На правах рукописи

Дробинцев Павел Дмитриевич

Интегрированная технология обеспечения качества программных продуктов с помощью верификации и тестирования

Специальность 05.13.11 –
Математическое и программное обеспечение
вычислительных машин, комплексов и компьютерных сетей

Автореферат
диссертации на соискание ученой степени кандидата технических наук

Санкт – Петербург - 2006

Работа выполнена в Государственном образовательном учреждении высшего профессионального образования «Санкт-Петербургский государственный политехнический университет».

Научный руководитель – кандидат технических наук, профессор
Котляров Всеволод Павлович

Официальные оппоненты – доктор технических наук, профессор
Лисс Александр Рудольфович
- кандидат технических наук, доцент
Кознов Дмитрий Владимирович

Ведущая организация - Санкт-Петербургский институт информатики и автоматизации РАН

Защита состоится «1» июня 2006 г. в 16 часов на заседании диссертационного совета Д 212.229.18 при ГОУ ВПО “Санкт-Петербургский государственный политехнический университет” по адресу: 195251, Санкт-Петербург, Политехническая ул., д.29, 9 уч. корп., ауд. 325.

С диссертацией можно ознакомиться в Фундаментальной библиотеке ГОУ ВПО “Санкт-Петербургский государственный политехнический университет”.

Автореферат разослан «28» апреля 2006 г.

Ученый секретарь
диссертационного совета Шашихин В.Н

1. ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

1.1 Актуальность работы. Одной из основных проблем, которую решают разработчики программного обеспечения (ПО), является проверка корректности его функционирования. Из практики известно, что проверка качества программного обеспечения начинается с разработки требований к проектируемой системе, продолжается до момента вывода системы из эксплуатации и занимает до 50% времени всего жизненного цикла ПО. Этим определяется повышение требований к полноте и производительности подобных проверок, что приводит к появлению на рынке новых технологий и инструментария автоматизации контроля правильности функционирования ПО.

Для создания качественного ПО прежде всего необходимо обеспечить ясность в постановке задачи и методах её решения. Поэтому с ростом сложности ПО возрастает потребность в средствах визуального моделирования и проектирования, а также в специалистах, знающих и умеющих использовать их на практике. Программистское сообщество осознаёт, что резервы повышения производительности труда при создании качественного ПО связаны с системным подходом и использованием передовых методов и средств автоматизации разработки больших программных проектов. В данном направлении ведётся значительное количество исследований по созданию новых стандартов и инструментальных средств, которые объединяются в рамках такого направления как Computer Aided Software Engineering (CASE).

Из работ Боэма известно, что стоимость исправления ошибок в пределах жизненного цикла продукта растёт экспоненциально. Именно поэтому в последнее время большое внимание специалистов привлечено к привнесению систематизации в область разработки ПО и, как следствие, к различным подходам разработки формальных требований на систему. Здесь наиболее популярными средствами описания являются стандартизованные языки UML, SDL, MSC. Формальные нотации позволяют применить к разработке требований и спецификаций формальные методы доказательства корректности различных свойств разрабатываемых систем, в том числе и корректности функционирования, что позволяет в процессе кодогенерации получить код, соответствующий спецификациям, в котором обнаружение значительной части ошибок смещено на ранние стадии разработки программного проекта.

Существующие в настоящее время на рынке технологии и инструменты для разработки ПО далеко не во всем обеспечивают решение поставленной проблемы. Это является следствием их узкой направленности на определённые этапы жизненного цикла. Существуют эффективные нотации для описания требований, такие как: UML, MSC, SDL,

VDM, Alloy, NP, B, RSL, LOTOS, ML, Isar, PVS, CASL, SMV, Promela, Esterel, SCR, Murphi и другие. Представленные нотации основаны на таких широко известных моделях программ как сети Петри (Петри), конечные автоматы (Мили, Мура и др.), темпоральные логики (Приора), алгебры параллельных процессов(Милнера, Хоара, Бергстры и Клоппа), транзиционные системы(Парка), агенты и среды(Летичевского и Гилберта). В тоже время существует множество инструментов верификации, разработанных в различных научных лабораториях по всему миру: LOTOS (организация ISO), Spin(лаборатория BellLabs), Kronos(лаборатория VERIMAG), SCR(лаборатория NavalResearch), VRS (организация ISS) и другие, и инструменты автоматизации тестирования созданные известными компаниями по разработке ПО, это Rational Rose компании IBM, Telelogic TAU компании Telelogic AB, TAT компании Motorola, Together компании Borland и другие.

Тем не менее, в настоящее время практически нет промышленных технологий, позволяющих совместно использовать инструментарий тестирования и верификации. Особенно актуальна данная проблема в области создания реактивных систем встроенного применения, где очень важны поведенческие свойства и где для полноценного обеспечения корректности поведения системы необходимо проверить множество сценариев за время не превышающее сроки, предусматриваемые на эту деятельность в программном проекте.

