

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«ЮЖНО-УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ГУМАНИТАРНО-
ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ЮУрГГПУ»)
ПРОФЕССИОНАЛЬНО-ПЕДАГОГИЧЕСКИЙ ИНСТИТУТ
КАФЕДРА АВТОМОБИЛЬНОГО ТРАНСПОРТА, ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ И МЕТОДИКИ ОБУЧЕНИЯ ТЕХНИЧЕСКИМ
ДИСЦИПЛИНАМ

«РАЗРАБОТКА МНОГОПОЛЬЗОВАТЕЛЬСКИХ ТРЕНАЖЕРОВ В
УСЛОВИЯХ ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ
ОБРАЗОВАТЕЛЬНОЙ ОРГАНИЗАЦИИ»

Магистерская диссертация
по направлению 44.04.04 «Профессиональное обучение»,
программа магистратуры «Управление информационной безопасности
в профессиональном образовании»

Выполнил:

магистрант группы ЗФ-309-210-2-1

Гафаров Вадим Фаизович

Научный руководитель:

д.т.н., профессор кафедры

АТ, ИТиМОТД Белевитин Владимир

Анатольевич


Проверка на объем заимствований:

76,23 % авторского текста

Работа рекомендована к защите

«01» февраля 2019 г.

Зав. кафедрой АТ, ИТиМОТД

 В.В. Руднев

Челябинск, 2019

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
**«ЮЖНО-УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ
ГУМАНИТАРНО-ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ЮУрГГПУ»)**
Профессионально-педагогический институт
Кафедра автомобильного транспорта, информационных технологий
и методики обучения техническим дисциплинам

*Направление подготовки 44.04.04 – Профессиональное обучение
(Управление информационной безопасностью в профессиональном
образовании)*

З А Д А Н И Е

на выпускную квалификационную (магистерскую) работу

1. Студенту Гафарову Вадиму Фаизовичу, обучающемуся в группе ЗФ-309/210-2-1 по направлению подготовки 44.04.04 «Профессиональное обучение (управление информационной безопасностью в профессиональном образовании)»
2. Научный руководитель квалификационной работы: д.п.н., профессор кафедры профессор кафедры АТ,ИТиМОТД Белевитин Владимир Анатольевич.
Тема магистерской диссертации: «РАЗРАБОТКА МНОГОПОЛЬЗОВАТЕЛЬСКИХ ТРЕНАЖЕРОВ В УСЛОВИЯХ ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ ОБРАЗОВАТЕЛЬНОЙ ОРГАНИЗАЦИИ» утверждена приказом ректора Южно-Уральского государственного гуманитарно-педагогического университета № 580-сз от «26» апреля 2017 г
2. Срок сдачи магистрантом законченной работы на кафедру 18 февраля 2019 года
3. Содержание и объем работы (пояснительной расчетной и экспериментальной частей, т.е. перечень подлежащих разработке вопросов):
 - 1) Раскрыть специфику многопользовательских тренажеров как вида электронных образовательных ресурсов, дидактические возможности, функционал, опыт применения в профессиональном образовании;
 - 2) Провести обзор и анализ сред разработки многопользовательских тренажеров;
 - 3) Выявить структуру, содержание, этапы разработки многопользовательского тренажера для учебной дисциплины;
 - 4) Разработать проект многопользовательского тренажера для образовательной организации СПО – ГПБОУ «Южно-Уральский государственный технический колледж» (далее – «ЮУрГТК»);

5) Провести анализ защищенности информационных ресурсов в «ЮУрГТК» и выявить подходы к совершенствованию информационной безопасности образовательной организации;

6) Предложить меры информационной защиты для многопользовательского тренажера «ЮУрГТК» и проверить их эффективность.

4. Материалы для выполнения магистерской работы:

1) Учебная, нормативно-правовая, научно-техническая, педагогическая, методическая литература по теме магистерской работы.

2) Материалы преддипломной практики по теме магистерской работы.

5. Перечень графического материала (с точным указанием обязательных таблиц, чертежей или графиков, образцов и др.).

1) Таблицы.

2) Рисунки и диаграмм.

6. Консультанты по специальным разделам магистерской работы:

Раздел	Консультант	Отметка о выполнении
Педагогика		
Информационная безопасность		
Экономика		
Охрана труда		

Дата выдачи задания

«26» апреля 2017 года

Задание выдал _____

Белевитин В.А., профессор, д.п.н.

Подпись научного руководителя

Фамилия, Имя, Отчество, ученое звание и степень

Задание принял

Гафаров Вадим Фаизович

Подпись студента

Фамилия, Имя, Отчество студента

КАЛЕНДАРНЫЙ ПЛАН

№ п/п	Наименование этапов подготовки выпускной квалификационной (магистерской) работы	Срок выполнения этапов ВКР	Отметка о выполнении
1.	Предзащита ВКР		
2.	Доработка ВКР после предзащиты		
3.	Нормоконтроль		
4.	Подписание ВКР научным руководителем		
5.	Оформление пояснительной записки и презентации ВКР		
6.	Подписание рецензии на ВКР		
7.	Защита ВКР на заседании ГАК		

Автор ВКР Гафаров Вадим Фаизович

Фамилия, Имя, Отчество студента

Подпись студента

Научный руководитель ВКР

Белевитин В.А., профессор, д.п.н.

