УДК 519.6
DOI: 10.53403/9785951505309_2022_27_1_110

Методика тестирования пакета программ ЛОГОС

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

Рассматривается проблема обеспечения качества сложных программных продуктов на примере разработанного в РФЯЦ-ВНИИЭФ пакета программ ЛОГОС. Описывается методика тестирования, разработанная и применяемая для обеспечения высокого качества пакета программ ЛОГОС. Описаны виды тестирования, их применение для различных компонентов ЛОГОС и периодичность проведения тестирования. Рассматриваются используемые для тестирования данные. Приведено описание применения инструментов автоматизации тестирования.

Одним из направлений работы РФЯЦ-ВНИИЭФ является реализация разрабатываемых физико-математических моделей в рамках собственного программного обеспечения (ПО). Приобретающее все больший масштаб создание сложного ПО большими коллективами специалистов РФЯЦ-ВНИИЭФ и необходимость его последующего отчуждения от разработчиков требует не только четкой организации работ по созданию ПО, но и обеспечения его высокого качества. В связи с этим разработка многофункционального пакета программ (ПП) инженерного анализа и суперкомпьютерного моделирования ЛОГОС [1, 2] и выведение его на уровень импортозамещающего типового программного продукта для высокотехнологичных отраслей промышленности Российской Федерации делают задачу обеспечения высокого качества разрабатываемого ПП одной из приоритетных.

Существуют государственные стандарты, регламентирующие процессы жизненного цикла ПО [3, 4], а также направленные непосредственно на обеспечение его качества [5, 6]. Однако эти стандарты, как и лучшие практики [7, 8] в данной области, не определяют жестко процесс тестирования продукта и рекомендуют адаптировать приведенные в них мероприятия к конкретной ситуации. Поэтому рассмотренные далее подходы к организации процесса обеспечения качества ПП ЛОГОС, выработанные на основании требований и рекомендаций государственных стандартов, учитывают особенности как самого ПП, так и процесса его разработки.

Основным инструментом обеспечения и контроля качества ПО является комплексное тестирование, встроенное в процессы жизненного цикла разрабатываемого программного продукта. Тестирование – это анализ ПО, направленный на оценку ПО и выявление отличий между его реально существующими и требуемыми свойствами.

Целью тестирования является минимизация количества ошибок в программном продукте, чтобы он работал в соответствии с предъявляемыми к нему требованиями с малым временем простоя. Тестирование ПО, согласно [5], должно быть направлено на предоставление информации о программном продукте и нахождение максимально возможного числа дефектов на возможно ранних этапах при заданных ограничениях стоимости и графика разработки.

Как уже упоминалось, стандарты в области качества ПО рекомендуют адаптировать приведенные в них мероприятия, как и состав, к конкретной ситуации. Поэтому первым шагом при выработке подходов к организации процесса обеспечения качества ПП ЛОГОС стало проведение анализа особенностей процесса разработки и непосредственно самого ПП.

Особенности пакета программ ЛОГОС и процесса его разработки

К особенностям самого пакета программ можно отнести:

– включение в его состав разнородных компонентов: консольных приложений счетных модулей, приложений препостпроцессинга с графическим пользовательским интерфейсом (ГПИ) ЛОГОС-Препост [9] и ScientificView [10], модуля документации, инсталляционного модуля;

– кросс-платформенность реализации;

– возможность проведения расчетов задач в локальном и удаленном режимах;

– высокий уровень распараллеливания.

К особенностям процесса разработки ПП ЛОГОС относятся:

– разработка разных компонентов отдельными командами большой численности, находящимися в разном подчинении;

– несколько ролей участников процесса разработки: разработчики, тестировщики, сотрудники команд внедрения на предприятия промышленности;

– разная скорость внесения изменений в исходные коды, напрямую влияющая на частоту подготовки исполняемых файлов компонентов ЛОГОС для проведения тестирования;

– отличающиеся для каждого компонента системные требования: версии операционных систем (ОС), набор необходимых библиотек и т. д.;

– ключевые ежегодные события: плановые выпуски версий (релизов) ПП ЛОГОС в среднем 2 раза в год.