Настоящая работа посвящена созданию технологии, обеспечивающей совместное использование верификации и тестирования на основе требований, описанных в формальных нотациях и поддерживающей все этапы жизненного цикла ПО. Технология основывается на создании требований в виде множества базовых протоколов, каждый из которых описывает элементарное событие в рамках поведения системы. На основе разработанных протоколов технология позволяет проводить верификацию различных свойств, а также автоматическую генерацию кода тестов и системы. Она также учитывает возможность применения сопутствующих технологий, использующих формальные нотации UML, SDL и MSC.

1.2 Цели и задачи диссертационной работы.

Целью является разработка методов и средств технологии автоматизированной верификации и тестирования реактивных программных систем. Технология обеспечивает совместное использование тестирования и верификации в процессе производства ПО встроенных реактивных систем с целью повышения качества конечного продукта на основе раннего обнаружения дефектов и автоматизации тестирования. В рамках достижения цели были решены следующие задачи:

- Определение множества инструментов верификации и тестирования, которые могут быть использованы в рамках единой интегрированной технологии.
- Создание методологической базы по совместному использованию возможностей верификации и тестирования на основе выбранного инструментария.
- Разработка программного обеспечения, позволяющего автоматизировать переход от фазы верификации к фазе тестирования.
- Проверка работоспособности предложенной технологии и инструментальных средств в нескольких проектах по производству ПО.

1.3 Предметом исследования являются методы и инструментарий обеспечения качества программного обеспечения.

1.5 Методы исследования. В диссертации используется теория базовых протоколов, аппарат математической логики, концепция объектно-ориентированных моделей, а также теория конечных автоматов. Основными критериями являлись технологичность и простота использования разрабатываемой технологии, а также возможность применения технологии, как для проектов малой сложности, так и для проектов большой сложности. Применялись стандарты Unified Modeling Language (UML), Specification and Description Language (SDL), Message Sequence Charts (MSC), идеология построения CASE-средств.

1.5 Обоснованность и достоверность полученных результатов обеспечена использованием строгого математического аппарата теории базовых протоколов при разработке единой технологии и сравнением результатов по достижению качества ПО, полученных различными методами.

1.6 Научные результаты и их новизна:

1. На базе теоретической модели базовых протоколов предложена методика формализации требований, позволяющая разрабатывать спецификации на программное обеспечение в виде элементарных поведенческих диаграмм, что значительно облегчает создание сложных систем и даёт возможность применения формальных методов для анализа спецификаций.
2. Разработана методика проверки заданных свойств системы на основе модели на языке SDL, позволяющая осуществлять поиск ошибок в SDL спецификациях.
3. Предложена методика генерации тестового набора, обеспечивающего полное покрытие при заданных ограничениях функциональности системы на основе верификации.

4. Предложена методика проверки недетерминированности поведения системы, позволяющая определять возможные места возникновения недетерминизмов в спецификациях.

5. Предложена методика проверки аннотированных сценариев поведения системы, позволяющая проверять свойства полноты системы, описанной множеством базовых протоколов.

6. Предложена методика автоматизации тестирования, позволяющая использовать результаты верификации на стадии тестирования.

1.7 Практическая значимость работы. На базе полученных научных результатов был разработан комплекс программных средств, предназначенных для совместного использования верификации и тестирования. Программный комплекс был использован в компании Motorola в 6 программных проектах по разработке встроенных систем в таких областях как: разработка мобильных устройств, телекоммуникационных и телематических систем. Кроме того созданная технология и программные средства были внедрены в проекте разработки функциональной программной подсистемы управления узлами связи “83Т54-2М” в ОАО “Интелтех” и проекте “Исследование работы контроллера в системе распределённого электропитания железнодорожного вагона” в ЗАО “Северо-Западная лаборатория Лтд”. Созданные методики и программные средства могут быть использованы для обеспечения качества реактивных систем в широком спектре встроенных применений.

1.8 Апробация работы. Основные результаты и выводы диссертации докладывались на следующих конференциях: Международная научная конференции «IEEE Russia Northwest Section, 110 Anniversary of Radio Invention conference» (СПб., 2005 г.); Международная научная конференция «The Annual Motorola Science Advisory Board Associates (SABA) » (San Francisco, California 2005 г); Motorola Technology Day (Spb 2005); Конференции Технологии Microsoft в теории и практике программирования 2002, 2004 и 2005 гг.; Конференциях XXIX-2001 г, XXX-2002 г, XXXI - 2003 г, XXXII - 2004 г недели науки СПбГПУ. По материалам диссертации опубликовано 9 печатных работ.

1.9 Внедрение. Методика интегрированной верификации и тестирования внедрена в ЗАО “Северо-Западная лаборатория”, НПО “Интелтех”, ЗАО “Моторола” и использовалась при разработке учебно-методического комплекса СПбГПУ по курсу «Введение в технологии верификации» на кафедре ИУС. Практическое использование представляемых на защиту результатов подтверждается соответствующими актами о внедрении.