Фамилия, Имя, Отчество, ученое звание

Подпись научного руководителя

Заведующий кафедрой Руднев Валерий Валентинович, доцент, к.т.н.

Фамилия, Имя, Отчество, ученое звание

Подпись заведующего кафедрой

РАЗРАБОТКА МНОГОПОЛЬЗОВАТЕЛЬСКИХ ТРЕНАЖЕРОВ В УСЛОВИЯХ ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ ОБРАЗОВАТЕЛЬНОЙ ОРГАНИЗАЦИИ

Введение.....	5
ГЛАВА 1. ТЕОРЕТИКО-МЕТОДИЧЕСКИЕ ОСНОВЫ РАЗРАБОТКИ МНОГОПОЛЬЗОВАТЕЛЬСКИХ ТРЕНАЖЕРОВ В ОБРАЗОВАТЕЛЬНОЙ ОРГАНИЗАЦИИ.	
1.1. Многопользовательские тренажеры: специфика вида электронных образовательных ресурсов, дидактические возможности, функционал, опыт применения в профессиональном образовании.....	11
1.2. Обзор сред разработки многопользовательских тренажеров.....	19
1.3. Структура, содержание, этапы разработки многопользовательского тренажера для профессионального модуля.....	22
Выводы по 1 главе.....	37
ГЛАВА 2. ОБЕСПЕЧЕНИЕ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ ЭЛЕКТРОННЫХ ОБРАЗОВАТЕЛЬНЫХ РЕСУРСОВ В ГПБОУ «ЮУрГТК»	
2.1. Состояние защищенности электронных образовательных ресурсов в ГПБОУ «ЮУрГТК».....	38
2.2. Подходы к совершенствованию информационной защиты электронных образовательных ресурсов в ГПБОУ «ЮУрГТК».....	44
Выводы по 2 главе.....	50
ГЛАВА 3 РАЗРАБОТКА И АПРОБАЦИЯ МЕР ИНФОРМАЦИОННОЙ ЗАЩИТЫ МНОГОПОЛЬЗОВАТЕЛЬСКОГО ТРЕНАЖЕРА В ГПБОУ «ЮУрГТК»	
3.1. Реализация многопользовательского тренажера для профессионального модуля на базе Moodle.....	51
3.2. Аprobация многопользовательского тренажера на базе ЮУрГТК.....	70
3.3. Меры информационной защиты многопользовательского тренажера по профессиональному модулю в ГПБОУ ЮУрГТК.....	80
Выводы по 3 главе.....	98
Заключение	99
Библиографический список.....	101
Приложения.....	111

Введение

Обеспечение безопасности информационных систем образовательных организаций является актуальным требованием современного профессионального образования. Информационные ресурсы и информационные системы относятся к ряду основных защищаемых элементов во всех сферах жизнедеятельности современных организаций среднего профессионального образования.

Современной реальностью являются также и средства негативного информационного воздействия на элементы информационной образовательной системы. Создание и организация функционирования любых современных структур и систем, прежде всего, требует обеспечения их информационного взаимодействия с внешней средой, которое должно быть максимально надежным и безопасным.

Направление научных исследований в области информационной безопасности Российской Федерации «Исследование проблем создания и развитие защищенных информационно-телекоммуникационных систем» признано одним из приоритетных направлений научных исследований и одобрено в числе других по информационной безопасности Научного совета при Совете Безопасности Российской Федерации [34], [38].

Для образовательных организаций данное направление актуально также в связи с тем, что интенсивная информатизация образовательного процесса требует разработки и внедрения все новых интерактивных форм электронных средств. Актуальным становится использование интерактивных электронных тренажеров различной тематической направленности при профессиональной подготовке и при повышении квалификации с учетом индивидуальных особенностей обучающихся. В последнее время в образовательных организациях оптимальным и эффективным является разработка и внедрение многопользовательских электронных тренажеров, позволяющих устранять пробелы при изучении учебной дисциплины и

закреплять полученные знания, самостоятельно определять уровень подготовки и оценивать свои результаты; получать необходимые теоретические сведения и практические профессиональные навыки. Многопользовательские электронные тренажеры базируются на методике, содержащей в себе целенаправленную тренировку в процессе многократного повторного выполнения практико-ориентированных заданий. Другим несомненным достоинством электронного тренажера является минимум временных затрат в совокупности с принципом объективной оценки результатов деятельности обучающегося в процессе работы с тренажером. Обучающийся видит результаты своей деятельности, исключая субъективную оценку педагога. Применение данного дидактического средства базируется на теоретической и практической основе большого поколения педагогов и ученых.