Анализ перечисленных особенностей выявил, что при интеграции процесса тестирования в жизненный цикл ПП ЛОГОС необходимо соблюдать определенные правила разработки, а также применять виды тестирования, подходящие как для консольных компонентов ПО, так и для компонентов с ГПИ, в том числе для проверки их взаимодействия. Дополнительно необходимо обеспечить проведение тестирования на платформах, определенных в требованиях к ПП в качестве стандартных, как в локальном, так и удаленном режимах. Кроме того, отобранные по данным критериям виды тестирования должны быть встроены в процесс разработки ПП ЛОГОС и его компонентов таким образом, чтобы обеспечивать своевременные выпуски версий ЛОГОС высокого качества, а также способствовать максимально быстрому выявлению дефектов ПО. Выработанная таким образом методика тестирования ЛОГОС подробнее описана далее.

Виды и уровни тестирования пакета программ ЛОГОС

С учетом всех перечисленных выше особенностей ПП ЛОГОС для применения в методике были отобраны следующие виды тестирования.

1. Функциональное – тестирование ПО в целях проверки выполнения функциональных требований, т. е. способности ПО в определенных условиях решать задачи, заданные в требованиях к программному продукту. При проведении 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 различного рода воздействиям и обработки исключительных ситуаций.

2. Модульное, или юнит-тестирование – процесс, позволяющий проверить на правильность отдельные компоненты или модули исходного кода программы, наборы из одного или более

программных модулей вместе с соответствующими управляющими данными, процедурами использования и обработки. Данный вид тестирования выполняется разработчиками ПП ЛОГОС.

3. Регрессионное – тестирование ПО с целью обнаружения ошибок в уже проверенных участках программ (или исходных кодах), а также проверки того, что изменения в исходном коде не внесли ошибок в реализацию уже существующих возможностей. Для регрессионного тестирования разработаны и используются наборы тестов (проекты) автоматизированного тестирования ГПИ ПП ЛОГОС, наборы тестов верификационного тестирования счетных модулей, сценарное тестирование при решении типовых и производственно-методических задач (указанные виды тестирования описаны далее) и ручная проверка часто выявляемых ошибок.

4. Сценарное – тестирование, при проведении которого средствами ГПИ ПП ЛОГОС проверяется правильность работы на стандартных сценариях использования пакета, например: построение конечно-элементной модели, задание параметров задачи для каждого счетного модуля и т. д. Для дальнейшей проверки при тестировании в сценариях описываются как работа каждого диалога, так и последовательности действий пользователя для выполнения какой-либо задачи в тестируемой программе. При этом задаются допустимые и недопустимые входные и выходные данные. К примеру, сценарным тестированием ПП ЛОГОС является проверка работы счетных модулей посредством выполнения тестовых задач, описанных в руководстве пользователя ПП ЛОГОС.

5. Верификационное тестирование – проверка, удовлетворяют ли результаты текущего этапа разработки ПО и его компонентов сформулированным требованиям. Данный вид тестирования позволяет гарантировать, что программная система реализована в соответствии с предъявляемыми требованиями. Результатом является вывод о соответствии (или несоответствии) продукции. Для счетных модулей ЛОГОС такой вид тестирования проводится путем задания значений параметров типовых задач, проведения расчетов и сравнения полученных результатов с эталонными решениями; в большинстве случаев выполняется автоматизированно.

6. Интеграционное тестирование – проверка корректности взаимодействия компонентов ПО при совместной работе. Для ПП ЛОГОС данный вид тестирования проводится с использованием тестовых примеров из пользовательской документации, а также при решении производственно-методических задач в рамках функционального и сценарного тестирования.

7. Инсталляционное тестирование – проверка правильности установки всех компонентов ПО для согласованных в техническом задании (ТЗ) или требованиях к ПП ЛОГОС конфигураций ОС Windows и Linux. Проверяются возможность самостоятельной установки программы пользователем и достаточность указаний по установке, настройке и запуску ПО. Также проверяется наличие и правильность установки всех компонентов ПП ЛОГОС, успешность их запуска на выполнение.

