

ТЕХНОЛОГИЯ ТЕСТИРОВАНИЯ КОМПЛЕКСА ПРОГРАММ «ЛОГОС»

М. Г. Лащманова, Н. С. Аверина, Т. Ю. Баканова, А. С. Косарева, Е. А. Савиных, Т. Н. Серова

ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской обл.

В рамках работ по созданию и внедрению на предприятиях промышленности и ВУЗах отечественного базового программного обеспечения для комплексного имитационного моделирования на супер-ЭВМ в Институте Теоретической и Математической Физики (ИТМФ) ФГУП «РФЯЦ-ВНИИЭФ» ведется разработка комплекса программ ЛОГОС. ЛОГОС включает в себя несколько компонентов: счетные коды инженерного анализа и, пре- и постпроцессоры для обеспечения данных счетных кодов интерактивным интерфейсом для подготовки расчетной модели, управления счетом и анализа результатов.

Разработка комплекса ЛОГОС ведется в соответствии с общепризнанными для коммерческих приложений стандартами разработки и обеспечения качества программного обеспечения [1, 2]. Все выпускаемые версии данного программного продукта проходят комплексное тестирование. Основная задача тестирования – выявить отклонения и сбои в работе тестируемого приложения, что позволит разработчикам вовремя исправить проблему и повысить качество продукта.

По мере возрастания сложности комплекса ЛОГОС увеличиваются трудности и затраты при его тестировании и подготовке версий к выпуску. Ограниченность ресурсов тестирования привела к необходимости тщательного подбора применяемых методов тестирования и тестовых данных с целью получения наибольшего покрытия и глубины проверок. Помимо этого, для обеспечения проведения своевременного и полного тестирования комплекса

ЛОГОС и всех его компонентов необходимо придерживаться определенного плана-графика работ.

В целом все описанные далее шаги технологии тестирования могут быть применены в разработке любого схожего программного обеспечения.

1. Порядок подготовки версии комплекса программ ЛОГОС к выпуску

На рис. 1 приведен порядок подготовки версии комплекса программ ЛОГОС к выпуску. Подготовка к выпуску состоит из следующих этапов:

1. За 4 недели до официальной даты передачи версии комплекса программ производится фиксация поставляемой версии ЛОГОС, т. е. в данную версию разрешается вносить только исправления выявленных ошибок.

2. После фиксации версии ответственные исполнители передают пробные версии счетных кодов (так называемых решателей) и пре- и постпроцессоров ЛОГОС-ПреПост и Scientific View исполнителю, ответственному за подготовку общего дистрибутива комплекса программ ЛОГОС, вместе со списками новых возможностей и изменений в своих компонентах. Версии компонентов должны быть собраны на всех утвержденных программных и аппаратных платформах согласно техническому заданию на разрабатываемое программное обеспечение.

3. Ответственный исполнитель подготавливает дистрибутив комплекса ЛОГОС, который содержит дистрибутивы как основных компонентов, так и вспомогательного программного обеспечения необходимого для работы всего комплекса.



Рис. 1. Процесс подготовки версии комплекса ЛОГОС к выпуску

4. Ответственный за подготовку дистрибутива исполнитель выкладывает подготовленный дистрибутив комплекса программ ЛОГОС на общедоступный ресурс и рассылает письма ответственным исполнителям и руководству проекта со ссылками на ресурс и информацией о готовности дистрибутива.

5. Затем происходит передача подготовленного дистрибутива ЛОГОС на тестирование. Проводятся различные виды тестирования для всех компонентов комплекса ЛОГОС, а также проверка единой документации комплекса ЛОГОС и тестирование пробного дистрибутива на предмет корректности установки всех его компонентов.

6. Найденные в ходе тестирования проблемы, подробно описываются в системе отслеживания ошибок Atlassian JIRA [3]. Контроль за распределением описаний ошибок и проблем (так называемых запросов) между разработчиками осуществляется руководителями направлений разработки компонентов ЛОГОС.

7. Разработчики исправляют, выявленные в их программном коде, ошибки.

8. За 3 дня до даты выпуска версии формируется документ со списком имеющихся в комплексе программ ЛОГОС ошибок, так называемый «History». Данный документ входит в дистрибутив комплекса программ ЛОГОС.

9. Тестирование может считаться завершенным при достижении согласованных критериев готовности программного продукта (описаны далее).