1.10 Структура и объём работы. Работа содержит 4 главы, 5 приложений, введение и заключение. Объём работы 150 страниц, количество иллюстраций 80, список использованной литературы состоит из 165 наименований.

2. СОДЕРЖАНИЕ РАБОТЫ

В **первой главе** диссертации на основе анализа источников рассмотрены основные методы обеспечения качества ПО. Приведена классификация методов тестирования по различным критериям и описаны основные методы автоматизации тестирования, а также проведён сравнительный анализ промышленных инструментальных средств автоматизации тестирования. Рассмотрен формальный подход к проблеме обеспечения качества ПО и проанализирован математический аппарат моделей программ. При выборе инструментальной основы технологии были описаны и классифицированы различные методы верификации и проведён сравнительный анализ инструментов верификации, применяемых в промышленных проектах по созданию ПО.

Анализ показал, что множество формальных моделей применяется для спецификации (описания) поведенческих свойств приложений, их верификации и тестирования. Наиболее популярными нотациями являются сети Петри, конечные автоматы, регулярные выражения, алгебры параллельных процессов, транзиционные системы и темпоральные логики.

Формальные модели описания являются основой моделей верификации, реализованных в различных подходах: на алгебре процессов CSP основываются подходы LOTOS, разработанный организацией ISO (International Organization for Standardization) и Spin, разработанный в Bell Labs в 80-х годах, на конечно автоматной модели описания основываются подходы Kronos, разработанный в лаборатории VERIMAG для верификации систем реального времени, SCR, разработанный в лаборатории Naval Reseasrch и SMV, разработанный в Carnegie Mellon University, на различных логиках основаны такие подходы как HOL, разработанный в университете Кембриджа и PVS, разработанный группой исследователей из лабораторий SRI International, National Science Foundation и Naval Research.

В свою очередь инструментарий промышленного тестирования в подавляющем большинстве случаев основывается на конечно-автоматных моделях, представленных в нотациях UML и SDL в рамках описания алгоритмов поведения систем, и на моделях описания взаимодействующих процессов, таких как MSC в рамках описания тестов.

На основе анализа были сделаны следующие выводы о текущем положении дел в области исследования:

- Нотации моделей программ хорошо применимы к решению небольших узко направленных практических задач, в то время как применение существующих нотаций к описанию требований промышленного программного продукта с большим количеством требований не представляется возможным, в силу сложности внедрения предлагаемых методов формального описания в процесс производства ПО.
- Большинство средств верификации не используют широко распространённых графических стандартов.
- Подавляющее большинство инструментов верификации не стыкуются с инструментами автоматизации тестирования в силу использования несогласованных вердиктов и исходных описаний в системах верификации и тестирования.
- Совместное использование верификации и автоматизации тестирования должно принести ощутимые выгоды в процесс производства ПО.

Полученные результаты позволили утверждать, что в настоящее время отсутствуют интегрированные программные средства и технологий верификации и тестирования, пригодные для использования в процессе производства ПО.

Во **второй главе** на основе формальных языков инженерного проектирования UML, SDL, MSC, а также нотации базовых протоколов (BP), в рамках которой базовый протокол определяется тройкой Хоара, представляющей собой выражение вида $"x(a \text{ fi} < u > b)$, где x – список (типизированных) параметров, a и b – формулы базового языка, u – процесс протокола, были предложены методики формализации требований на основе базовых протоколов, доказательства свойств системы на основе SDL спецификации, генерации тестового набора, доказательства детерминированности поведения системы, проверки аннотированных сценариев поведения системы и автоматизации тестирования.

Методика формализации требований на основе базовых протоколов(MBP) позволяет разработчику ПО описывать требования на программное обеспечение в виде элементарных MSC диаграмм с добавленными логическими формулами, выражающими пред и пост условия. При разработке методики была создана стандартная форма, по результатам заполнения которой может быть осуществлено построение формальных спецификаций. Разработанная стандартная форма содержит информацию по проведению следующие этапов формализации:

1. Определение процессов – агентов. В рамках данного этапа выделяются задачи: определения состава множества агентов с присвоением имён, определения свойств агентов, определения начальных значений свойств агентов.
2. Определение множества сигналов, которое полностью описывает поведение системы.

3. Определение множества фильтров (выражающихся в виде формул, использующих равенства, неравенства и логику второго порядка), которые могут быть использованы для идентификации состояний или событий в процессе генерации трасс на основе множества BP.
4. Определение пред и пост условий и процессных компонент для каждого базового протокола.

Формально методика MBP осуществляет преобразование: $MBP:(A,S,St)\ \text{fi}\ (BP)$, где: $A=\{a_1,a_2...a_n\}$ – множество агентов описываемых в системе, $S=\{s_1,s_2...s_{n1}\}$ – множество сигналов, $St(st_1,st_2...st_m)$ – множество состояний описывающих пред и пост условия, $BP(bp_1,bp_2...bp_k)$ – множество базовых протоколов.