В части педагогического обоснования мы опирались на труды отечественных ученых в области профессионального образования с применением информационных технологий, необходимо отметить таких ученых как: Е.С. Полат, И.В. Роберт, Е.К. Хеннера, В.Ф. Белов, Э.Ф. Зеер, А.Н. Леонтьев, А.М. Матюшкин и ряд других [36], [39], [40], [53], [15]. Методологической и научной основой проведенного исследования в области теории и практики безопасности сложных информационных систем, информационного взаимодействия, защиты технических, программных и информационных ресурсов являются труды отечественных ученых: Н.А. Кузнецова, Микрина Е.А., Кульбы В.В. В.А. Садовничий, К.В. Фролова, а также исследования таких известных ученых, как В.А. Герасименко, В.А. Конявский, А.А. Грушо, П. Д. Зегжда, Г.О. Крылов, А. Г. Сабанова, И.Б. Шубинского и ряд других [30], [42], [43], [55].

Исследователями в области информационной безопасности была сформирована теоретическая и практическая база для разработки теоретических положений и практического использования аппаратно-программных средств защиты информации, электронных образовательных

ресурсов и информационных технологий электронного документооборота. На практике, в большинстве образовательных организаций среднего профессионального образования (далее – СПО) преобладает комплексный подход к обеспечению безопасности образовательного процесса. При комплексном подходе защитные функции внедряются в информационную систему образовательной организации на этапе ее разработки и развертывания, и являются ее неотъемлемой частью. При этом защищается не только информация в форме сведений на носителе, процессы преобразования информации, но и разрабатываются и организуются виртуальные среды для создания дополнительной защиты в формате отграничения и частичной или полной изоляции информационных ресурсов. В практической деятельности по обеспечению режима информационной безопасности образовательные организации уделяют основное внимание выполнению требований и рекомендаций соответствующей российской нормативно-методической базы в области защиты информации.

Специфика образовательных организаций состоит в отсутствии стандартного подхода при проведении информатизации объектов профессионального образования, в связи с чем, каждая образовательная организация имеет уникальную корпоративную сеть, со своими особенностями, сложившимися пользовательскими традициями, разной степенью обеспеченностью квалифицированными кадрами, различными техническими характеристиками и архитектурными решениями. Обновление и апгрейд программно-аппаратного базиса корпоративной сети образовательной организации в связи с постоянно обновляющимися угрозами информационной безопасности должно происходить с учетом специфики такой сети и в соответствии с эффективными трендами обеспечения информационной безопасности, реализуемыми при комплексном подходе.

К наиболее эффективным направлениям относятся технологии виртуализации, в частности, создание так называемых «песочниц».

Песочница (англ. sandbox) – специальный механизм для безопасного исполнения программ, используется как часть проактивной защиты от вредоносного кода, от сетевых атак и является контролируемым набором ресурсов для выполнения гостевой программы.

Необходимость обеспечения нормативных требований и учета специфики программно-аппаратных систем защиты информационных ресурсов в организации профессионального образования при разработке и внедрении современных интерактивных обучающих средств, базирующихся на современных информационно-коммуникационных технологиях, определяют **актуальность** темы диссертации.

Целью диссертации является разработка и апробация многопользовательского тренажера в условиях обеспечения информационной безопасности образовательной организации (ГБПОУ «ЮУрГТК»).

Объектом исследования выступает образовательный процесс в образовательной организации среднего профессионального образования (СПО), а **предметом исследования** – процесс применения многопользовательского тренажера в образовательном процессе организации СПО.

Гипотеза исследования состоит в предположении о повышении защищенности информационных ресурсов образовательной организации при реализации комплексного подхода обеспечения безопасности образовательного процесса с применением технологий виртуализации посредством реализации изолированной среды (песочницы) и разработки двухфакторной аутентификации для доступа к ней.

Для достижения поставленной цели в работе решались следующие **задачи**:

- 1) Изучить научно-методические, технические информационные источники, на основе которых провести анализ содержания,

структуры, этапов разработки и опыта применения в педагогической практике многопользовательских тренажеров;

- 2) Провести анализ сред разработки многопользовательских тренажеров;
- 3) Проанализировать состояние защищенности электронных образовательных ресурсов в ГБПОУ «ЮУрГТК», выявить существующие подходы к совершенствованию информационной безопасности;
- 4) Разработать многопользовательский тренажер для профессионального модуля ГБПОУ «ЮУрГТК»;
- 5) Реализовать меры информационной защиты многопользовательского тренажера по профессиональному модулю в ГБПОУ «ЮУрГТК», проверить их эффективность.

Методологическую основу исследования составляют системный подход, метод моделирования, процессный подход, метод сравнения и аналогии, метод динамических испытаний и другие.

Научная новизна проведенных исследований и полученных в работе результатов заключается в следующем:

- показана возможность необходимого обновления существующей системы комплексной безопасности образовательного процесса в образовательной организации среднего профессионального образования путем реализации технологий виртуализации;
- уточнено понятие многопользовательского тренажера в контексте применения его в образовательном процессе СПО.