10. По результатам тестирования формируется отчет со статистикой по выявленным и исправленным ошибкам, «особо проблемным» модулям/компонентам тестируемого ПО. Отчет предоставляется руководителю проекта, руководителям разработки всех компонентов. Решение о возможности выпуска версии ПО принимается руководством проекта.

11. После того, как версия признана стабильной, производится ее окончательная сборка на всех утвержденных программных и аппаратных платформах.

12. Обновляется общий дистрибутив и передается на тестирование.

13. Производится тестирование собранного дистрибутива на предмет корректности установки всех его компонентов и еще раз проводится автоматизированное регрессионное тестирование пользовательского интерфейса.

14. Созданный дистрибутив комплекса программ ЛОГОС, а также все собранные счетные модули для всех возможных платформ выкладываются на FTP-сервер, доступный предприятиям, в день утвержденной датой передачи версии или ранее.

2. Особенности тестирования комплекса программ ЛОГОС

При подготовке версии ПО к поставке на предприятия все виды тестирования, кроме инсталляционного и автоматизированного, проводятся на установленном дистрибутиве комплекса ЛОГОС. При

тестировании дополнительное внимание уделяется новым возможностям и доработанным функциональностям, которые описываются разработчиками ЛОГОС в списках об изменениях в компонентах.

Также при сценарном и функциональном тестировании используется документация комплекса программ ЛОГОС, где подробно описаны функциональные возможности входящих в его состав программ и примеры выполнения типовых задач и получаемые результаты, с которыми в процессе тестирования проводится сравнение как с эталонами.

В случае подготовки версии ЛОГОС с урезанными возможностями, тестирование отключения указанных возможностей и работоспособности остальных проводится согласно указаниям от разработчиков компонентов и на предоставленных ими данных.

Для проведения всех видов тестирования используется единая база согласованных исходных и эталонных тестовых данных, а также входящие в состав дистрибутива примеры.

Выявленные случаи конфликта настроек систем или установленных драйверов оборудования в обязательном порядке должны быть либо исправлены, либо описаны в пользовательской документации.

Ошибки, обнаруженные в ходе тестирования, и предложения по совершенствованию возможностей тестируемого ПО, оформляются в системе управления ошибками для исправления разработчиками.

Для ЛОГОС была выбрана используемая в РФЯЦ-ВНИИЭФ система управления ошибками JIRA [3]. После исправления ошибки или доработки соответствующая возможность снова проверяется.

3. Виды тестирования комплекса программ ЛОГОС

Для тестирования комплекса программ ЛОГОС и отдельных его компонентов в рамках подготовки версии к выпуску применяются следующие виды тестирования:

- инсталляционное тестирование;
- тестирование графического пользовательского интерфейса или функциональное тестирование (ручное и автоматизированное);
- сценарное тестирование (ручное и автоматизированное);
- регрессионное тестирование;
- интеграционное тестирование;
- верификационное тестирование;
- модульное тестирование;
- тестирование документации.

Основные виды тестирования (инсталляционное, функциональное, сценарное и интеграционное) комплекса программ ЛОГОС проводятся на всех утвержденных программных и аппаратных платформах.

Для проведения всех видов тестирования используется единая база согласованных исходных и эталонных тестовых данных. По результатам каждого запуска модульного и функционального тестирования

ведется журнал (log-файл) прохождения тестов, а также проводится анализ выявленных ошибок, в том числе для определения «проблемных» модулей.

В инсталляционном тестировании проводится проверка работы инсталляционной системы для утвержденных программных и аппаратных платформ – ОС Windows, ОС Linux (список конкретных ОС и платформ формируется согласно техническому заданию на разрабатываемое программное обеспечение). Проверяется возможность самостоятельной установки программы пользователем и достаточность указаний по установке, настройке и запуску ПО. Также проверяется наличие и правильность установки всех компонентов программного комплекса ЛОГОС, успешность их запуска на выполнение.

Тестирование графического интерфейса пользователя или функциональное тестирование проводится на основе ранее определенных требований к возможностям разрабатываемого программного продукта и сценариям использования. Цель функциональных тестов состоит в том, чтобы проверить соответствие разработанных компонентов установленным требованиям. Для данного вида тестирования часть тестов комплекса ЛОГОС была автоматизирована. Автоматизированное тестирование применяется для проверки базовых возможностей в том случае, когда можно проверить выходную информацию. В остальных случаях используется ручное тестирование.