Разработанная на данном этапе модель является базовой для проведения этапов верификации и тестирования на основе следующих методик.

Методика доказательства заданных свойств системы (SAC) - на основе её модели на языке SDL с последующей генерацией кода позволяет :

- Провести верификацию SDL спецификаций (P) на основе заданных на формальном языке свойств (S).
- Проверить полученные данные путём исполнения контр-примеров (TR) на сгенерированном по SDL спецификации (P) коде системы.

Методика SAC осуществляет преобразование: $SAC:(S,F,P)\ \text{fi}\ (V,TR)$, где: $S=\{s_1,s_2...s_n\}$ – множество свойств системы, $F=\{f_1,f_2...f_k\}$ – множество фильтров, $P=\{p_1,p_2...p_m\}$ – множество процессов, описанных в синтаксисе SDL, $V=\{v_1,v_2...v_j\}$ – множество вердиктов, $TR=\{tr_1,tr_2...tr_l\}$ – множество трасс в синтаксисе MSC.

Методика генерации тестового набора (TG) - обеспечивает полное относительно зафиксированного критерия покрытие свойств системы на основе метода базовых протоколов и позволяет проводить генерацию тестового набора с 100% покрытием поведенческих требований на систему.

Методика TG осуществляет преобразование: $TG:(ENV,EVENT,FILTER,BP)\ \text{fi}\ (TR,V)$, где: ENV – описание окружения генерации, $EVENT=\{e_1,e_2...e_n\}$ – описание возможных событий, $FILTER=\{f_1,f_2...f_m\}$ – описание фильтров, обеспечивающих формулировку анализируемых свойств, $BP=\{bp_1,bp_2...bp_k\}$ – множество базовых протоколов, $TR=\{tr_1,tr_2...tr_l\}$ – множество трасс в синтаксисе MSC, $V=\{v_1,v_2...v_j\}$ – множество вердиктов.

Методика доказательства детерминированности поведения системы (ТСС) - на основе модели ВР позволяет проводить автоматический поиск недетерминизмов на основе формального описания спецификации на систему, представленного в виде множества сценариев поведения и базовых протоколов (SC, BP).

Методика ТСС осуществляет преобразование: $TCC : (SC, BP)$ fi (V), где: $SC = \{sc_1, sc_2 ... sc_n\}$ – множество сценариев в формате MSC, $BP = \{bp_1, bp_2 ... bp_k\}$ – множество базовых протоколов, $V = \{v_1, v_2 ... v_j\}$ – множество вердиктов.

Методика проверки аннотированных сценариев поведения системы (АС) - позволяет автоматически осуществлять проверку того факта, что полученные в результате декомпозиции части спецификаций в виде базовых протоколов (BP) полностью покрывают сценарии поведения системы (SC), а также проверять сценарии поведения системы, дополненные логическими условиями (аннотациями).

Методика АС осуществляет преобразование: $AC : (SC, BP, ISt)$ fi (V), где: $SC = \{sc_1, sc_2 ... sc_n\}$ –множество сценариев в формате MSC, $BP = \{bp_1, bp_2 ... bp_k\}$ – множество базовых протоколов, $ISt = \{ist_1, ist_2 ... ist_n\}$ -множество начальных состояний, $V = (v_1, v_2 ... v_m)$ – множество вердиктов.

Методика автоматизации тестирования (ТАТ) - на основе символьных трасс полученных в результате верификации, позволяет проводить генерацию тестов в целевой код на основе сценариев поведения системы (SC), представленных в виде контр-примеров или трасс, полученных на стадии верификации.

Методика ТАТ может быть представлена в виде: $TAT : (SC, CONF, TOOLCONF, PIPECONF)$ fi (V), где: $SC = \{sc_1, sc_2 ... sc_n\}$ –множество сценариев в формате MSC, CONF – тестовая конфигурация, TOOLCONG – конфигурация модулей ТАТ, PIPECONF – конфигурация цепочек запуска, $V = (v_1, v_2 ... v_m)$ – множество вердиктов.

Представленные методики используют согласованные входные данные и позволяют получить согласованные результаты, что делает эффективным их совместное применение в рамках процесса производства ПО. Общая схема взаимодействия и применения описанных методик представлена на рис. 1.

