

# Формирование компетенций управления качеством программного обеспечения в вузе

Говорится о важности комплексного подхода к освоению технологий и инструментов тестирования в рамках проектной деятельности по разработке программного обеспечения. Обсуждается многоуровневое освоение компетенций управления качеством и тестирования программного обеспечения от простого к сложному, роль тестирования и значимость организации процесса управления качеством



## Р.М. Хамитов<sup>1</sup>

Казанский государственный энергетический университет (КГЭУ), канд. техн. наук, доцент, hamitov@gmail.com

## Н.К. Петрова<sup>1</sup>

КГЭУ, канд. физ.-мат. наук, доцент, nk\_petrova@mail.ru

## А.Р. Низамова<sup>2</sup>

КГЭУ, alinarafaelevna\_kgeu@mail.ru

Информационные технологии — одно из приоритетных направлений развития науки, технологии и техники в Российской Федерации [1]. Для обеспечения рынка труда квалифицированными кадрами Институт цифровых технологий и экономики Казанского государственного энергетического университета (далее — Институт) ведет подготовку специалистов в области информационных технологий (ИТ) по следующим направлениям:

- ▶ 01.03.04 — Прикладная математика (бакалавриат);
- ▶ 09.03.01 — Информатика и вычислительная техника (бакалавриат);
- ▶ 09.03.03 — Прикладная информатика (бакалавриат);
- ▶ 15.03.06 — Мехатроника и робототехника (бакалавриат);
- ▶ 09.04.01 — Информатика и вычислительная техника (магистратура);
- ▶ 15.04.06 — Мехатроника и робототехника (магистратура).

Оценить качество образования выпускников Институту помогает обратная связь с работодателями республики и соседних регионов. Ее результаты говорят о том, что работодатели в первую очередь заинтересованы в соискателях, которые владеют современными инструментами и технологиями, обладают профессиональными коммуникациями, что позволяет им стать успешными специалистами.

В то же время сегодня среди проблем высшего образования нужно выделить недостаток у студентов мотивации на применение освоенных навыков и знаний [2]. Как правило, мотивация приходит уже в процессе трудовой деятельности, когда без применения полученных знаний и умений невозможно решать конкретные практические задачи. Следовательно,

формирование у обучающихся мотивации на изучение новых инструментов, технологий и средств для достижения качественного результата своей профессиональной деятельности остается одной из актуальных задач высшей школы.

Важным этапом разработки программного обеспечения (ПО) является процесс управления качеством и тестирования [3–5]. В образовательных программах вузов данный процесс рассматривается лишь в теоретическом аспекте, а тестирование ограничивается демонстрационными практически заданиями. В работах [6–9] тестирование программного обеспечения рассмотрено в отрыве от управления качеством, вне комплексного подхода в рамках полного цикла обучения. Поэтому студентам трудно сформировать навыки прохождения внешнего контроля полученных знаний и самопроверки разработанных алгоритмов в аспекте оценки качества.

## Знакомство с процессом управления качеством программного обеспечения

Освоение языков программирования — важный этап становления специалиста в области информационных технологий, но, как правило, оно ограничивается формальным описанием языковых конструкций и способов их применения. Поэтому формирование культуры оформления кода и навыков проверки результатов является основной задачей студентов младших курсов.

В Институте в качестве первого и базового языка выступает язык C++. Конечным результатом учебного курса «Алгоритмизация и программирование на C++» становится создание

<sup>1</sup> доцент, г. Казань, Республика Татарстан, Россия

<sup>2</sup> инженер кафедры, г. Казань, Республика Татарстан, Россия

Для цитирования: Хамитов Р.М., Петрова Н.К., Низамова А.Р. Формирование компетенций управления качеством программного обеспечения в вузе // Компетентность / Competency (Russia). — 2021. — № 5. DOI: 10.24412/1993-8780-2021-5-16-22

### ключевые слова

информационные технологии, квалифицированные кадры, мотивация, новые инструменты, качественный результат труда

отдельных приложений в рамках конкретных практических работ и модулей приложения, разрабатываемого в рамках курсовой работы по данной дисциплине. Полученный программный код обычно обладает низким качеством: отсутствуют комментарии, оформление кода не соответствует выбранным стандартам, встречаются различные ошибки и дефекты программирования.