Практическая значимость работы заключается в следующем:

- разработан многопользовательский тренажер по профессиональному модулю в ГБПОУ «ЮУрГТК», который может быть применен и в других образовательных организациях СПО;

- реализована изолированная виртуальная среда (песочница) для применения многопользовательского тренажера;
- проведена апробация двухфакторной аутентификации для доступа к изолированной среде.

Проведенные исследования и полученные результаты могут быть использованы для создания, внедрения и управления комплексной системой защиты информационных ресурсов в образовательных организациях.

Ход исследования и его результаты докладывались и обсуждались на международных и всероссийских конференциях:

1. Всероссийская научно-практическая конференция «Национальная безопасность России: актуальные аспекты», Санкт-Петербург, 24-27 июля 2017 г. Публикация: «Подходы к обеспечению информационной безопасности электронных образовательных ресурсов. [Текст] Гафаров В.Ф., Гафарова Е.А.//Сб. «Материалы конференции ГНИИ «НАЦРАЗВИТИЕ», Санкт-Петербург, 2017, с.121-126
2. Международная научно-практическая конференция «Непрерывное образование в интересах устойчивого развития: новые вызовы», Казахстан, Қостанай, 6 декабря 2018 года. Публикация: «Обзор сред разработки многопользовательских тренажеров для системы профессионального образования». [Текст]/В.Ф. Гафаров.// Сборник материалов Международной научно-практической конференции «Непрерывное образование в интересах устойчивого развития: новые вызовы». Национальная академия образования им.И. Алтынсарина. – Астана, 2018 г.– с. 410-413.

База исследования: ГБПОУ СПО «Южно-Уральский государственный технический колледж», г. Челябинск.

Диссертационная работа состоит из введения, трех глав, заключения, библиографического списка и приложений.

ГЛАВА 1. ТЕОРЕТИКО-МЕТОДИЧЕСКИЕ ОСНОВЫ РАЗРАБОТКИ МНОГОПОЛЬЗОВАТЕЛЬСКИХ ТРЕНАЖЕРОВ В ОБРАЗОВАТЕЛЬНОЙ ОРГАНИЗАЦИИ

1.1. Многопользовательские тренажеры: специфика вида электронных образовательных ресурсов, дидактические возможности, функционал, опыт применения в профессиональном образовании

Образовательные технологии направлены на обеспечение включенности каждого обучающегося в учебно-познавательную деятельность. При этом для формирования нового понятия или отработки алгоритма обучающийся должен выполнить определенный набор действий. Реализации этой цели способствует использование на занятиях преподавателем всевозможных тренажеров. Нередко тренажерами называют оригинальную методику обучения, контроля и оценки знаний обучающихся, предлагающую обучающему набор заданий на заданную тему с контролем правильности их выполнения [14].

Как одна из методик обучения, электронный тренажер выполняет три основные взаимосвязанные функции: диагностическую, обучающую и воспитательную [19].

Диагностическая функция выявляет уровень знаний, умений, навыков обучающегося. Электронный тренажер помогает выявить и устранить пробелы в знаниях обучающегося, т.к. в основном тренажер представляет собой набор тестовых заданий, то по объективности, широте и скорости диагностирования он превосходит все остальные формы педагогического контроля.

Обучающая функция тренажера проявляется в активизации работы обучающегося по усвоению учебного материала. Так, многие электронные тренажеры содержат наводящие вопросы и подсказки; после прохождения задания предоставляются ссылки на разделы учебного материала или

вопросы, по которым обучающийся ответил неверно; существует возможность повторного решения задания, решения однотипной группы заданий или задания определенного уровня сложности.

Воспитательная функция проявляется в дисциплинированности и самоорганизации деятельности обучающихся; в формировании стремлений развить способности, инициативность, самостоятельность и ответственность у обучающегося.

Актуальным становится использование интерактивных электронных тренажеров по разным темам учебных предметов, с учетом индивидуальных особенностей обучающихся. Тренажеры применимы на занятиях, когда важно не только систематизировать изученный материал, но и акцентировать внимание обучающихся на основных моментах изучаемой темы, необходимой для дальнейшего восприятия темы или подготовки к зачетной работе. Кроме того, визуальный ряд, используемые для создания электронного тренажера, дополняет рисунки учебника, что повышает наглядность занятий. Это делает электронный тренажер, как пособие, незаменимым не только при объяснении нового материала, но и помогает понять сложный учебный материал в случае самостоятельного его изучения обучающимся. Тренажеры можно использовать для фронтальной или индивидуальной работы обучающихся, для самостоятельной работы вне аудитории, для ликвидации пробелов в обучении, отработки навыков решения задач или теоретических основ изученной темы.

Самостоятельная работа обучающихся с электронными тренажерами повышает их мотивацию к обучению и активность в процессе обучения, позволяет обучающемуся работать в индивидуальном, комфортном для него темпе, тем самым снимает психологическое напряжение, а игровая основа вносит положительную эмоциональную окраску в занятие. Для обучающегося такая работа создает ситуацию успеха, а преподаватель ненавязчиво достигает своей цели через заинтересованность и

мотивированность обучающегося, освоить обязательный минимум по предмету и довести до автоматизма определенные навыки.