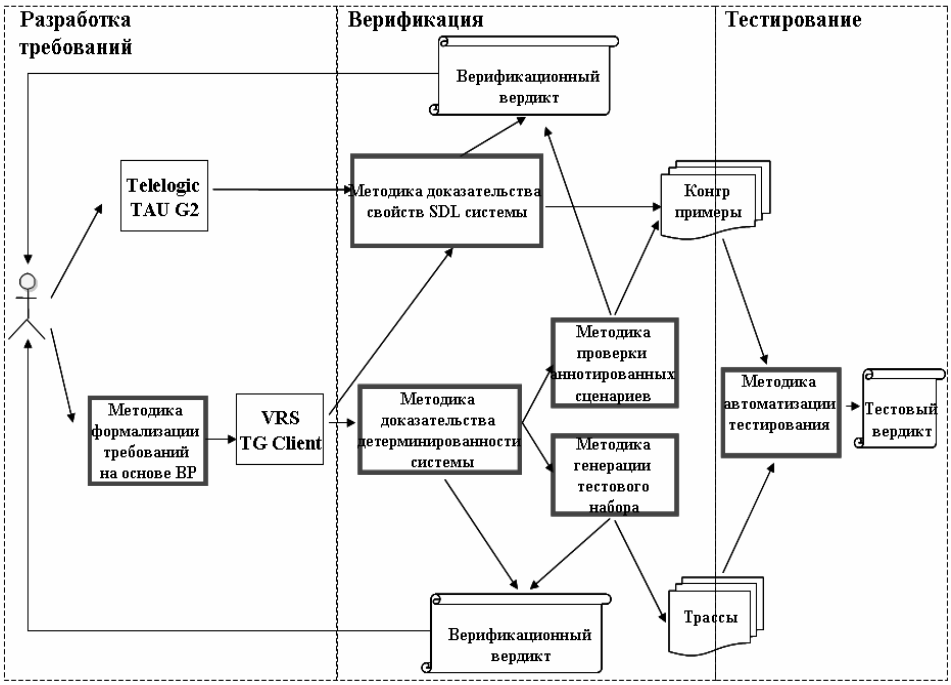

Рис. 1 Технология автоматизированной верификации и тестирования

Методика формализации требований на основе базовых протоколов используется на этапе разработки требований. Четыре методики доказательства и проверки свойств системы используются на стадии верификации разработанных требований. При этом входом для них служат результаты работы методики формализации требований. В свою очередь методика автоматизации тестирования основывается на результатах этапа верификации, что позволяет упростить процесс тестирования и уменьшить необходимое количество тестов, гарантирующее качество разрабатываемой системы. Методики позволяют обнаружить большую долю ошибок, традиционно выявляемых на этапе тестирования, на более раннем этапе верификации, что приводит к снижению издержек процесса производства ПО.

Таким образом, вторая глава содержит методическую базу разрабатываемой технологии. Конструктивность и эффективность предложенных методик подтверждена при их реализации и пилотировании на реальных программных проектах.

В **третьей главе** предложен программный комплекс поддержки разработанных в ходе диссертационной работы методик, основанный на использовании существующих инструментов автоматизации тестирования и верификаций.

Общая схема взаимодействия инструментов в рамках технологии изображена на рис.2.

Рис. 2 Общая схема взаимодействия инструментов в рамках интегрированной технологии

На основе анализа, проведённого в первой главе, было принято решение использовать в интегрированной технологии 3 инструмента (рис. 2):

- Tau G2 – для создания формальных требований в языках UML, MSC, SDL.
- VRS – для верификации созданных спецификаций.
- TAT – для проведения тестирования.

Следует отметить, что использование инструмента Telelogic TAU G2 на этапе разработки формальных требований позволяет автоматически переходить от языка UML к языкам SDL и MSC в зависимости от типа решаемых в рамках проекта задач.

В рамках реализованной технологии обмен информации осуществляется с использованием нотаций SDL, MSC и XML. При этом в формате SDL передаются поведенческие спецификации на систему подлежащую верификации и тестированию, в формате MSC - тестовые сценарии, контр-примеры и тестовый вердикт, а в формате XML - файл конфигурации для подсистем тестирования и верификации.

Для создания интегрированной технологии были разработаны программные средства, решившие следующие задачи:

- Интеграцию представления спецификаций в UML, SDL и MSC.
- Автоматизацию перехода от стадии верификации к стадии тестирования.
- Унификацию входных потоков для VRS и TAT посредством использования единого синтаксического анализатора MSC диаграмм.

Представленные программные реализации вошли в релиз версии продукта VRS/TAT и позволили обеспечить создание интегрированной технологии, базирующейся на методиках, описанных в рамках второй главы. Все модули оттестированы и использованы при пилотировании технологии в реальных программных проектах.

Четвёртая глава посвящена использованию разработанных методик и программного комплекса для обеспечения качества в промышленных проектах по разработке ПО.

Предложенные в главах 2 и 3 методики и средства были проверены на работоспособность в реальных программных проектах из следующих областей применения: телекоммуникации, разработка мобильных телефонов и телематика – встроенные применения в автомобилестроении.

В состав методов и средств технологии, используемых в пилотировании, входили:

1. Методика формализации требований на основе базовых протоколов.
2. Методика автоматизации тестирования.
3. Методики проверки, доказательства и разработки формальных требований.