В ходе освоения данной дисциплины студенты приобретают базовые навыки в тестировании и отладке программного обеспечения, а выполняя практические работы, осваивают технологии и инструменты тестирования, применяют полученные знания для тестирования модулей при выполнении курсовой работы.

Инструментами выступают среда тестирования Visual Studio Community [10] и Google Test [11]. В Visual Studio 2017 и более поздних версиях решение Google Test интегрировано в среду Visual Studio как компонент рабочей нагрузки «Разработка классических приложений на C++». Полезные функции среды разработки, например автоформатирование и автозавершение, встроенный модуль для запуска тестов и формирования отчетов, позволяют студентам повысить качество программного кода. При освоении среды Visual Studio Community и Google Test обучающимся необходимо скомпилировать и отладить простое приложение, содержащее разработанный модуль, а также написать несколько тестов для функций этого модуля. В ходе подготовки курсовой работы студенты закрепляют полученные навыки в организации многомодульных приложений, к выделению компонент приложения. Кроме того, они учатся решать различные технические проблемы (например, разработанная функция не возвращает результат работы, а выводит его на консоль). Таким образом, становятся видны проблемы некачественной разработки программного обеспечения.

В ходе занятий обучающиеся разбираются на пары и тестируют проекты по разработке программного средства.

---

Формирование у обучающихся мотивации на изучение новых инструментов, технологий и средств для достижения качественного результата своей профессиональной деятельности остается одной из актуальных задач высшей школы

---

После завершения курсового проекта студенты получают рабочий программный продукт с набором тестов и отчеты о результатах тестирования. Коллективная деятельность приобщает будущих специалистов к качественному написанию кода и ответственности за результаты своего труда.

Освоение дисциплины «Алгоритмизация и программирование на C++» формирует у студентов понимание важности получения качественного программного продукта, знакомит на практике с инструментами и средствами разработки и тестирования. Результаты выполнения курсового проекта могут быть добавлены в портфолио студента, что может заинтересовать будущих работодателей.

### **Роль тестирования в жизненном цикле программного обеспечения**

**Д**ля бакалавров третьего года обучения специальностям «Технология разработки программного обеспечения», «Программное обеспечение средств вычислительной техники и автоматизированных систем» в КГЭУ читается курс «Производственная практика (проектная)». Освоение дисциплины предполагает использование проектных групп и выполнение группового проекта по разработке ПО.

Следует отметить, что если студент I курса получает тему проекта согласно своим вариантам, то на III курсе к составлению списка проектов привлекаются как сотрудники КГЭУ, так и потенциальные работодатели, которым важен результат. Проектная группа

## Тестирование и документирование процесса разработки программного обеспечения являются ключевыми аспектами создания качественного программного обеспечения

выбирает проект также с учетом мнения заказчика. Подобная схема работы позволяет мотивировать студентов на получение качественного продукта и зарекомендовать себя с профессиональной точки зрения.

Работа над проектом состоит из нескольких этапов: сбор и анализ требований к разрабатываемому программному продукту, разработка проекта и составление плана тестирования, написание программного кода, тестирование и аттестация проекта.

Рассмотрим детали обучения управлению качеством.

Проектная деятельность на этапе тестирования предполагает совместную работу с одними и теми же документами или кодом разных сотрудников, поэтому описания тестов должны быть подробными и понятными, а программный код выдержан в принятом стиле и документирован, что позволит формировать у студентов ответственность за свой труд.

На написание собственно кода отводится лишь небольшая часть времени, его основная часть связана с разработкой и написанием документации по проекту.

