

УДК 519.6

**МЕТОДИКА ТЕСТИРОВАНИЯ ПАКЕТА ПРОГРАММ "ЛОГОС"**

Н. С. Аверина, Т. Ю. Баканова, М. Г. Лашманова, Е. А. Савиных, Т. Н. Серова  
(ФГУП "РФЯЦ-ВНИИЭФ", г. Саров Нижегородской области)

Рассматривается проблема обеспечения качества сложных программных продуктов на примере разработанного во ФГУП "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. Также рассматриваются используемые для тестирования данные. Приведено описание применения инструментов автоматизации тестирования.

*Ключевые слова:* тестирование, пакет программ ЛОГОС, тестовый базис, методика, виды тестирования, качество программного обеспечения.

**Введение**

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 требуют 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. Соответственно рассмотрен-

ные далее подходы к организации обеспечения качества ПП ЛОГОС, выработанные на основании требований и рекомендаций государственных стандартов, также 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 выявление отличий между его реально существующими и требуемыми свойствами и на оценку ПО.

Стратегической целью работ по тестированию является минимизация количества ошибок в программном продукте так, 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 их состав, к конкретной ситуации. Соответственно первым шагом при выработке подходов к организации обеспечения качества ПП ЛОГОС стало прове-

дение анализа особенностей процесса разработки и непосредственно самого ПП.

### Особенности ПП ЛОГОС и процесса его разработки

К особенностям ПП ЛОГОС можно отнести следующее:

- включение в его состав разнородных компонентов: консольных приложений счетных модулей; приложений препостпроцессинга ЛОГОС-Препост [9] и ScientificView [10] с графическим пользовательским интерфейсом (ГПИ); модуля документации; инсталляционного модуля;
- кросс-платформенность реализации;
- возможность проведения расчетов задач как в локальном, так и удаленном режиме;
- 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 для работы библиотек и т. д.;
- ключевые ежегодные события: плановые выпуски версий (так называемых 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 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 режиме. 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 должного высокого качества, а также способствовать максимально быстрому выявлению дефектов ПО.

Выработанная таким образом методика тестирования ЛОГОС описана далее.

### Виды и уровни тестирования ПП ЛОГОС

С учетом всех вышеперечисленных 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, нестандартной последовательности действий пользователя) для проверки устойчивости программного продукта к различного рода воздействиям и проверки 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, направленное на обнаружение ошибок в уже проверенных участках программ (или исходных кодах). В том числе проверяется, что проведенные 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, набо-

- ры 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, таких как, 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 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 посредством 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 точность указаний по установке, настройке и запуску ПО. Также проверяются 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 непосредственно разработчиками каждого компонента.

Тестирование первого уровня имеет следующие особенности:

- оно 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 тестирование, проводимое силами команды тестировщиков с периодичностью раз в месяц и обладающее следующими особенностями:

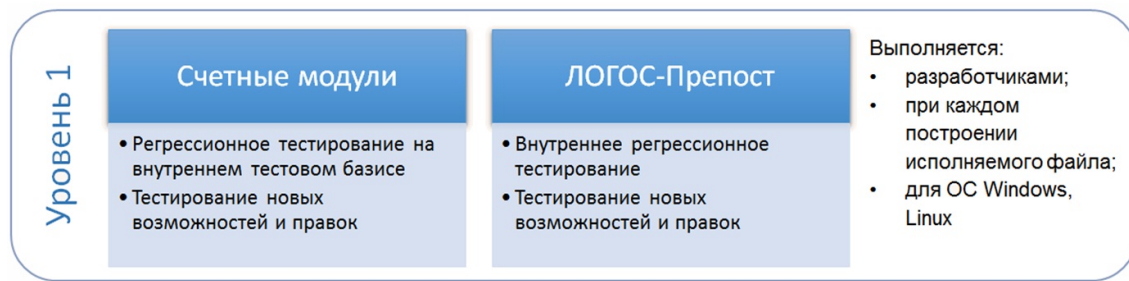


Рис. 1. Уровень 1. Внутреннее тестирование компонентов ЛОГОС

- тестируются все компоненты ЛОГОС, кроме инсталляционного модуля и пользовательской документации;
- выполняется регрессионное тестирование счетных модулей и ЛОГОС-Препост, особенно для периодически проявляющихся ошибок;
- выполняются верификационное тестирование счетных модулей, функциональное тестирование новых и измененных возможностей компонентов препостпроцессинга, сценарное тестирование всего ПП ЛОГОС;
- выполняется на отобранных типовых задачах тестового базиса (см. далее), в том числе задачах из обучающих материалов руководства пользователя;
- тестирование выполняется для согласованных стандартных конфигураций ОС Windows и Linux.

К третьему уровню (рис. 3) отнесено тестирование ПП ЛОГОС, выполняемое при подготовке версии к плановому выпуску (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 предприятия.

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;
- выполняются верификационное тестирование счетных модулей, функциональное тестирование новых и измененных возможностей компонентов препостпроцессинга, сценарное тестирование всего ПП ЛОГОС. 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. 2. Уровень 2. Межрелизное тестирование



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 ПП и соответственно завершения тестирования в рамках подготовки версии к выпуску:

- 1) 80 % всех видов тестов должны проходить успешно;
- 2) в системе отслеживания дефектов для конкретной версии программного продукта из всех зарегистрированных ошибок с указанным приоритетом открытых (неисправленных) должно оставаться не более допустимого количества (табл. 1).

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

В случае невыполнения критериев из табл. 1 версия считается не готовой к выпуску.

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

Таблица 1

**Допустимое количество открытых ошибок**

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

с заключением о готовности или неготовности к выпуску дистрибутива.

**Тестовый базис ПП 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 используется на этих уровнях соответственно.

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. 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



Рис. 4. Уровни тестового базиса

- модели и подготовки расчета задачи до обработки полученных результатов;
- постановки типовых задач. Предназначены 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. Используются в предрелизном тестировании, а также 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 тестового покрытия постоянно ведется работа по расширению тестового базиса всех уровней.

### Автоматизация тестирования

ПП ЛОГОС является сложным и постоянно развивающимся программным продуктом, при этом ресурсы участников процессов разработки и тестирования ограничены. В связи с этим стандарты и лучшие практики в области разработки и обеспечения качества ПО рекомендуют применять инструменты автоматизации в различных процессах жизненного цикла ПП. Останемся на инструментах, применяемых при автоматизации тестирования, и их интеграции в общий процесс разработки ПП 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 предоставляет следующие преимущества:

- исключение человеческого фактора, что приводит к отсутствию сопутствующих ошибок;



- автоматическая генерация отчетов соответствующим инструментом по результатам проведения 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 построения, 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 пост-процессора ScientificView (при подготовке выпуска новой версии).

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

- система непрерывной интеграции Jenkins [15] — для управления регулярным построением и запуском автоматизированного тестирования;
- система контроля версий Mercurial Hg [11] — для создания и управления репозиториями скриптов автоматизации тестирования и тестовых данных;
- система Redmine [16] — для фиксации и отслеживания всех выявленных дефектов и замечаний.

Рассмотрим подробнее использование 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 пред релизного тестирования.

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

С использованием системы Froglogic Squish авторами разработан набор скриптов тестирования ГПИ ЛОГОС-Препост, которые условно разбиты 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, под-областями;
- прохождение примеров расчетов задач из руководства пользователя;
- работа с интерфейсом программы.

Таблица 2

**Планы скриптов автоматизированного тестирования ГПИ**

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 04 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
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



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 из разных групп для выбранного модуля, в частности, компонента ПП 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.

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. Данная методика тестирования была внедрена в его жизненный цикл и активно применяется в настоящее время. Также авторами были подобраны и внедрены в рабочий процесс инструменты автоматизации тестирования. Поддержка тестирования и его интеграция в процесс разработки обеспечиваются современными технологиями и инструментами.