8. Тестирование документации – проверка пользовательской документации на актуальность и полноту.

С точки зрения интеграции различных видов тестирования в процесс разработки ПО и с учетом ключевых событий в виде периодического выпуска версий (релизов) ЛОГОС тестирование можно разделить на два больших направления:

1) межрелизное – текущее тестирование между релизами, которое требует меньшей интенсивности проводимых мероприятий и при этом способствует раннему выявлению дефектов;

2) предрелизное – тестирование в рамках подготовки релизов ЛОГОС к выпуску.

Дополнительно процесс тестирования был условно разделен на три уровня в соответствии с масштабностью проводимых мероприятий.

К первому уровню (рис. 1) относится внутреннее тестирование компонентов ЛОГОС, проводимое на стадии разработки непосредственными разработчиками для каждого компонента.

Рис. 1. Уровень 1: внутреннее тестирование компонентов ЛОГОС

Тестирование первого уровня имеет следующие особенности:

– может проводиться как при каждом построении исполняемого файла компонента, так и с удобной для разработчиков периодичностью;

– выполняется на отдельных наборах тестовых данных разработчиков, которые могут быть гораздо объемнее общедоступной базы тестовых данных;

– выполняется для согласованных в ТЗ или требованиях к ПП ЛОГОС версий ОС Windows и Linux.

Ко второму уровню (рис. 2) отнесено ежемесячное тестирование, проводимое силами команды тестировщиков и имеющее следующие особенности:

– тестируются все компоненты ЛОГОС, кроме инсталляционного модуля и пользовательской документации;

– выполняется регрессионное тестирование счетных модулей и ЛОГОС-Препост, особенно для периодически проявляющихся ошибок;

– выполняется верификационное тестирование счетных модулей, функциональное тестирование новых и измененных возможностей компонентов препостпроцессинга, сценарное тестирование всего пакета программ;

– выполняется на отобранных типовых задачах тестового базиса (см. далее), в том числе на задачах из обучающих материалов руководства пользователя;

– выполняется для согласованных стандартных конфигураций ОС Windows и Linux.

Рис. 2. Уровень 2: межрелизное тестирование

К третьему уровню (рис. 3) отнесено тестирование ПП ЛОГОС, выполняемое при подготовке версии к плановому релизу с периодичностью раз в полгода. Тестирование на этом уровне отличается большим масштабом по сравнению с первыми двумя уровнями. Оно выполняется всеми участниками процессов разработки и тестирования – разработчиками, тестировщиками, сотрудниками команд внедрения.

На третьем уровне:

– тестируются все компоненты ЛОГОС. По сравнению с уровнем 2 дополнительно проверяются инсталляционный модуль и модуль документации;

– выполняется регрессионное тестирование всех компонентов, кроме инсталляционного модуля и документации;

– выполняется верификационное тестирование счетных модулей, функциональное тестирование новых и измененных возможностей компонентов препостпроцессинга, сценарное тестирование всего пакета программ. Дополнительное внимание уделяется тестированию генераторов сеточных моделей и заданию параметров расчетов;

– все виды тестирования выполняются на типовых задачах тестового базиса и модельных производственно-методических задачах верификационного базиса, в том числе характерных для отраслей промышленности;

– тестирование выполняется для согласованных стандартных конфигураций версий ОС Windows и Linux.

Рис. 3. Уровень 3: предрелизное тестирование

В случаях подготовки дистрибутивов ПП ЛОГОС для нестандартных системных конфигураций по запросам отдельных организаций-заказчиков проводится сокращенный вариант тестирования третьего уровня.

С учетом особенностей разработки ПП ЛОГОС были предложены следующие критерии готовности пакета программ и, соответственно, завершения тестирования в рамках подготовки версии к релизу:

1) 80 % всех видов тестов должны проходить успешно;

2) в системе отслеживания дефектов с указанным приоритетом открытых (нерешенных) для конкретной версии программного продукта из всех зарегистрированных ошибок должно оставаться не более допустимого количества (табл. 1).