В качестве инструментов совместной работы над документами могут быть выбраны технологии Git [12], MediaWiki [13] и другие. В этом случае обучающиеся и преподаватели получают следующие преимущества:

- ▶ доступ к документации из любого места при наличии компьютера и сети интернет;
- ▶ возможность восстановить ранее измененное содержимое;
- ▶ возможность совместно и параллельно наполнять содержимое докумен-

тации без необходимости находиться в одном месте;

- ▶ возможность отслеживать динамику содержимого страниц и персональную ответственность за изменения.

Одним из этапов создания проекта по разработке ПО является написание и подготовка тестов. Причем их подготовка идет до написания кода, и это — важнейшее технологическое требование к проектным группам. На данном этапе многие студенты ошибаются, поэтому задача курса — показать преимущества методологии разработки через тестирование. Таким образом вырабатываются навыки абстрактного мышления, предугадывания возможных ситуаций, возникающих в ходе использования результатов проекта.

Подготовке тестов предшествует создание раздела, посвященного спецификации требований, в котором необходимо перечислить критерии аттестации проекта, описывающие, как будет проходить его проверка на соответствие требованиям.

Данный этап направлен на формирование у студентов организационно-управленческих навыков, таких как составление и контроль плана выполняемой работы, планирование необходимых ресурсов, оценка результатов своего труда, а при необходимости адаптация вида и характера своей профессиональной деятельности.

### Организация процесса управления качеством и тестирования программного обеспечения

**Н**а IV курсе дисциплина «Управление качеством и тестирование программного обеспечения» совместно с параллельно изучаемыми дисциплинами «Проектный практикум по управлению разработкой и разработке программного обеспечения» и «Документирование и сертификация программного обеспечения» обобщают полученные теоретические знания и практические навыки обучающихся.

В курсе «Управление качеством и тестирование программного обеспе-

чения» рассматриваются методические вопросы организации процесса управления качеством ПО: подходы к анализу предметной области и планированию тестирования, методы построения тестов, методы анализа исходного кода программных продуктов, средств автоматизации тестирования.

Основной упор делается на организационные моменты. Студенты знакомятся с такими ключевыми моделями создания программного обеспечения, как разработка через тестирование (англ. test-driven development) и разработка через поведение (behavior driven development). Применение этих методологий позволяет представить тестирование как инструмент контроля качества и обеспечения выполнения требований заказчика.

Важным моментом процесса тестирования являются планирование и документирование. Студенты приобретают практические навыки в области разработки и оформления плана и журнала тестирования, протоколов ошибок, отчета о тестировании. При этом используются общепринятые системы и стандарты документирования: ЕСПД [14, 15] и IEEE [16–18].

В ходе практических занятий студенты выполняют разработку тестов и оформление документации проекта на тему, соответствующую теме выпускной квалификационной работы, придерживаясь следующей структуры документации:

1. Описание объекта, перечень функциональных требований, границы тестирования.
2. План тестирования. Перечень модульных, интеграционных, аттестационных и специальных тестов.
3. Журнал тестирования. Дата, тестирущик, объект тестирования, перечень выполненных тестов с указанием количества запусков, перечень найденных ошибок.
4. Журнал ошибок. Номер отчета об ошибке, дата составления отчета, номер теста, ожидаемый результат, фактический результат.
5. Отчет о тестировании. Оценка качества результатов тестирования.

Подготовка к тестированию описывается в отдельном документе, который называется планом тестирования. Согласно ГОСТ Р 56922–2016 план должен включать следующие разделы:

1. Спецификация документа.
2. Введение.
3. Контекст тестирования.
4. Обмен информацией о тестировании.
5. Реестр рисков.
6. Стратегия тестирования.
7. Действия и оценка тестирования.
8. Комплектность персонала.
9. Расписание.

Студенты могут самостоятельно выбрать набор инструментов и средств для проведения тестирования.

