромное количество библиотек, написанных на языке C, однако они используют не объектный, а структурно-процедурный интерфейс.

#### 6. Этапы миграции программного кода

Рассмотрим основные этапы миграции и их характеристики:

Перенос кода – основной этап миграции. Основные проблемы переноса:

- Выбор типов данных – некоторые встроенные типы данных C# отсутствуют на языке C++;
- Использование шаблонов – различия между реализациями шаблонов в двух языках программирования довольно большие;
- Одновременная работа с несколькими экземплярами приложения – при переносе на C++ был сделан расчет на возможность запуска нескольких экземпляров приложения, которые должны взаимодействовать.

Тестирование – настолько же важный этап, как и первый.

- Написание тестовых сценариев для отладки работы – проверка удачной работы кода и получения ожидаемых кодов ошибок при соответствующих сценариях;
- Анализ обратной связи от отдела тестирования - после самостоятельной проверки продукта, необходима проверка программного обеспечения специалистами из отдела тестирования;
- Анализ обратной связи от тестовой группы - небольшой группе реальных пользователей продукта подменяется приложение на перенесенное;
- Дальнейшее улучшение кода и исправление найденных ошибок.

#### 7. Оценка миграции программного обеспечения

Для оценки миграции используются метрики программного обеспечения.

Метрика - это мера, позволяющая получить численное значение некоторого свойства программного обеспечения. Объектно-ориентированные метрики вводятся с целью: улучшить понимание качества продукта, оценить эффективность процесса конструирования, улучшить качество работы на этапе проектирования [2]. Определяемые наборы метрик должны

ориентироваться на особенности и уникальные характеристики программного продукта.

## 8. Математическая модель анализа миграции

Имеется программное обеспечение  $P_1 = \langle C_1, F_1, I_1 \rangle$  где  $C_1$  – исходный код приложения,  $F_1$  – функционал приложения,  $I_1$  – взаимодействие с программой.

Необходимо спроектировать приложение  $P_2 = \langle C_2, F_2, I_2 \rangle$  так, чтобы:

Множество функционала нового приложения включало множество функционала старого:  $F_1 \subseteq F_2$

Разность между эксплуатацией/производительностью старой и новой программой было минимальным:  $\min(I_1 - I_2)$

Оценка исходного кода нового приложения по показателям метрик, не должна быть ниже, чем у старого кода:  $M(C_2) \geq M(C_1)$ .  $M$  – анализ метрик программного обеспечения.

## 9. Выбор метрик для оценки миграции

С точки зрения метрик выделяются пять характеристик объектно-ориентированных систем: локализация, инкапсуляция, информационная закрытость, наследование и способы абстрагирования объектов. Каждая из этих характеристик напрямую влияет на показатели.

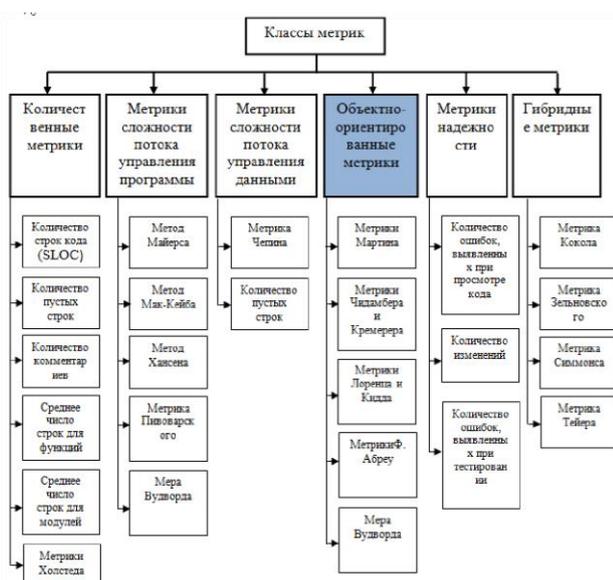


Рис. 1. Классы метрик

Общая классификация метрик, которая чаще всего встречается в научной литературе, показана на Рис. 1 [2]. Из рисунка видно, что классификация метрик состоит из семи основных классов. Каждый класс содержит наиболее часто используемые типы метрик. Наиболее простые в использовании – количественные показатели. Этот класс включает такие показатели, как: количество пустых строк; количество