При этом для каждого проекта применялись методики, соответствующие постановке задачи. Например, для проектов, находящихся на стадии проектирования, применялись методики верификации, а для проектов, находящихся на стадии тестирования, совместно применялись методики верификации и автоматического тестирования.

Обобщённая структура экспериментального комплекса для проведения автоматизированной верификации и тестирования приведена на рис. 3.

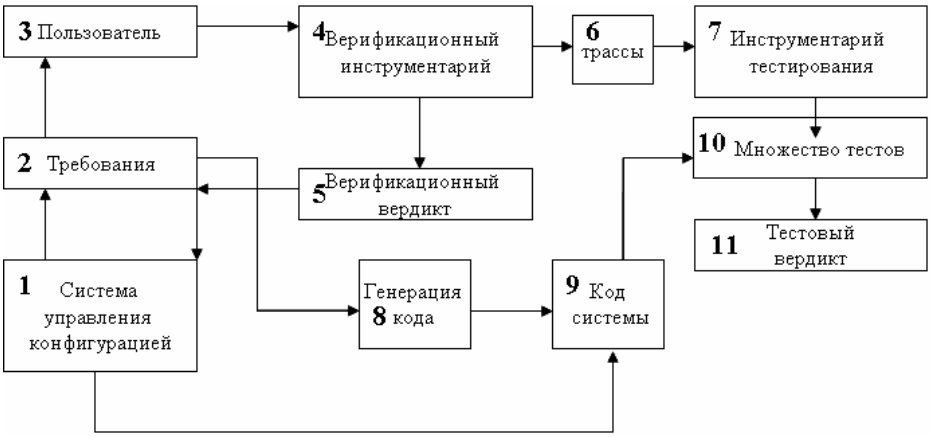

Рис. 3 Структура экспериментального комплекса

Из рис. 3 видно, что требования (2), поступившие из системы управления конфигурацией (1), формализуются пользователем (3) и попадают на инструментарий верификации (4). По результатам верификации происходит корректировка требований (5). В то же время требования служат основой для автоматической генерации (8) или ручной разработки кода. Результат верификации в виде трасс (6), обеспечивающих полное покрытие свойств системы или в виде контр-примеров, поступает на вход инструмента

тестирования (7), который порождает множество тестов (10). И на последнем этапе код системы тестируется на основе порождённого множества тестов (11).

Совместное применение тестирования и верификации в рамках интегрированной технологии позволило сократить в среднем на 10% затраты на разработку проектов. Таблица 1 содержит данные, полученные в ходе реализации следующих проектов:

Проект№ 1 – описание протокола GPNet, являющегося протоколом стандарта GSM.

Проект№ 2 – описание протокола сети Iden, являющегося стандартом в сетях CDMA.

Проект№ 3 – ПО для сотовых телефонов (обеспечение функций внешнего экрана).

Проект№ 4 – Встроенная в автомобиль система радаров, предназначенная для обеспечения безопасности и удобства маневрирования.

Проект№ 5 – Проект по интеграции мобильных устройств в автомобиле, обеспечивающий совместное использование различных устройств на основе беспроводной передачи данных.

Проект№ 6 – Проект по интеграции систем управления автомобилем, обеспечивающий централизованное управление различными встроенными модулями.

Table 1 . Характеристики применения технологии в промышленных проектах

Проект	Характеристики				
	Количество поведенческих требований (шт.)	Количество ВР (шт.)	Обнаружено дефектов (шт.)	Сгенерировано тестов (шт.)	Оценка сокращения трудоёмкости (%)
№1	67	56	8	73	6
№2	200	137	86	362	8
№3	75	50	15	550	8
№4	26	11	3	110	11
№5	520	219	38	16	5
№6	2222	533	117	25	17

Полученные данные показывают, что с ростом количества требований, растёт количество базовых протоколов, представляющих формальную модель, данная зависимость характерна для любого проекта из любой области, так как каждый протокол является формальным выражением требования на систему. При этом количество протоколов растёт медленнее, чем количество требований. Это связано с тем, что в промышленных проектах с большим количеством модулей в составе системы, несколько требований формализуются в один протокол. Таким образом, можно утверждать, что сложность формального описания, создаваемого в рамках применения технологии, растёт медленнее, чем количество требований. На Рис. 4 представлен график зависимости

количества обнаруженных с использованием технологии дефектов в 6 проектах по производству ПО.