В настоящее время первый критерий стабильно выполняется: количество успешно пройденных тестов больше 95 %.

В случае невыполнения данных критериев (см. табл. 1) версия считается не готовой к выпуску.

Таблица 1

Допустимое количество нерешенных ошибок

Приоритет ошибки	Открытые ошибки, %
Наивысший (Blocker)	0
Высокий (Critical)	0
Средний (Major)	10
Низкий (Minor)	25

Для нормативного обеспечения тестирования были разработаны организационные документы, регламентирующие процессы разработки и тестирования ПП ЛОГОС, такие как план тестирования и регламент разработки и тестирования, регламент работы в системе отслеживания дефектов. По окончании предрелизного тестирования версий ПП ЛОГОС выпускается отчет о результатах тестирования каждого компонента с заключением о готовности или неготовности к выпуску дистрибутива.

Тестовый базис пакета программ ЛОГОС

Для проведения всех видов тестирования используется единая база согласованных с разработчиками ПП ЛОГОС исходных и эталонных данных, называемая тестовым базисом. Тестовый базис также условно разделяется на три уровня (рис. 4), аналогичных описанным выше уровням тестирования, и соответственно используется на этих уровнях.

Тестовый базис размещен в хранилище системы контроля версий (репозитории) под управлением Mercurial Hg [11] и включает в себя следующие данные:

– геометрические и сеточные модели разных форматов, которые используются для тестирования возможностей ЛОГОС-Препост, генераторов сеточных моделей, импорта/экспорта данных и проверки так называемого сквозного цикла расчета, т. е. действий от импорта в ЛОГОС-Препост файла геометрической модели и подготовки расчета задачи до обработки результатов расчета;

– постановки типовых задач, предназначенные для тестирования счетных модулей, импорта/экспорта данных и проверки сквозного цикла расчета. Постановка задачи включает в себя файл, содержащий параметры задачи (в формате YAML или YAMLX), и файл с сеточной моделью, необходимые для выполнения расчета;

– постановки верификационных задач, используемые в предрелизном тестировании, для тестирования счетных модулей, импорта/экспорта данных и проверки сквозного цикла расчета. Включают как типовые, так и модельные производственно-методические задачи, в том числе характерные для базовых отраслей промышленности.

Тесты первого уровня отличаются небольшой длительностью (2–3 часа) и не требуют больших ресурсов для выполнения. Тесты второго уровня отличаются большей длительностью и могут требовать достаточно много ресурсов для выполнения. В большинстве случаев набор тестов для одного компонента выполняется в течение 2–3 дней. Тесты третьего уровня, включающие производственно-методические задачи верификационного базиса, характерные для базовых отраслей промышленности, отличаются большой длительностью (до нескольких недель) и требуют много ресурсов для выполнения.

В целях увеличения охвата тестового покрытия постоянно ведется работа по расширению тестового базиса всех уровней.

Рис. 4. Уровни тестового базиса

Автоматизация тестирования

Пакет программ ЛОГОС является сложным и постоянно развивающимся программным продуктом, при этом ресурсы участников процессов разработки и тестирования ограниченны. В связи с этим стандарты и лучшие практики в области разработки и обеспечения качества ПО рекомендуют применять инструменты автоматизации в различных процессах жизненного цикла пакета программ. Остановимся на инструментах, применяемых при автоматизации тестирования, и их интеграции в общий процесс разработки ЛОГОС.

Автоматизированное тестирование ПО – это процесс верификации ПО, при котором основные функции и шаги теста (запуск, инициализация, выполнение, анализ и выдача результата) выполняются автоматически при помощи специализированных инструментов [12]. Внедрение автоматизации тестирования дает следующие преимущества:

– исключение человеческого фактора и, как следствие, отсутствие сопутствующих ошибок;

– автоматическая генерация отчетов соответствующим инструментом по результатам тестирования;

– меньшие затраты на сопровождение разработанных скриптов автоматизации по сравнению с проведением того же объема тестирования вручную.