При проведении ручного и автоматизированного функционального тестирования выполняется как «позитивное» тестирование (использование верных значений и «правильных» последовательностей действий пользователя), так и «негативное» (использование ошибочных значений и данных, нештатной последовательности действий пользователя), для проверки устойчивости программного продукта к воздействиям различного рода и проверки обработки исключительных ситуаций.

Сценарное тестирование проводится согласно разработанным сценариям использования программы. В сценариях описываются как работа каждого диалога, так и последовательности действий пользователя для выполнения какой-либо задачи в тестируемой программе. При этом описываются допустимые и недопустимые входные и выходные данные. Сценарное тестирование автоматизируется по мере возможности. Сценарным тестированием комплекса ЛОГОС можно также считать проверку работы решателей посредством выполнения тестовых задач, описанных в руководствах пользователя.

Регрессионное тестирование – повторное выполнение тестов, выявивших ошибки или проблемы во время предыдущих раундах тестирования. Помимо проверки исправления ошибок при регрессионном тестировании проверяется, не внесло ли сделанное исправление ошибок в другие части программы. Для регрессионного тестирования графического пользовательского интерфейса (далее ГПИ) комплекса ЛОГОС используются разработанные проекты автоматизированного тестирования, сценарное тестирование, про-

екты верификационного тестирования решателей и ручная проверка часто выявляемых ошибок.

Интеграционное тестирование – тестирование взаимодействия и правильной работы всех подключаемых модулей, библиотек, решателей. Для комплекса программ ЛОГОС проводится с использованием тестовых примеров из пользовательской документации решателей комплекса программ ЛОГОС в рамках функционального и сценарного тестирования.

Верификационное тестирование позволяет гарантировать, что программная система реализована без непредусмотренного функционала и соответствует предъявляемым требованиям. Результатом является вывод о соответствии (или несоответствии) продукции. На текущий момент верификационное тестирование для комплекса ЛОГОС проводится командами разработчиков конкретных компонентов самостоятельно. В частности, для решателей ЛОГОС такой вид тестирования проводится посредством задания значений параметров и проведения расчетов типовых задач и сравнения полученных результатов с эталонными аналитическими решениями. Начата автоматизация этого вида тестирования.

Модульное тестирование – процесс в программировании, позволяющий проверить на корректность отдельные модули исходного кода программы.

Тестирование документации – проверяется документация комплекса программ ЛОГОС на полноту, точность, ясность и правильность оформления.

4. Автоматизированное тестирование комплекса программ ЛОГОС

Автоматизированное тестирование - это процесс верификации ПО, при котором основные функции и шаги ГПИ-теста, такие как запуск, инициализация, выполнение, анализ и выдача результата, выполняются автоматически при помощи инструментов для автоматизированного тестирования.

Для автоматизации тестирования ГПИ комплекса ЛОГОС применяется система тестирования Qt-приложений Froglogic Squish [4]. Для управления автоматизированным тестированием, как модульным, так и функциональным, используется специализированная система Atlassian Bamboo [5].

Преимущества такого тестирования заключаются в том, что оно проходит автоматически, без участия пользователя по настроенному заранее сценарию (или плану в терминах Bamboo) и результаты тестирования, включая лог-информацию, отображаются в системе Bamboo.

Для автоматического тестирования ГПИ комплекса ЛОГОС разработан ряд скриптов, проверяющих правильность функционирования базовых возможностей комплекса ЛОГОС по работе с геометрией и сеточными данными, а также работу генераторов сеток. Начата автоматизация выполнения тестовых задач, описанных в руководствах пользователей решателей ЛОГОС-Аэрогидромеханика и ЛОГОС-Прочность. Разработанные скрипты объединены

в наборы для запуска и хранятся в специализированном репозитории под управлением системы контроля версий Mercurial Hg [6]. Согласно этим наборам для комплекса ЛОГОС создано и используется следующие планы автоматизированного тестирования:

- «часовое» тестирование – используется набор ГПИ-скриптов, выполнение которых длится не больше часа;
- полное тестирование – используется набор ГПИ-скриптов, выполнение которых занимает достаточно длительное время (порядка 12 часов);
- выполнение набора ГПИ-скриптов по тестированию генераторов сеток;
- выполнение примеров из руководств пользователя решателей.