Всеми участниками 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 процессы разработки и тестирования ПП ЛОГОС.

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

### Список литературы

1. *Погосян М. А., Савельевских Е. П., Шагалев Р. М., Козелков А. С., Стрелец Д. Ю.,*

*Рябов А. А., Корнев А. В., Дерюгин Ю. Н., Спиридонов В. Ф., Циберев К. В.* Применение отечественных суперкомпьютерных технологий для создания перспективных образцов авиационной техники // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2013. Вып. 2. С. 3–17.

*Pogosyan M. A., Savelevskikh E. P., Shagaliev R. M., Kozelkov A. S., Strelets D. Yu., Ryabov A. A., Kornev A. V., Deryugin Yu. N., Spiridonov V. F., Tsiberev K. V.* Primenenie otechestvennykh superkompyuternykh tekhnologiy dlya sozdaniya perspektivnykh obraztsov aviatsionnoy tekhniki // Voprosy atomnoy nauki i tekhniki. Ser. Matematicheskoe modelirovanie fizicheskikh protsessov. 2013. Vyp. 2. S. 3–17.

2. *Барабанов Р. А., Дьянов Д. Ю., Каньгин И. И., Спиридонов В. Ф., Филлимонкин Е. А., Циберев К. В.* 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.

*Barabanov R. A., Dyanov D. Yu., Kanygin I. I., Spiridonov V. F., Filimonkin E. A., Tsiberev K. V.* Paket programm LOGOS. Metod resheniya zadach staticheskoy prochnosti tonkostennykh i sterzhnevyykh konstruktсий na osnove SOLID-SHELL-tekhnologii // Tam zhe. 2016. Vyp. 3. S. 26–36.

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

GOST R ISO/MEK 12207-2010. Informatsionnaya tekhnologiya. Sistemnaya i programmaya inzheneriya. Protsessy zhiznennogo tsikla programmnykh sredstv. <http://docs.cntd.ru/>