Таким образом, одним из основных достоинств тренажеров является целенаправленная тренировка в процессе многократного повторного решения заданий. Другим несомненным достоинством электронного тренажера является минимум временных затрат в совокупности с принципом объективной оценки результатов деятельности обучающегося в процессе работы с тренажером. Обучающийся видит результаты своей деятельности, исключая субъективную оценку педагога.

Итак, электронный тренажер позволяет обучающемуся:

- устранять пробелы при изучении учебной дисциплины и закреплять полученные знания;
- самостоятельно подготавливаться к зачетной работе (контрольной работе, тестированию и т.п.);
- самостоятельно определять уровень подготовки и оценивать свои результаты;
- получать необходимые теоретические сведения, практические примеры и разъяснения к каждому тестовому заданию в процессе работы с тренажером;
- подготовиться к сдаче государственных экзаменов по предмету, используя варианты тренажеров, структура которых соответствует содержанию государственных образовательных стандартов.

Реализация методических функций в компьютерных тренажерах выглядит следующим образом [20] :

- справочно-информационные: конспект самой полезной информации по теме обучения доступен в справочной документации в тренажере;
- контролирующие: тестирование и отчет о проделанном технологическом процессе позволяют преподавателю оценить степень усвоения материала и принять решение о возможности допуска к реальному производственному процессу, есть возможность проводить технологический

процесс до тех пор, пока не получится продукт или полупродукт заданного качества;

- имитационные: построенная в тренажере модель описывает процессы так, как они проходили бы в действительности, порядок технологических операций в тренажере соответствует реальному процессу;

- моделирующие: математическая модель в основе тренажера позволяет рассчитывать точный результат технологического процесса на основе заданных параметров и действий пользователя;

- демонстрационный: тренажер может использоваться для проведения демонстрационных экзаменов по дисциплине.

Справочно-информационная функция позволяет представить конспект самой полезной информации по теме обучения доступен в справочной документации в тренажере.

Контролирующая функция позволяют преподавателю оценить степень усвоения материала и принять решение о возможности допуска к реальному производственному процессу через тестирование и отчет о проделанном технологическом процессе, также есть возможность имитировать технологический процесс до тех пор, пока не получится продукт или полупродукт заданного качества, либо пока не будут отработаны навыки необходимой регламентации и заданных технических параметров.

Имитационная функция реализуется благодаря построенной в тренажере модели, порядок технологических операций в тренажере соответствует реальному процессу.

Благодаря диагностической функции выявляется уровень знаний, умений, навыков обучающего, при этом выявляются пробелы в знаниях обучающего, по объективности, широте и скорости диагностирования тренажеров превосходит все остальные формы педагогического контроля.

Обучающая функция проявляется в активизации работы обучающего по усвоению учебного материала. Так, многие электронные тренажеры содержат наводящие вопросы и подсказки; после прохождения задания

предоставляются ссылки на разделы учебного материала или вопросы, по которым обучающийся ответил неверно; существует возможность повторного решения задания, решения однотипной группы заданий или задания определенного уровня сложности.

Воспитательная функция проявляется в дисциплинированности и самоорганизации деятельности обучающихся; в формировании стремлений развить способности, инициативность, самостоятельность и ответственность у обучающегося.

Актуальным становится использование электронных тренажеров по разным темам учебных предметов, с учетом индивидуальных особенностей обучающихся.

Учитывая такой большой дидактический ресурс электронных тренажеров, закономерно встает вопрос о возможности введении многопользовательского режима, при котором использование данного средства могут проходить от двух и более лиц одновременно, что существенно расширяет возможности организации образовательного процесса.

На основе анализа изученных источников [19], [20], [23], [24] можно дать следующее определение многопользовательского тренажера (далее – МТ) в рамках настоящего исследования: *многопользовательский тренажер* - это программно-методический комплекс, в основу которого положена система диагностики и интерпретации полученных ответов, алгоритмы целенаправленной тренировки обучающегося в процессе многократного повторного выполнения имитации технологического процесса, а также тестовых заданий различного формата.

Образовательная практика имеет положительный опыт применения МТ в профессиональном обучении [37], [49], [51], [52],[56].

Имеется успешный опыт применения МТ в самых разных направлениях профессиональной подготовки. Рассмотрим некоторые

наиболее успешные проекты реализации многопользовательских электронных тренажеров.

Совместно с компанией ООО "Газпром трансгаз Екатеринбург" был реализован многопользовательский сетевой тренажер для обучения персонала автомобильных газонаполнительных компрессорных станций (далее - АГНКС) [56].

Учебно-методическое пособие – компьютерный тренажер-имитатор "АГНКС" предназначен для подготовки и поддержания необходимого уровня квалификации руководителей, специалистов и рабочих занимающихся эксплуатацией объекта, проходящих обучение в Учебно-производственном центре – филиала ООО "Газпром трансгаз Екатеринбург". Эффективность МТ – компьютерного тренажера-имитатора "АГНКС"- заключается в существенном повышении качества обучения персонала, а так же в снижении рисков возникновения нештатных ситуаций, связанных с человеческим фактором (профессиональной компетентностью персонала).