В рамках курса требуется применить как минимум четыре группы вариантов тестов:

- ▶ модульные тесты по методу «белого ящика» для тестирования одного изолированного участка модуля кода (обычно это методы класса или функции модуля). Описание должно включать набор входных параметров, предусловия для выполнения теста и ожидаемый результат. Пример описания тестовых случаев одного учебного проекта приведен на рис. 1;
- ▶ интеграционные тесты для проверки взаимодействия между модулями или классами. Перед проведением данных тестов составляется последователь-

**Рис. 1.** Пример описания тест-кейса  
 [Example of a test case description]

<b>Идентификатор [ID]</b>	<b>UC-3 Ввод команд и ответов чат-бота [UC-3 Entering chatbot commands and responses]</b>
<b>Действующее лицо</b>	Пользователь
<b>Описание</b>	Пользователь вводит в специальную форму команды, на которые реагирует чат-бот, и ответные сообщения Система принимает и записывает команды и ответы
<b>Триггер</b>	Ввод сообщений в форму «Команды бота»
<b>Предусловия</b>	PRE-1. Платформа выбрана PRE-2. Код доступа получен PRE-3. Код доступа записан
<b>Выходные условия</b>	POST-3. Сообщения бота записаны
<b>Нормальное направление</b>	1. Пользователь вводит команду, на которую реагирует бот 2. Пользователь вводит ответное сообщение бота 3. Система принимает сообщения и сохраняет бот
<b>Альтернативное направление</b>	Нет
<b>Исключения</b>	3.0. E1. Сбой системы

## Многоуровневое обучение освоению компетенций управления качеством и тестирования ПО от простого к сложному позволяет закрепить у выпускников навыки создания качественного программного продукта и написания тестов

ность, в которой будут тестироваться и добавляться новые модули или группы модулей в виде схемы интеграции;

- ▶ тесты пользовательского интерфейса по методу «черного ящика». Тесты по возможности должны использовать средства автоматизации тестирования;
- ▶ аттестационные тесты, направленные на комплексную оценку качества программного продукта на соответствие требованиям спецификации.

Для визуализации покрытия кода тестами составляется матрица соответствия требований, в которой по столбцам перечислены тесты, а по строкам — требования. В ячейках отмечается, как данный тест проверяет данное требование.

Далее следует самый важный этап с точки зрения обеспечения качества ПО — проведение тестирования, в ходе которого оформляется журнал тестирования (или отчет о выполнении тестирования). Для каждого проведенного теста оформляется запись, содержащая следующую информацию: дата проведения теста, описание теста (ссылка на тест из документа плана тестирования), номер попытки, исполнитель, входные данные, выходные данные и результат. Пример тестового журнала приведен на рис. 2.

**Рис. 2.** Пример журнала тестирования  
[Sample test log]

Идентификатор [ID]	Дата [Date]	Тестировал [Tested]	Попытки [Attempts]	Входные данные [Input data]	Ожидаемый результат [Expected result]	Фактический результат [Actual result]
1	1.03.2021	Бурунин В.В.	3	1. Открываем web-сервис INDIGO 2. Кликаем на выпадающий список с языками интерфейса	Присутствует язык интерфейса “简单汉语”	Отсутствует язык интерфейса “简单汉语”

При выявлении дефекта в программном обеспечении студент-тестировщик должен составить описание возникновения данного дефекта: последовательность шагов, приводящих к его появлению, тестовое окружение (в том числе используемые аппаратные и программные средства). От качества представленной информации зависят качество и скорость устранения дефекта программистами. Подобный подход формирует у студентов чувство ответственности за проделанную работу и стимулирует их на применение полученных знаний и навыков для достижения результата.

Завершающий этап разработки — составление общего отчета о тестировании, в котором подсчитываются метрики проекта, в том числе количество запланированных тестов, количество проведенных тестов, количество выявленных и задокументированных дефектов. Строятся диаграммы метрик, которые позволяют количественно оценить качество программного продукта.