Планы выполняются с автоматическим обновлением ГПИ-скриптов до актуального состояния из ветки репозитория, соответствующей тестируемой версии ЛОГОС, и для самой «свежей» на момент запуска плана тестирования сборки дистрибутива ЛОГОС.

В системе Vamboо страница каждого из завершивших выполнение планов автоматизированного тестирования содержит расширенную информацию о процессе выполнения на вкладке «Branch summary» (рис. 2):

- дата и время запуска автоматизированного тестирования, продолжительность выполнения;
- последние изменения, внесенные в общий репозиторий ГПИ-скриптов;

- число тестов, впервые возвративших ошибку выполнения, и тестов, которые ранее уже возвращали ошибку выполнения.

Результатом выполнения плана автоматизированного тестирования в Vamboо является список ГПИ-скриптов, в которых были выявлены ошибки. Также выводится информация о новых ошибках выполнения тестирования и уже проявившихся в более раннем тестировании (рис. 3).

Также выводится информация о конкретных ошибках в каждом из неуспешно выполнившихся тестов, которая позволяет выявить причины их возникновения (рис. 4). Выводится также информация об ошибках работы тестируемой программы (ключевое слово в логге: Failure), и об ошибках выполнения скриптов (ключевое слово: Error), доступен список удачно завершенных тестов.

После завершения работы каждого плана автоматизированного тестирования, ответственный исполнитель анализирует полученную информацию об ошибках и описывает каждую выявленную проблему в системе отслеживания ошибок JIRA.

Аналогичным образом средствами Vamboо автоматизирован запуск модульного тестирования для отдельных компонентов ЛОГОС-ПреПост. Тесты модульного тестирования запускаются автоматически при каждой сборке проекта ЛОГОС-ПреПост. Разработка тестов проводится программистами, преимущественно авторами тестируемого модуля.