В целях сокращения трудовых и временных затрат на проведение тестирования, а также увеличения объема тестов и, соответственно, повышения качества ПП ЛОГОС были автоматизированы отдельные виды его тестирования. Это возможно, когда анализ выходной информации и сравнение ее с эталонной не требуют вмешательства человека. В частности, автоматизированы:

– модульное тестирование отдельных компонентов ЛОГОС;

– функциональное, сценарное и регрессионное тестирование ГПИ ЛОГОС-Препост;

– верификационное и регрессионное тестирование счетных модулей;

– сценарное и регрессионное тестирование ScientificView.

Для автоматизации тестирования часто применяются системы непрерывной интеграции. Автоматический запуск тестирования, выполняемый сразу после внесения изменений в исходный код и его построение, позволяет обнаруживать и устранять ошибки в короткие сроки. В случае ПП ЛОГОС для автоматизации тестирования используются следующие инструменты:

– система Logos Testing System (LTS) (разработка РФЯЦ-ВНИИЭФ) для верификационного регрессионного тестирования счетных модулей;

– коммерческая система Froglogic Squish [13] для тестирования пользовательского интерфейса ЛОГОС-Препост;

– инструменты Microsoft Visual Studio [14] для внутреннего тестирования компонентов ЛОГОС;

– инструмент разработки РФЯЦ-ВНИИЭФ для функционального тестирования постпроцессора ScientificView (при подготовке релиза новой версии).

В процесс разработки данные инструменты автоматизированного тестирования интегрированы посредством следующих программных продуктов:

– системы непрерывной интеграции Jenkins [15] для управление регулярным построением и запуском автоматизированного тестирования;

– системы контроля версий Mercurial Hg [11] для создания и управления репозиториями скриптов автоматизации тестирования и тестовых данных;

– системы Redmine [16] для фиксации и отслеживания всех выявленных дефектов и замечаний.

Рассмотрим подробнее некоторые особенности перечисленных инструментов для автоматизации тестирования ПП ЛОГОС.

Важной особенностью системы верификационного регрессионного тестирования LTS является возможность автоматического запуска тестов при каждом новом построении счетных модулей. На данный момент такая возможность используется для проведения тестирования первого уровня.

Тестовые задачи второго уровня запускаются каждую ночь при условии успешного прохождения тестов первого уровня.

Тесты, предназначенные для проверки решения производственно-методических задач, относятся к группе тестов третьего уровня и выполняются в случае успешного прохождения тестов второго уровня во время предрелизного тестирования.

На текущий момент в LTS подготовлены и регулярно автоматически запускаются 900 тестов, из них 350 тестов первого уровня, 527 тестов второго уровня и 23 теста третьего уровня.

С использованием системы Froglogic Squish авторами разработан набор скриптов тестирования ГПИ ЛОГОС-Препост, которые условно разделены на группы согласно проверяемым возможностям:

– импорт/экспорт данных;

– работа генераторов конечно-элементных моделей;

– работа с деталью;

– работа с геометриями;

– операции с сеточными данными;

– работа с регионами, наборами данных, подобластями;

– прохождение примеров расчетов задач из руководства пользователя;

– работа с интерфейсом программы.

На данный момент в системе непрерывной интеграции создано и поддерживается 12 планов удаленного запуска автоматизированного тестирования. Под планом подразумевается скрипт в системе непрерывной интеграции, запускающий по определенному событию (например, построение исполняемого файла) или расписанию определенный набор скриптов тестирования для

выбранного модуля, в частности компонента ЛОГОС. Все планы настроены на тестирование последней стабильной версии ЛОГОС и текущей разрабатываемой версии. Краткое описание работы, выполняемой по каждому плану, представлено в табл. 2.