### Анализ комплексного подхода к обеспечению качества программного обеспечения

Тестирование и документирование процесса разработки программного обеспечения являются ключевыми аспектами создания качественного программного обеспечения. Применяемый в КГЭУ подход к подготовке ИТ-специалистов в части компетенции управления качеством и тестировании ПО позволяет добиться у обучающихся понимания необходимости разработки качественного программного обеспечения, знания средств и методов достижения этого, приобрести навыки владения инструментами тестирования, коллективной

ответственности за выполняемую работу [19].

Многоуровневое обучение освоению компетенций управления качеством и тестирования ПО от простого к сложному (изучение инструментов

тестирования, командная разработка, организация процесса управления) позволяет получить и закрепить у выпускников навыки создания качественного программного продукта и написания тестов. ■

*Статья поступила  
в редакцию 6.03.2021*

## Список литературы

1. Указ Президента РФ от 7.07.2011 № 899 «Об утверждении приоритетных направлений развития науки, технологий и техники в Российской Федерации и перечня критических технологий Российской Федерации»; <http://www.kremlin.ru/acts/bank/33514> (дата обращения: 20.03.2021).
2. Малаион С.Ф., Куценко С.М. Повышение мотивации обучающихся при изучении общеобразовательных дисциплин в высшей школе // Современное образование: содержание, технологии, качество. — 2018. — Т. 1.
3. Сафиуллина Л.Х., Торкунова Ю.В., Исмагилов И.Р., Красильников О.В. Организация интеграционного тестирования в задачах системного программирования // Ученые записки ИСГЗ. — 2018. — Т. 16. — № 2.
4. Абдульмянов Т.Р., Петрова Н.К. Основные этапы разработки и тестирования высокоточных специализированных интеграторов // Вестник КГЭУ. — 2015. — № 2(26). ISSN 2072-6007.
5. Куликов С. Тестирование программного обеспечения. Базовый курс. Хроники детерминированности; [http://svyatoslav.biz/software\\_testing\\_book/](http://svyatoslav.biz/software_testing_book/) (дата обращения: 20.03.2021).
6. НОУ ИНТУИТ Основы тестирования программного обеспечения. Информация; <https://intuit.ru/studies/courses/48/48/info> (дата обращения: 20.03.2021).
7. Вишневская Т.И. Тестирование программного обеспечения как учебная дисциплина // Образовательные ресурсы и технологии. — 2014. — № 4.
8. Чикунов В.Д., Ломаш Д.А. Новые подходы к подготовке IT-специалистов в современных условиях / Транспорт: наука, образование, производство: сб. науч. трудов Межд. науч.-практ. конф. — 2017.
9. Полевщиков И.С. Особенности изучения способа тестирования базового пути студентами бакалавриата в рамках дисциплины Тестирование программного обеспечения (часть 1) // Молодой ученый. — 2015. — № 18.
10. Visual Studio Community 2019 — Free IDE and Developer Tools; <https://visualstudio.microsoft.com/ru/vs/community/> (дата обращения: 20.03.2021).
11. GitHub — google/googletest: Googletest — Google Testing and Mocking Framework; <https://github.com/google/googletest> (дата обращения: 20.03.2021).
12. Git; <https://git-scm.com/> (дата обращения: 20.03.2021).
13. MediaWiki; <https://www.mediawiki.org/wiki/MediaWiki> (дата обращения: 20.03.2021).
14. ГОСТ 19.001–77. Единая система программной документации (ЕСПД). Общие положения. — М.: Стандартинформ, 2010.
15. ГОСТ 19.301–79. Единая система программной документации (ЕСПД). Программа и методика испытаний. — М.: Стандартинформ, 2010.
16. ГОСТ Р 56920–2016/ISO/IEC/IEEE 29119-1:2013. Системная и программная инженерия. Тестирование программного обеспечения. Часть 1. Понятия и определения; <http://docs.cntd.ru/document/1200134996> (дата обращения: 20.03.2021).
17. ГОСТ Р 56921–2016/ISO/IEC/IEEE 29119-2:2013. Системная и программная инженерия. Тестирование программного обеспечения. Часть 2. Процессы тестирования; <http://docs.cntd.ru/document/1200134997> (дата обращения: 20.03.2021).
18. ГОСТ Р 56922–2016/ISO/IEC/IEEE 29119-3:2013. Системная и программная инженерия. Тестирование программного обеспечения. Часть 3. Документация тестирования; <http://docs.cntd.ru/document/1200134998> (дата обращения: 20.03.2021).
19. Малаион С.Ф., Куценко С.М. Оценка качества образования обучающихся // Приборостроение и автоматизированный электропривод в топливно-энергетическом комплексе и жилищно-коммунальном хозяйстве / VI Национальная науч.-практ. конф. — Казань, 2020. — Т. 1.