МТ содержит учебные сценарии: технологическая схема АГНКС: входной газопровод, компрессорная установка, блок осушки, блок аккумуляторов газа; технологические операции: прием смены, регламентные работы в течение смены на АГНКС, пуск АГНКС, блок осушки газа, регламентные работы, безопасная заправка; план ликвидации аварийной ситуации: прекращение подачи электроэнергии; прекращение подачи газа на входе; утечка газа на входном газопроводе, загазованность в компрессорном отделении, пожар в техническом блок-боксе, утечка газа на коллекторе, разрыв рукава высокого давления при заправке, утечка антифриза из системы охлаждения, возникновение угрозы террористического акта.

Компьютерное воспроизведение виртуальной реальности находит применение в современных авиационных тренажерах. Применение распределенных систем виртуальной реальности для построения проекционных многоканальных систем визуализации, состоящих из множества компонент, взаимодействующих между собой по сети с целью

создания единой для большого числа пользователей виртуальной среды. Главным условием обеспечения согласованности распределенных данных является одновременное поддержание заданной чувствительности и времени реакции системы [41].

В [56] описан МТ для обучения сотрудников предприятий нефтегазовой отрасли правильному и безопасному обслуживанию трубчатых печей. В разработанном МТ имеется множество сценариев, применимых для производственных ситуаций: запуск печи, остановка печи, поддержание рабочего режима печи, обнаружение и устранение неисправности в работе печи.

Кроме того, в МТ встроен симулятор аварийных ситуаций для проверки у сотрудника всех его знаний и навыков, полученных в ходе выполнения обучающих программ. Симулятор, в отличие от обучающих программ, имеет особенности: отсутствие каких-либо подсказок на графическом интерфейсе; аварийная ситуация случайным образом генерируется при каждом запуске симулятора; нет права на ошибку.

Симулятор позволяет проверить и отточить свое мастерство, имитируя работу реального объекта. Сотрудник, который пользуется симулятором, должен сам обнаружить и устранить причину неполадки. Каждый раз неисправности генерируются случайным образом и могут появиться спустя некоторое время после запуска симулятора. Цель такого тренажера заключается в том, чтобы наработать определенный оперативный опыт у специалиста для последующей рациональной оценки ситуации, правильные и своевременные действия по её устранению. В случае ошибки со стороны оператора симулятор выводит информацию о текущей сессии. В ней указывается допущенная оператором ошибка, неисправность в запущенной сессии, комментарии по её устранению.

МТ нашли свое применение не только в политехнических отраслях знаний, но и в гуманитарных. Так, в [35] приводится описание компьютерного тренажера, предназначенного для освоения идеологии

рыночной экономики, знакомства с основами финансовой грамотности и инвестиционными механизмами. Тренажер ориентирован на неэкономистов, может работать как в однопользовательском, так и в многопользовательском режиме. Толчком для представляемой работы послужили результаты маркетинговых исследований, в которых отражены те обстоятельства, что большая часть населения РФ до сих пор не понимает идеологии рыночной экономики и не владеют даже основами финансовой грамотности; обучение основам экономики в том или ином виде ведется в большинстве ВУЗов, а также тот факт, что в настоящее время в учебных планах ВУЗов сокращается число аудиторных занятий и растет доля самостоятельной работы студентов, для чего нужны специальные инструментальные средства, в том числе компьютерные тренажеры.

Тренажер содержит симуляции по проведению операций покупки-продажи бизнесов, ценных бумаг, участков земли, драгметаллов, также были добавлены вычисление синергетического эффекта группы бизнесов, возможность развития предприятий путем повышения специфических коэффициентов - коэффициентов развития НИР, маркетинга и др., предложена гибкая смена ситуации на рынке, генерация образовательных программ для развития предприятий. Тренажер работает в однопользовательском и многопользовательском режимах. Для многопользовательского режима были введены стандартные для такой архитектуры компоненты – база данных и сервер базы данных. Взаимодействие между всеми компонентами тренажера осуществляется с использованием современных технологий Microsoft NetRemoting. NET представляет мощную и в то же время простую модель программирования и поддержку периода выполнения, благодаря чему удаленное взаимодействие становится прозрачными.

1.2. Обзор сред разработки многопользовательских тренажеров

Образовательная практика реализует МТ посредством онлайн-платформ для организации дистанционного обучения, то есть названные системы традиционно являются средами разработки для МТ.

Рассмотрим основные системы, пригодные для создания МТ.