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

GOST R ISO/MEK 15271-2002. Informatsionnaya tekhnologiya. Rukovodstvo po prime-

- neniyu GOST R ISO/MEK 12207 (Protsessy zhiznennogo tsikla programmnykh sredstv). <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. Ч. 1. Понятия и определения. <http://docs.cntd.ru/document/1200134996>. GOST R 56920-2016/ISO/IEC/IEEE 29119-1:2013. Sistemnaya i programmnyaya inzheneriya. Testirovanie programmnoho obespecheniya. Ch. 1. Ponyatiya i opredeleniya. <http://docs.cntd.ru/document/1200134996>.
  6. ГОСТ Р ИСО/МЭК 12119-2000. Информационная технология. Пакеты программ. Требования к качеству и тестирование. <http://docs.cntd.ru/document/1200025075>. GOST R ISO/MEK 12119-2000. Informatsionnaya tekhnologiya. Pakety programm. Trebovaniya k kachestvu i testirovanie. <http://docs.cntd.ru/document/1200025075>.
  7. SWEBOK (Software Engineering Body Knowledge). Software Engineering Coordinating Committee, 1998–2013. <http://www.swebok.org/>.
  8. *Липаев В. В.* Программная инженерия сложных заказных программных продуктов: Уч. пособие. М.: МАКС Пресс, 2014. *Lipaev V. V.* Programmnyaya inzheneriya slozhnykh zakaznykh programmnykh produktov: Uch. posobie. М.: МАКС Press, 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. Математическое моделирование физических процессов. 2014. Вып. 3. С. 78–84. *Anishchenko A. A., Santalov A. S., Dupin V. N., Deryugin V. I.* Podkhod k avtomaticheskomu postroeniyu polzovatel'skogo interfeysa dlya zadaniya parametrov raschetnykh metodik v prepostprotsessore LOGOS-Prepost // Voprosy atomnoy nauki i tekhniki. Ser. Matematicheskoe modelirovanie fizicheskikh protsessov. 2014. Vyp. 3. S. 78–84.
  10. *Потехин А. Л., Тарасов В. И., Фирсов С. А., Логинов И. В., Никитин В. А., Кузнецов М. Г., Попова Н. В., Деманова А. К., Козачек Ю. В.* ScientificView — параллельная система постобработки результатов, полученных при численном моделировании физических процессов // Там же. 2008. Вып. 4. С. 37–45. *Potekhin A. L., Tarasov V. I., Firsov S. A., Loginov I. V., Nikitin V. A., Kuznetsov M. G., Popova N. V., Demanova A. K., Kozachek Yu. V.* ScientificView — parallelnaya sistema postobrabotki rezultatov, poluchennykh pri chislennom modelirovanii fizicheskikh protsessov // Tam zhe. 2008. Vyp. 4. S. 37–45.
  11. Mercurial program. <http://docs.cntd.ru/document/1200025075>.
  12. *Дастин Э., Рэшка Дэс., Пол Дэс.* Автоматизированное тестирование программного обеспечения. Внедрение, управление и эксплуатация: Пер. с англ. М.: Лори, 2003. *Dastin E., Reshka Dzh., Pol Dzh.* Avtomatizirovannoe testirovanie programmnoho obespecheniya. Vnedrenie, upravlenie i ekspluatatsiya: Per. s angl. М.: Lori, 2003.
  13. Froglogic Squish program. [www.froglogic.com/squish](http://www.froglogic.com/squish).
  14. Microsoft Visual Studio. <http://www.visualstudio.com/ru/>.
  15. Jenkins. <http://jenkins.io>.
  16. Redmine program. <http://www.redmine.org>.
  17. *Козелков А. С., Дерюгин Ю. Н., Зеленский Д. К., Полищук С. Н., Лашкин С. В., Жучков Р. Н., Глазунов В. А., Яцевич С. В., Курюлин В. В.* Многофункциональный пакет программ ЛОГОС: физико-математические модели расчета задач аэро-, гидродинамики и тепломассопереноса: Препринт № 111. Саров: РФЯЦ-ВНИИЭФ, 2013. *Kozelkov A. S., Deryugin Yu. N., Zelen'skiy D. K., Polishchuk S. N., Lashkin S. V., Zhuchkov R. N., Glazunov V. A., Yatsevich S. V., Kurulin V. V.* Mnogofunktsionalny paket programm LOGOS: fiziko-matematicheskie modeli rascheta zadach aero-, gidrodinamiki i teplomassoperenosa: Preprint № 111. Sarov: RFYaTs-VNIIEF, 2013.
  18. *Александрова О. Л., Барабанов Р. А., Дьянов Д. Ю., Косарим С. С., Наумов А. О., Спиридонов В. Ф., Филимонкин Е. А., Циберев К. В.* Пакет программ ЛОГОС.

Конечно-элементная методика расчета задач статической прочности конструкций с учетом эффектов физической и геометрической нелинейности // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2014. Вып. 3. С. 3–17.

*Aleksandrova O. L., Barabanov R. A., Dyanov D. Yu., Kosarim S. S., Naumov A. O., Spiridonov V. F., Filimonkin E. A., Tsibe-*

*rev K. V. Paket programm LOGOS. Konечно-elementnaya metodika rascheta zadach staticheskoy prochnosti konstruktсий s uchetom effektov fizicheskoy i geometricheskoy nelineynosti // Voprosy atomnoy nauki i tekhniki. Ser. Matematicheskoe modelirovanie fizicheskikh protsessov. 2014. Vyp. 3. S. 3–17.*

Статья поступила в редакцию 24.05.19.

TESTING PROCEDURE FOR THE LOGOS SOFTWARE PACKAGE / N. S. Averina, T. Yu. Bakanova, M. G. Lashmanova, E. A. Savinykh, T. N. Serova (FSUE "RFNC-VNIIEF", Sarov, N. Novgorod region).

The paper considers the problem of quality assurance for complex software products by the example of the LOGOS software package developed at FSUE "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.

*Keywords:* testing, LOGOS software package, test basis, procedure, types of test efforts, quality of software.

---