## Как подготовить статью для журнала «Компетентность»

Оригинал статьи и аннотацию к ней необходимо передать в редакцию в электронном виде (на магнитном носителе или по электронной почте [komp@asms.ru](mailto:komp@asms.ru)). При передаче информации по электронной почте желательно архивировать файлы. В названиях файлов необходимо использовать латинский алфавит. Допускаемые форматы текстовых файлов — TXT, RTF, DOC.

Допустимые форматы графических файлов:

- ▶ графики, диаграммы, схемы — AI 8-й версии (EPS, текст переведен в кривые);
- ▶ фотографии — TIFF, JPEG (RGB, CMYK) с разрешением 300 dpi.

К каждой статье необходимо приложить сведения об авторах — фамилия, имя, отчество, ученая степень, ученое звание, место работы и должность, телефон служебный и домашний, адрес электронной почты.

# Competencies Formation of Software Quality Management in Higher Education Institutions

R.M. Khamitov<sup>1</sup>, Kazan State Power Engineering University (KSPEU), Assoc. Prof. PhD, hamitov@gmail.com

N.K. Petrova<sup>1</sup>, KSPEU, Assoc. Prof. PhD, nk\_petrova@mail.ru

A.R. Nizamova<sup>2</sup>, KSPEU, alinarafaelevna\_kgeu@mail.ru

<sup>1</sup> Associate Professor, Kazan, Republic of Tatarstan, Russia

<sup>2</sup> Engineer of Department, Kazan, Republic of Tatarstan, Russia

**Citation:** Khamitov R.M., Petrova N.K., Nizamova A.R. Competencies Formation of Software Quality Management in Higher Education Institutions, *Kompetentnost' / Competency (Russia)*, 2021, no. 5, pp. 16–22. DOI: 10.24412/1993-8780-2021-5-16-22

## key words

information technologies, qualified personnel, motivation, new tools, high-quality work results

We considered the importance of an integrated approach to the development of technologies and testing tools by students of KSPEU in the framework of project activities for software development. The need for multilevel mastering of quality management competencies and software testing from simple to complex was also studied. We believe that testing and documenting the software development process are key aspects of creating quality software. The approach applied at KSPEU to the training of IT specialists in terms of the competence of quality management and software testing allows students to understand the need to develop high-quality software, knowledge of the means and methods for achieving this, acquire skills in testing tools, and collective responsibility for the work performed. Multilevel training in mastering the competencies of quality management and software testing allows graduates to acquire and consolidate the skills of creating a high-quality software product and writing tests.