На сегодняшний день Moodle [58] – это, несомненно, одна из самых популярных систем дистанционного образования с открытым исходным кодом. Moodle предлагает пользователю различные панели инструментов, возможность отслеживать прогресс обучающихся и поддержку мультимедиа, позволяет создавать курсы, адаптированные для мобильных приложений, имеет функциональные возможности для интеграции дополнений от сторонних разработчиков. Кроме того, Moodle имеет набор готовых шаблонов курсов, что дает для преподавателя максимальное количество свобод в проектировании образовательного процесса, в частности для разработки МТ. Заслуживает внимания реализуемая в Moodle технология проведения активного обучения и взаимодействия, которая обеспечивает информационно-коммуникационное взаимодействие через сервисы «Форум», «Семинар», «Анкетный опрос», «Сообщение», «Wiki», «Глоссарий». Единственным, на наш взгляд, недостатком является платность данной платформы. Дидактические возможности применения системы дистанционного обучения Moodle для организации самостоятельной работы обучающихся представлены через возможность использования элементов онлайн-курса, таких как «Лекция», «Тест», «Семинар», «Задание» на различных этапах освоения учебного материала. Единственным, на наш взгляд, недостатком является платность данной платформы. Однако, для образовательных организаций, к которым относятся и организации СПО, действует специальный тариф, что позволяет его полноценно использовать без значительных финансовых затрат.

Ё-Стади, как заявлено разработчиками [57], это полноценная электронная образовательная среда. Она поддерживает SSL шифрование для безопасной передачи данных, используется HTTPS протокол для обеспечения персональных данных.

Ё-Стади - бесплатная российская разработка команды единомышленников по развитию дистанционного образования. Отличие от других подобных платформ состоит в том, что функционал ориентирован на практическую работу. Ё-Стади, безусловно, позволяет публиковать учебные материалы, но большая часть системы предназначена для всевозможной оценки знаний и тестирования, то есть ни о какой имитации технологического процесса здесь не может идти речь, тем более, что отсутствует возможность самостоятельной доработки и нет поддержки SCORM. В целом Ё-Стади заслуживает отличной оценки и является хорошим решением для небольших компаний, желающих организовать обучение персонала без каких-либо затрат на приобретение систем дистанционного обучения, либо создания МТ.

Origno [60] - это платформа электронного обучения с открытым исходным кодом на основе Drupal, которая позволяет управлять онлайн-тренингами и эффективно следить за тем, чтобы навыки обучающихся поддерживались на заданном уровне. Origno предназначена для компаний, корпораций и университетов, дает гибкое решение для электронного обучения и легко масштабируется.

Возможности, предоставляемые системой Origno, не могут не радовать педагогов-практиков: сертификаты, расписание занятий, форумы, авторские инструменты электронного обучения, система оценок и видео галереи – это лишь немного из внушительного списка функций, доступных пользователю и пользователю - разработчику.

Все перечисленное дает возможность управлять учебными программами, отслеживать динамику обучения у слушателей и даже интегрировать электронную коммерцию, используя всего этот один

инструмент. Также Origno предлагает пользователю онлайн-опросы, возможность пересылки мгновенных сообщений и чат, что дает возможность для быстрого предоставления и получения обратной связи и эффективного сотрудничества. Преимущества Origno нивелируются существенным недостатком – она платная и стоимость данной платформы высока, при этом никаких преференций для образовательных организаций не предусмотрено.

Инструменты оценки для электронного обучения, социальная интеграция и домашняя страница обучающегося – лишь несколько из многих преимуществ OLAT [59]. В этой системе можно установить расписание, email-уведомления, возможность добавления закладок, файловое хранилище и сертификаты. С помощью OLAT можно легко и быстро добавить новых пользователей в систему, а также разрабатывать комплексные курсы электронного обучения. Еще одна интересная функция – это возможность проверки совместимости с браузерами. Нажатием всего нескольких кнопок вы сможете убедиться, что учебный материал корректно отображается во всех браузерах. OLAT идеально подходит для мультиплатформенных курсов электронного обучения, предназначенных для различных устройств.

Однако, и OLAT, и Origno несмотря на высокий функционал, скорее всего, будут доступны только для крупных корпораций из-за высокой стоимости, тогда так для самостоятельной разработки МТ преподаватели среднего и высшего профессионального образования будут вынуждены работать с Ё-Стади, либо в Moodle, если они уже установлены в образовательной организации.

1.3. Порядок разработки, структура и содержание многопользовательского тренажера для профессионального модуля

Существенным отличием электронного тренажера от электронного учебного пособия является наличие сценария.

Под сценарием (scripts) [22] понимается некоторая предопределенная последовательность команд, способных выполняться в автоматическом режиме. Сценарий тренажера - это пок кадровое распределение содержания учебного курса и его процессуальной части в рамках программных структур разного уровня и назначения. В общем случае сценарий представляет собой два взаимосвязанных руководства по реализации конкретного проекта:

- педагогический сценарий;
- технологический сценарий.

Педагогический сценарий отражает авторское представление о содержательной стороне курса или практической работы, о структуре материала, предоставляемого обучаемому, порядку и условиям выдачи информации на экран монитора. Планирование педагогического сценария предполагает четкое видение автором образовательной цели и содержания конкретной учебной дисциплины, его умение определить педагогические технологии в соответствии с особенностями целевых учебных групп, проектирование содержания учебной деятельности. Для решения этих задач на этапе проектирования преподаватель должен четко определить порядок изучения учебного материала.