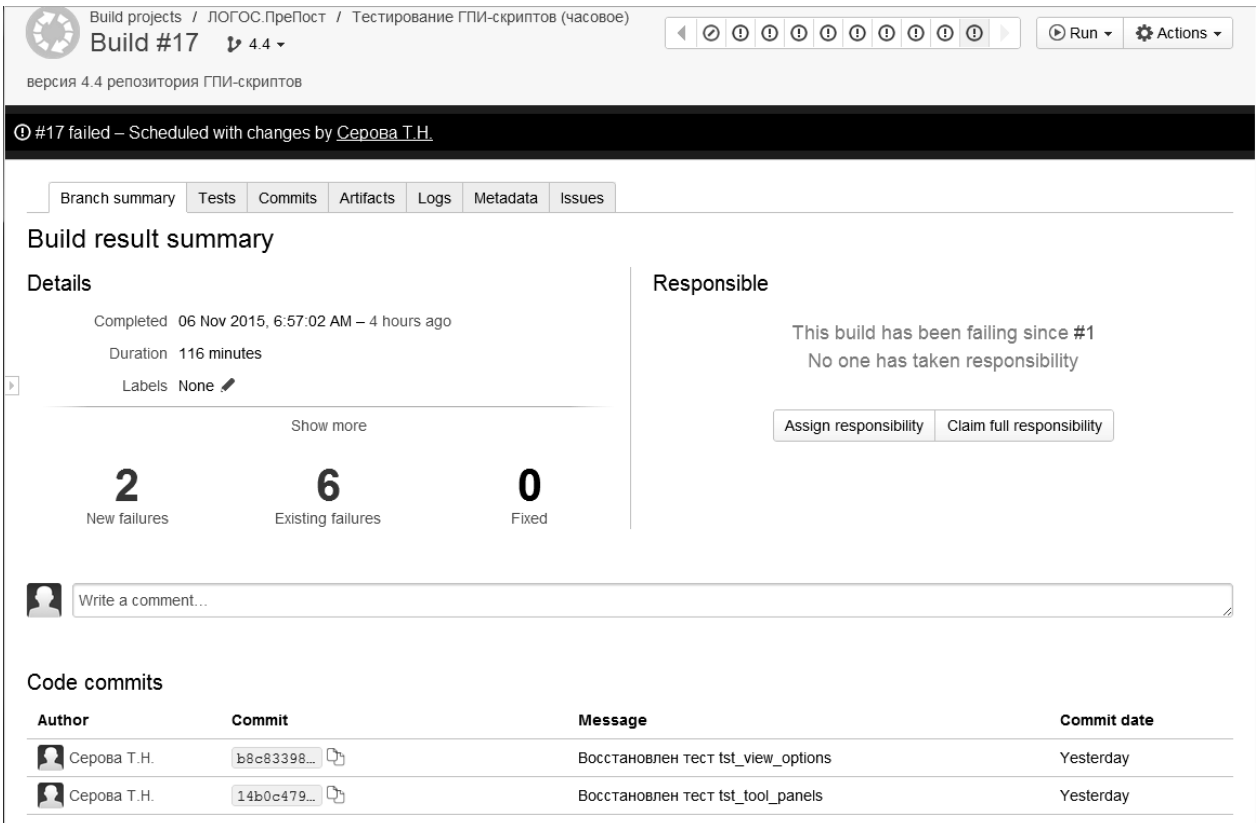


Рис. 2. Общие сведения о процессе выполнения плана автоматизированного тестирования в системе Vamboо

Code commits

Author	Commit	Message	Commit date
Попова Маргарита Юрьевна	61faf53c...	tst_tutorial_pump Устранено задание дублирующего поля температуры	2 days ago
Попова Маргарита Юрьевна	385d1477...	Перевод теста на новый вариант 1	2 days ago
Попова Маргарита Юрьевна	7a78e78f...	tst_tutorial_pump Изменено задаваемое значение	2 days ago

Configuration changes

Plan configuration has changed since the last successful build. See the plan audit log for more details.

Tests

Test	View job	Duration	
New test failures 2			
tst_cut_on_edge_and_hide_top_blocks .E:/Bamboo/bld/UNISYS-GUI1-JOB1/suite_1_hour_tests/tst_cut_on_edge_and_hide_top_blocks/test.py:319	windowsx64	< 1 sec	
ERROR: Все упало! Прибиваем процесс в памяти Error in clickButton() invocation: Widget '{text='Восстановить центр времени' type='QPushButton' unnamed='1' visible='0' window=(name='mainwindow' type='appMainWindow' visible='1			
tst_cut_on_edge_and_hide_top_blocks .E:/Bamboo/bld/UNISYS-GUI1-JOB1/suite_1_hour_tests/tst_cut_on_edge_and_hide_top_blocks/test.py:320	windowsx64	< 1 sec	
ERROR: Script Error NameError: global name 'failLog' is not defined			
Existing test failures 1			
Test	Failing since	View job	Duration
tst_check_picking_on_geometry .E:/Bamboo/bld/UNISYS-GUI1-JOB1/suite_1_hour_tests/tst_check_picking_on_geometry/test.py:353	#240 (Scheduled with changes by Лашманова И.Г. and Сорова Т.Н)	windowsx64	< 1 sec
FAILURE: Информация о 1й версии выведена неправильно			

Рис. 3. Общие сведения о процессе выполнения плана автоматизированного тестирования. Данные об ошибках в тестовых скриптах

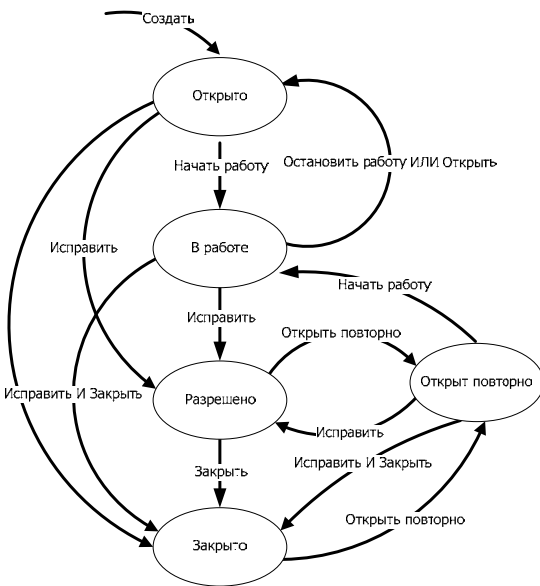


Рис. 4. Процесс работы с запросом в системе JIRA

5. Тестирование документации

Единая документация комплекса программ ЛОГОС представляет собой платформу-независимую интерактивную документацию и содержит описания всех компонентов и в общей сложности содержит около 10 тысяч страниц печатного текста.