## References

1. RF President Decree of 7.07.2011 N 899 On the Approval of Priority Areas for the Development of Science, Technology and Technology in the Russian Federation and the List of Critical Technologies of Russian Federation; <http://www.kremlin.ru/acts/bank/33514> (acc.: 20.03.2021).
2. Malatsion S.F., Kutsenko S.M. Povyshenie motivatsii obuchayushchikhsya pri izuchenii obshcheobrazovatel'nykh distsiplin v vysshey shkole [Increasing the motivation of students in the study of general disciplines in higher education], *Sovremennoe obrazovanie: sodержanie, tekhnologii, kachestvo*, 2018, vol. 1, pp. 211–214.
3. Safiullina L.Kh., Torkunova Yu.V., Ismagilov I.R., Krasilnikov O.V. Organizatsiya integratsionnogo testirovaniya v zadachakh sistemnogo programmirovaniya [Organization of integration testing in system programming problems], *Uchenye zapiski ISGZ*, 2018, no. 2, pp. 178–188.
4. Abdul'myanov T.R., Petrova N.K. Osnovnye etapy razrabotki i testirovaniya vysokotochnykh spetsializirovannykh integratorov [Main stages of development and testing of high-precision specialized integrators], *Vestnik KSPEU*, 2015, no. 2(26), pp. 26–36. ISSN 2072-6007.
5. Kulikov S. Software testing. Basic course. Chronicle of determinism; [http://svyatoslav.biz/software\\_testing\\_book/](http://svyatoslav.biz/software_testing_book/) (acc.: 20.03.2021).
6. NOU INTUIT Software Testing Basics. Information; <https://intuit.ru/studies/courses/48/48/info> (acc.: 20.03.2021).
7. Vishnevskaya T.I. Testirovanie programmnoy obespecheniya kak uchebnaya distsiplina [Software testing as an academic discipline], *Obrazovatel'nye resursy i tekhnologii*, 2014, no. 4, pp. 83–89.
8. Chikunov V.D., Lomash D.A. Novye podkhody v podgotovke IT spetsialistov v sovremennykh usloviyakh / [New approaches in the training of IT specialists in modern conditions], *Transport: nauka, obrazovanie, proizvodstvo: Mezhd. nauch.-prakt. konf.*, 2017, pp. 221–224.
9. Polevshchikov I.S. Osobennosti izucheniya sposoba testirovaniya bazovogo puti studentami bakalavriata v ramkakh distsipliny Testirovanie programmnoy obespecheniya (chast' 1) [Features of studying the method of testing the basic path by undergraduate students within the software testing discipline, part 1], *Molodoy uchenyy*, 2015, no. 18, pp. 10–12.
10. Visual Studio Community 2019 — Free IDE and Developer Tools; <https://visualstudio.microsoft.com/ru/vs/community/> (acc.: 20.03.2021).
11. GitHub — google/googletest: Googletest — Google Testing and Mocking Framework; <https://github.com/google/googletest> (acc.: 20.03.2021).
12. Git; <https://git-scm.com/> (acc.: 20.03.2021).
13. MediaWiki; <https://www.mediawiki.org/wiki/MediaWiki> (acc.: 20.03.2021).
14. GOST 19.001–77 Unified System of Program Documentation (USPD). General provisions, Moscow, *Standartinform*, 2010, 6 P.
15. GOST 19.301–79 Unified System of Program Documentation (USPD). Test program and methodology, Moscow, *Standartinform*, 2010, 6 P.
16. GOST R 56920–2016/ISO/IEC/IEEE 29119-1:2013 System and software engineering. Software testing. Part 1. Concepts and definitions; <http://docs.cntd.ru/document/1200134996> (acc.: 20.03.2021).
17. GOST R 56921–2016/ISO/IEC/IEEE 29119-2:2013 System and software engineering. Software testing. Part 2. Testing processes; <http://docs.cntd.ru/document/1200134997> (acc.: 20.03.2021).
18. GOST R 56922–2016/ISO/IEC/IEEE 29119-3:2013 System and software engineering. Software testing. Part 3. Testing documentation; <http://docs.cntd.ru/document/1200134998> (acc.: 20.03.2021).
19. Malatsion S.F., Kutsenko S.M. Otsenka kachestva obrazovaniya obuchayushchikhsya [Assessment of the quality of education of students], *Priborostroeniye i avtomatizirovannyy elektroprivod v toplivno-energeticheskom komplekse i zhillshchno-kommunal'nom khozyaystve: IV Nats. nauch.-prakt. konf.*, Kazan', 2020, vol. 1, pp. 180–182.