Таблица 2

Планы автоматизированного тестирования

Название плана	Краткое описание	Регулярность запуска	Кол-во скриптов	Продолжительность, мин
Часовой (Windows)	Проверка основных возможностей ЛОГОС-Препост длительностью не более часа в ОС Windows	Ежедневно в 05 ч 00 мин. Ручной запуск при необходимости	35	88
Полный (Windows)	Запуск всех имеющихся ГПИ-скриптов ЛОГОС-Препост в ОС Windows	Ежедневно в 01 ч 00 мин. Ручной запуск при необходимости	127	193
Часовой (Linux)	Запуск ГПИ-скриптов на платформе Linux. Проверка основных возможностей ЛОГОС-Препост длительностью не более часа	Ежедневно в 05 ч 00 мин. Ручной запуск при необходимости	35	78
Полный (Linux)	Запуск всех имеющихся ГПИ-скриптов ЛОГОС-Препост на платформе Linux	Ежедневно в 00 ч 00 мин. Ручной запуск при необходимости	127	96
Cellclipper (Linux)	Проверка работы генераторов конечно-элементных моделей на платформе Linux	Ежедневно в 19 ч 00 мин. Ручной запуск при необходимости	17	68
Cellclipper (Windows)	Проверка работы генераторов конечно-элементных моделей в ОС Windows	Ежедневно в 19 ч 00 мин. Ручной запуск при необходимости	17	36
Cellclipper_big (Windows)	Проверка работы генераторов конечно-элементных моделей на больших задачах в ОС Windows	Ручной запуск при необходимости	16	118
ЛОГОС-Гидродинамика [17] (Windows)	Подготовка задач согласно описанию тестовых примеров из раздела ЛОГОС-Гидродинамика руководства пользователя	Ежедневно в 19 ч 30 мин. Ручной запуск при необходимости	15	37
ЛОГОС-Аэродинамика [17] (Windows)	Подготовка задач согласно описанию тестовых примеров из раздела ЛОГОС-Аэродинамика руководства пользователя	Ежедневно в 4 ч 00 мин. Ручной запуск при необходимости	11	27
ЛОГОС-Тепло [17] (Windows)	Подготовка задач согласно описанию тестовых примеров из раздела ЛОГОС-Тепло руководства пользователя	Ежедневно в 18 ч 00 мин. Ручной запуск при необходимости	14	29

Окончание табл. 2

Название плана	Краткое описание	Регулярность запуска	Кол-во скриптов	Продолжительность, мин
ЛОГОС-Прочность [18] (Windows)	Подготовка задач согласно описанию тестовых примеров из раздела ЛОГОС-Прочность руководства пользователя	Ежедневно в 20 ч 00 мин. Ручной запуск при необходимости	39	100
HYBRID (Windows)	Проверка основных возможностей модели ЛОГОС-Прочность по работе с геометриями и построению конечно-элементных моделей	Ручной запуск при необходимости	31	43
Любой (Windows)	Запуск любого плана тестирования по запросу разработчика на указанной им конкретной версии ЛОГОС-Препост	Ручной запуск при необходимости	35	41

Регулярное проведение автоматизированного тестирования позволяет значительно экономить время тестировщиков и выявлять примерно 60 % дефектов разрабатываемого ПО.

Заключение

В целях обеспечения высокого качества ПП ЛОГОС коллективом авторов была разработана рассмотренная в статье методика тестирования. При ее создании были учтены особенности самого ПП ЛОГОС и процесса его разработки. Методика тестирования была внедрена в жизненный цикл ПП ЛОГОС и активно применяется в настоящее время. Также авторами методики были подобраны и внедрены в рабочий процесс инструменты автоматизации тестирования.

Силами участников процессов разработки и тестирования ПП ЛОГОС был сформирован тестовый базис и постоянно ведется работа по его расширению в целях увеличения тестового покрытия.

Для эффективного обеспечения высокого качества ПП ЛОГОС были разработаны собственные организационные документы, регламентирующие процесс разработки и тестирования ПП ЛОГОС.

Применение на практике описанной методики тестирования позволяет выявлять дефекты в минимальные сроки на возможно ранних этапах процесса разработки, что способствует повышению качества ПП ЛОГОС.

Список литературы

1. Погосян М. А., Савельевских Е. П., Шагалиев Р. М. и др. Применение отечественных суперкомпьютерных технологий для создания перспективных образцов авиационной техники // Вопросы атомной науки и техники. Сер. Матем. моделирование физ. процессов. 2013. Вып. 2. С. 3–17.