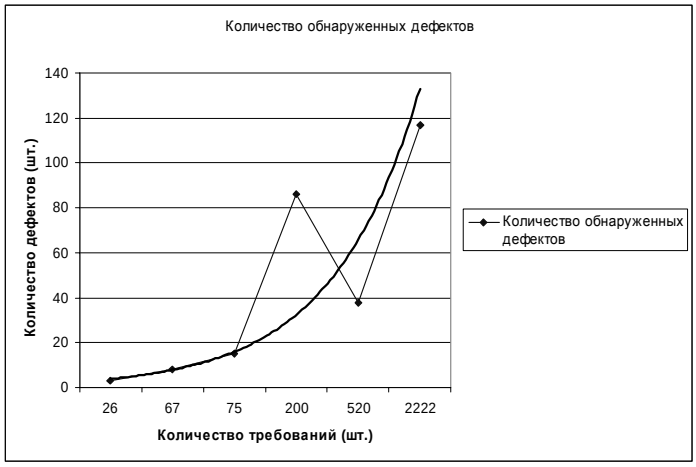

Рис. 4 Рост количества обнаруженных дефектов при увеличении количества требований

Из рис. 4 видно, что график имеет выброс. Это связано с большим количеством дефектов, обнаруженных в проекте разработки протокола сети Iden. Основной особенностью данного проекта является то, что технология была использована на всех этапах разработки ПО, что позволило увеличить количество найденных дефектов. Это подтверждает тот факт, что технология даёт больший выигрыш при применении на всех стадиях разработки.

Анализ результатов полученных в ходе проведения анализа пилотирования технологии, позволяет сформулировать следующие выводы:

- Количество протоколов растёт медленнее, чем количество требований, что приводит к упрощению описания спецификаций для больших проектов.
- Технология может использоваться как на всех, так и на отдельных этапах создания ПО.
- Технология даёт больший эффект при использовании на протяжении всего процесса создания продукта.

Для оценки применения технологии в проектах с большим объёмом требований, в рамках работы был проведен сбор данных для проектов различной сложности. Сложность оценивалась по количеству исходных требований к проекту. По этому показателю все проекты разбиты на 5 групп: небольшие проекты (Н) – проекты, количество требований к которым составляет 10-ки, средние проекты (С) с объёмом требований 10^2, большие проекты (Б) с объёмом требований 10^3, большие промышленные проекты (БП) с объёмом требований превышающим $5*10^3$, сверх большие промышленные проекты (СБП) с объёмом требований составляющим 10^4.

Для двух последних групп проектов пилотирование не проводилось. Тем не менее, основываясь на экспериментальных данных по применению технологии в 6 реальных проектах по разработке ПО и, полагая неизменность полученных в рамках анализа результатов зависимостей, можно провести оценку использования технологии для больших и сверх больших промышленных проектов

На рис. 5 приведены обобщающие данные по различным зависимостям для описанных выше пяти типов проектов.

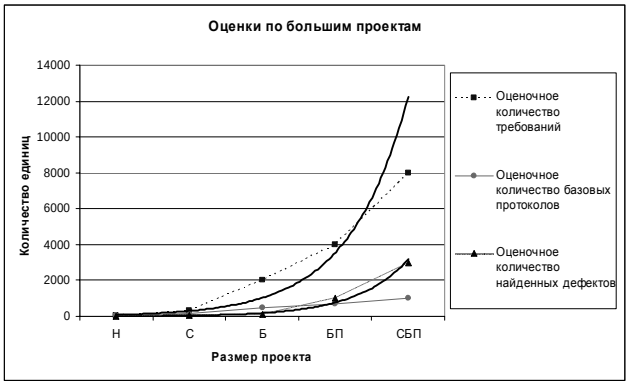

Рис. 5 Усреднённые данные по количественным характеристикам проектов различных типов

На основе представленных данных можно сделать вывод о том, что рост количества требований и количества, найденных с использованием технологии дефектов, имеет экспоненциальную зависимость. В тоже время рост количества базовых протоколов характеризуется линейной зависимостью и, следовательно, эффективность применения технологии растёт с ростом объёма проекта. На рис. 6 показана зависимость относительного сокращения трудоёмкости от количества требований.

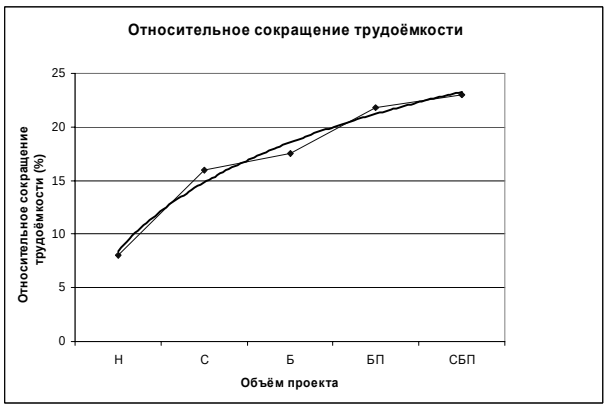

Рис. 6 Относительное сокращение трудоёмкости

Использование аппроксимации методом наименьших квадратов дало следующую зависимость: $K_{эф} = 9{,}23 \ln x + 8{,}41; R = 0{,}98$ где х-сложность проекта. При этом для средних проектов с использованием технологии сокращение времени разработки

составляет порядка 5%, а для сверх больших проектов этот показатель составляет уже 23%.