Создание, дополнение и исправление статей документации производится в специально созданном проекте с использованием специализированной системы Adobe RoboHelp [7]. Там же для подключения вызова статей документации в пользовательский интерфейс программного продукта генерируется специализированный map-file, содержащий ссылки на статьи документации. Данный map-файл добавляется в исходные коды комплекса ЛОГОС. Затем средствами RoboHelp выполняется генерация готовой документации в нужном формате – html, doc или pdf.

При подготовке новой версии комплекса программ ЛОГОС к выпуску проводится проверка сгенерированной документации на полноту, точность, актуальность и правильность оформления статей, а также на правильность их подключения в графический пользовательский интерфейс комплекса.

Все выявленные в ходе тестирования документации неточности и несоответствия регистрируются в системе управления ошибками JIRA. После того, как все статьи отредактированы и недостающие добавлены в проект, проводится повторная проверка документации.

6. Учет и контроль ошибок

Как уже упоминалось, для учета ошибок и замечаний используется система отслеживания и учета ошибок Atlassian JIRA [3].

В системе используются различные типы запросов – ошибка, усовершенствование (существующей функции или задачи), новая функция, задача на выполнение.

Процесс работы с запросом в системе JIRA приведен на рис. 3.

Все найденные ошибки и замечания должны быть зарегистрированы с определенным приоритетом:

- наивысший (Blocker) – Блокирующая ошибка, приводящая приложение в нерабочее состояние, в результате которого дальнейшая работа с тестируемой системой или ее ключевыми функциями становится невозможна;
- высокий (Critical) – Крах операционной системы, аварийное завершение работы программы, общее нарушение защиты, зависание объекта испытания;
- средний (Major) – Неработоспособность основных функций программы, невозможность выполнения действий, предусмотренных сценариями использования системы;
- низкий (Minor) – Некорректная работа приложения при определенных действиях пользователя;
- незначительный (Trivial) – Приложение имеет незначительные ошибки или запрос на новую функциональность в будущих версиях.

7. Критерии завершения тестирования

Предлагается принять следующие критерии готовности продукта и, соответственно, завершения тестирования в рамках подготовки версии к выпуску:

- 80 % всех видов тестов должны проходить успешно.
- в системе учета ошибок, из всех зарегистрированных группой тестирования ошибок для этой версии, должно оставаться не более следующего процента открытых (нерешенных) ошибок/проблем, см. таблицу:

Допустимое количество нерешенных проблем

Приоритет ошибки	% открытых
Блокирующий (Blocker)	0
Критический (Critical)	0
Серьезный (Major)	10
Средний (Minor) + Низкий (Trivial)	25

Группа тестирования считает невозможной передачу версии ПО сторонним пользователям/заказчикам при невыполнении указанных критериев.

В случае невыполнения данных критериев решение о возможности выпуска версии комплекса программ ЛОГОС принимает руководство проекта.

Заключение

Быстрое увеличение сложности и расширение функциональных возможностей разрабатываемого комплекса ЛОГОС усложняет его тестирование. Необходимость повышения качества выпускаемого программного обеспечения увеличивает требования к покрытию и глубине тестирования. Регулярное инсталляционное и сценарное тестирование, постоянное обновление и расширение базы тестовых данных и скриптов автоматизированного тестирования, другие используемые для оценки качества комплекса ЛОГОС виды тестирования позволяют своевременно выявлять ошибки, что несомненно влияет на повышение качества продукта.

В настоящее время выполняются работы по автоматизации тестирования решателей, что позволит значительно увеличить качество выдаваемого ими результата и повысить в целом качество комплекса ЛОГОС, а также повлияет на увеличение скорости тестирования при подготовке к выпуску версии.

Литература

1. ГОСТ Р ИСО/МЭК 12207-2010 Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств.
 2. ГОСТ Р ИСО/МЭК 12119-2000 Информационная технология. Требования к качеству и тестированию.
 3. Official site of Jira program [Electronic resource]. Mode of access: www.atlassian.com/software/jira, 2015.
 4. Official site of Froglogic Squish program [Electronic resource]. Mode of access: www.froglogic.com/squish, 2015.
 5. Official site of Bamboo program [Electronic resource]. Mode of access: www.atlassian.com/software/bamboo, 2015.
 6. Official site of Mercurial program [Electronic resource]. Mode of access: www.mercurial.selenic.com, 2015.
- Official site of RoboHelp program [Electronic resource]. Mode of access: www.adobe.com/ru/products/robohelp/, 2015.