2. Барабанов Р. А., Дьянов Д. Ю., Каныгин И. И. и др. Пакет программ ЛОГОС. Метод решения задач статической прочности тонкостенных и стержневых конструкций на основе Solid-Shell-технологии // Там же. 2016. Вып. 3. С. 26–36.

3. ГОСТ Р ИСО/МЭК 12207-2010. Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств [Электронный ресурс]. – http://docs.cntd.ru/document/gost-r-iso-mek-12207-2010.

4. ГОСТ Р ИСО/МЭК 15271-2002. Информационная технология. Руководство по применению ГОСТ Р ИСО/МЭК 12207 (Процессы жизненного цикла программных средств) [Электронный ресурс]. – http://docs.cntd.ru/document/gost-r-iso-mek-to-15271-2002.

5. ГОСТ Р 56920-2016/ISO/IEC/IEEE 29119-1:2013. Системная и программная инженерия. Тестирование программного обеспечения. Часть 1. Понятия и определения [Электронный ресурс]. – http://docs.cntd.ru/document/ 1200134996.

6. ГОСТ Р ИСО/МЭК 12119-2000. Информационная технология. Пакеты программ. Требования к качеству и тестирование [Электронный ресурс]. – http://docs.cntd.ru/document/1200025075.

7. SWEBOK (Software Engineering Body Knowledge). Software Engineering Coordinating Committee, 1998–2013 [Electronic resource]. – http://www.swebok.org.

8. Липаев В. В. Программная инженерия сложных заказных программных продуктов : учеб. пособие. – М.: МАКС Пресс, 2014.

9. Анищенко А. А., Санталов А. С., Дюпин В. Н., Дерюгин В. И. Подход к автоматическому построению пользовательского интерфейса для задания параметров расчетных методик в препостпроцессоре ЛОГОС-Препост // Вопросы атомной науки и техники. Сер. Матем. моделирование физ. процессов. 2014. Вып. 3. С. 78–84.

10. Потехин А. Л., Тарасов В. И., Фирсов С. А. и др. ScientificView – параллельная система постобработки результатов, полученных при численном моделировании физических процессов // Там же. 2008. Вып. 4. С. 37–45.

11. Official site of Mercurial program [Electronic resource]. – https://www.mercurial.selenic.com.

12. Дастин Э., Рэшка Дж., Пол Дж. Автоматизированное тестирование программного обеспечения. Внедрение, управление и эксплуатация / Пер. с англ. – М.: Лори, 2003.

13. Official site of Froglogic Squish program [Electronic resource]. – http://www.froglogic.com/squish.

14. Microsoft Visual Studio [Электронный ресурс]. – https://www.visualstudio.com/ru.

15. Official site of Jenkins [Electronic resource]. – https://jenkins.io.

16. Official site of Redmine program [Electronic resource]. – https://www.redmine.org.

17. Козелков А. С., Дерюгин Ю. Н., Зеленский Д. К. и др. Многофункциональный пакет программ ЛОГОС: физико-математические модели расчета задач аэро-, гидродинамики и тепломассопереноса : препринт № 111. – Саров: РФЯЦ-ВНИИЭФ, 2013.

18. Александрова О. Л., Барабанов Р. А., Дьянов Д. Ю. и др. Пакет программ ЛОГОС. Конечно-элементная методика расчета задач статической прочности конструкций с учетом эффектов физической и геометрической нелинейности // Вопросы атомной науки и техники. Сер. Матем. моделирование физ. процессов. 2014. Вып. 3. С. 3–17.

Testing Procedure for the LOGOS Software Package

N. S. Averina, T. Yu. Bakanova, M. G. Lashmanova, E. A. Savinykh, T. N. Serova

The paper considers the problem of quality assurance for complex software products by the example of the LOGOS software package developed at RFNC-VNIIEF. The testing procedure developed and used to assure a high quality of the LOGOS software package, the types of test efforts for various LOGOS components, and frequency of testing are described. Also, the data used for testing are considered. The description of how to use tools to automate the execution of tests is given.