3. ОСНОВНЫЕ РЕЗУЛЬТАТЫ И ВЫВОДЫ

В диссертационной работе поставлена и решена проблема обеспечения качества программного обеспечения реактивных встроенных систем, измеряемого плотностью остаточных ошибок в промышленном программном продукте, путем использования интегрированной технологии верификации и тестирования, в результате применения которой обеспечивается сокращение издержек разработки до уровня приемлемого в условиях промышленного производства ПО.

Основными результатами диссертационной работы являются:

1. Методики позволяющие обеспечить необходимый уровень качества разрабатываемого ПО за счёт совместного использования верификации и тестирования. В диссертационной работе предложены методики:
   a. используемые на стадии разработки требований - методика формализации требований на основе базовых протоколов (MBP);
   b. используемые на стадии верификации - методика доказательства свойств SDL системы (SAC), методика генерации тестового набора (TG), методика доказательства детерминированности поведения системы (TCC) и методика проверки аннотированных сценариев поведения (AC);
   c. используемые на стадии тестирования - методика автоматизации тестирования (TAT).
2. Программный комплекс поддержки методик интегрированной технологии верификации и тестирования, созданный на базе теоретической модели базовых протоколов.
3. Технические отчёты, содержащие анализ ошибок, обнаруженных в пилотируемых проектах с помощью интегрированной системы автоматизации тестирования и верификации и разработанной в диссертационной работе интегрированной технологии. Обнаруженные ошибки в документации и спецификациях были исправлены разработчиками и внесены в обновлённые версии программных систем.
4. Оценка эффективности предложенных методик и ПО на базе использования в программных проектах различной сложности в ЗАО “Motorola ЗАО”, ЗАО “Северо-Западная лаборатория”, НПО “Интелтех”.

Результаты, полученные в процессе выполнения проектов с использованием технологии, позволили сделать выводы о работоспособности и границах эффективности применения разработанных методов и средств интегрированной технологии обеспечения качества программных продуктов с помощью верификации и тестирования.

4. СПИСОК ПУБЛИКАЦИЙ ПО ТЕМЕ ДИССЕРТАЦИИ

1. Sergey. N. Baranov, Vsevolod P. Kotlyarov, Alexander A. Letichevsky, Pavel D. Drobintsev. The technology of automation verification and testing in industrial projects. IEEE Russia Northwest Section, 110 Anniversary of Radio Invention conference. P. 81-84.
2. S. Baranov, V. Kotlyarov, A. Letichevsky, P. Drobintsev. Implementation of an integrated verification and testing technology in telecommunication project. IEEE Russia Northwest Section, 110 Anniversary of Radio Invention conference. P. 87-91.
3. Баранов С.Н., Дробинцев П.Д., Летичевский А.А., Котляров В.П. Верификационная технология базовых протоколов для разработки и проектирования программного обеспечения. “Программные продукты и системы” 1(69) 2005. С. 25-28.
4. Даишев М.Ш., Дробинцев П.Д., Котляров В.П., Песков Д.В., Юсупов Ю.В. Применение интегрированной технологии тестирования и верификации к модели телефонной сети. “Программные продукты и системы” 1(69) 2005. С. 35-38.
5. Дробинцев П.Д., Котляров В.П. Разработка комплекса автоматизации тестирования для операционных систем реального времени.// Конкурс-конференция студенческих работ в области современных технологий программирования компании Microsoft: Материалы межвузовского конкурса-конференции./ СПб.: СПбГПУ, 2002, стр 32-33
6. Дробинцев П.Д., Дворянский Л.В., Котляров В.П. Платформенно-независимый метод тестирования.// Технологии Microsoft в теории и практике программирования: Материалы межвузовского конкурса-конференции студентов, аспирантов и молодых ученых Северо-Запада./ СПб.: Изд-во СПбГПУ, 2004, стр 23-24
7. Дробинцев П.Д., Даишев М.Ш., Песков Д.В., Юсупов Ю.В., Котляров В.П. Пилотирование интегрированной технологии тестирования и верификации// Технологии Microsoft в теории и практике программирования: Материалы межвузовского конкурса-конференции студентов, аспирантов и молодых ученых Северо-Запада./ СПб.: Изд-во Политехн. Ун-та, 2005, стр 26-27
8. Дробинцев П.Д., Котляров В.П. Технологии автоматизированной верификации функциональных спецификаций программного обеспечения.// Технологии Microsoft в теории и практике программирования: Материалы межвузовского конкурса-конференции студентов, аспирантов и молодых ученых Северо-Запада./ СПб. Изд-во СПбГПУ, 2004, стр 24-25
9. Дробинцев П.Д., Вдовец Е.М., Котляров В.П. Автоматизация тестирования верификатора функциональных спецификаций // XXX Юбилейная неделя науки СПбГТУ. Материалы межвузовской научной конференции / СПбГТУ. 2002.С. 17-18.