комментариев; процент комментариев (отношение количества строк, содержащих комментарии к общему числу строк, выраженных в процентах); среднее число строк для функций (классов, файлов); среднее число строк, содержащих исходный код для функций (классов, файлов); среднее число строк для модулей.

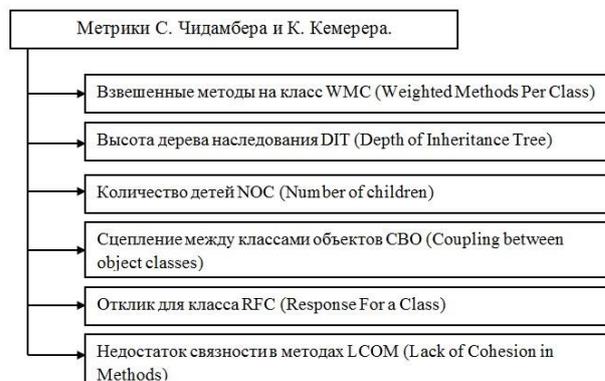


Рис.2. Шесть метрик С. Чидамбера и К. Кемерера

На современном этапе развития разработки программного обеспечения большинство проектов создаются на основе объектно-ориентированного подхода; в связи с этим существует много метрик, позволяющих оценить сложность объектно-ориентированных проектов [3].

В данном классе наиболее часто используемыми являются набор метрик Мартина, набор метрик Чидамбера и Кемерера (Рис.2.). Метрики для отдельного класса, иерархии классов и взаимодействия классов являются наиболее ценными для оценки качества работы [2].

## 10. Расчет производительности программного обеспечения

Оценка производительности приложения зависит от многих факторов. Для анализа миграции необходимо провести проверку свойств программы [4], таких как:

- время выполнения (пользователь не должен испытывать замедления времени работы программы);
- количество оперативной памяти, потребляемой программой (влияет на количество одновременно запущенных программ и на их нормальную работу);
- объём потребляемых вычислительных ресурсов (чрезмерное использование ресурсов может снизить общую скорость работы компьютера).

Для анализа данных характеристик необходимо проводить профилирование.

## 11. Заключение

Исследована задача миграции программного комплекса для работы с торговым оборудованием с

языка C# на C++. Произведен анализ текущей кодовой базы, формализованы требования к миграции, которые легли в основу математической модели анализа миграции ПО. Анализ метрик для оценки эффективности миграции показал преимущества объектно-ориентированных метрик для решения поставленной в работе задачи.

#### **Список используемых источников**

1. Алексюк А. О. Ицкун В. М. Семантически-ориентированная миграция java-программ: опыт практического применения // Моделирование и анализ информационных систем. - Ярославский государственный университет им. П.Г. Демидова (Ярославль). – Т.24 № 6(72). – С. 677-690.
2. Самойликова О. А. Использование объектно-ориентированных метрик для анализа программного обеспечения / О. А. Самойликова, Г. М. Рудакова // Образовательные ресурсы и технологии. 2014. № 1(4). -С. 78-83.
3. Евсеев В.В. Применение программных метрик кода на раннем этапе жизненного цикла программного обеспечения/В.В. Евсеев//Восточно-европейский журнал передовых технологий. -Харьков, 2011. -№ 1/2 (49). -С.19-21.
4. Вартанов С.П., Ермаков М.К., Герасимов А.Ю. Прикладное применение динамического анализа программ, исполняющихся в интерпретирующих средах // Труды Института Системного Программирования РАН. – Институт системного программирования РАН (Москва), 2017 – Т.29 №1. – С.135-148