Подготовив все необходимые компоненты педагогического сценария, преподаватель должен определить наиболее эффективные траектории изучения курса с учетом индивидуальных особенностей восприятия материала в зависимости от образовательного уровня обучающихся и успешности или неуспешности их действий на каждом этапе работы с электронным тренажером.

Подобранная автором первичная учебная информация, предоставленная в электронном виде, на этапе проектирования тренажера должна быть скомпонована в соответствии с идеями автора в интерактивные учебные кадры так, чтобы, с одной стороны, обучаемый имел возможность сам выбирать темп и в определенных пределах последовательность изучения материала, а, с другой стороны, чтобы процесс обучения оставался управляемым. Этот этап -

построение детального технологического сценария курса - является наиболее ответственным, т.к. именно он позволяет найти оптимальное соединение педагогических задач и наиболее целесообразных для них технологических решений.

Технологический сценарий [22] - это описание информационных технологий, используемых для реализации педагогического сценария. В технологическом сценарии, как и в педагогическом, также реализуется авторский взгляд на содержание и структуру курса, его методические принципы и приемы его организации, с учетом технологических средств используемых для создания электронного тренажера. Авторское представление о курсе отражает и пользовательский интерфейс - визуальное представление материала и приемы организации доступа к информации разного уровня. В сценарии необходимо выстроить материал по уровням, а также указать:

- какие компоненты мультимедиа курса будут разработаны для наиболее эффективного обучения;
- характер доступа к ним;
- авторские пожелания по дизайну;
- ключевые слова и средства навигации по материалу;
- необходимые мультимедиа приложения.

Разработчик в составлении технологического сценария обеспечивает качественное решение педагогических задач, соединение в едином мультимедиа курсе педагогических и информационных образовательных технологий.

Приступая к созданию технологического сценария электронного тренажера следует учитывать, что вся учебная информация, благодаря гипертекстовой технологии, распределяется на нескольких содержательных уровнях. Смысловые отношения между уровнями могут быть выстроены различными способами.

Эргономичность, удобство навигации электронного тренажера определяется пользовательским интерфейсом, который должен обладать в самом общем случае следующими основными свойствами:

- функциональностью;
- привлекательным дизайном;
- уникальностью и запоминаемостью;
- гибкостью.

При реализации конкретной информационной системы и ее интерфейса юзабилити следует рассматривать как совокупность факторов, влияющих на работу обучаемого с конкретным тренажером, включая:

- легкость обучения – как быстро обучаемый может изучить интерфейс, чтобы решать образовательные задачи;
- эффективность – как быстро может решать свои задачи обучаемый, имеющий опыт работы с системой;
- запоминаемость – может ли обучаемый после перерыва в работе быстро вспомнить методы работы с тренажером;
- частота и серьезность ошибок – как часто делают ошибки пользователи при работе с тренажером и какова серьезность этих ошибок;
- субъективное удовлетворение – нравится ли обучаемым работать с тренажером.

Перечисленные выше факторы, в значительной степени оцениваются субъективно и такая оценка зависит от содержательной части – количества и качества иллюстративного материала, от организации интерфейса ЭТ, в целом – удобства и понятности органов управления, цветовой гаммы и ряда других факторов.

Создание пользовательского интерфейса должно быть основано на понимании ЭТ как средства комплексного воздействия на обучаемого путем сочетания концептуальной, информационной, иллюстративной, справочной, тренажерной и контролирующей частей.

Структура и пользовательский интерфейс этих частей курса должны обеспечить эффективную помощь при освоении учебного материала. Интерфейс ЭТ может иметь свои особенности в зависимости от предметной области. Кроме традиционных для гипертекстовых систем методов ключевых слов, активация которых вызывает либо переход к другому документу, либо вывод краткого "всплывающего" (pop-up) текста - комментария, инструментальные средства позволяют создавать и другие активные элементы - командные кнопки, снабженные надписями или пиктограммами, надписи и изображения, реагирующие на щелчок или перемещение мыши, кнопки-переключатели и многое другое. Знание автором возможных интерфейсных решений позволяет ему при разработке педагогического и технологического сценариев наиболее эффективно структурировать учебную информацию и максимально задействовать все каналы восприятия этой информации.

Подготовка тестовых заданий (далее - ТЗ) для компьютерного контроля знаний - это сложный и кропотливый процесс. Проблема создания качественных тестовых заданий важна для образовательных учреждений любого уровня общего, среднего и высшего профессионального образования. Исследования ряда ученых [18],[20],[35] показывают, что до 50 % тестовых заданий, составляемых преподавателями без экспериментальной проверки, не пригодны для определения уровня знаний обучаемых.

Тестирование как одна из форм аттестации представляет собой процедуру, позволяющую объективно для каждого обучающегося установить уровень: теоретических знаний, интеллектуальных умений, практических навыков. Тестирование используется на следующих этапах учебного процесса:

1. Входное тестирование в начале изучения дисциплины для определения исходного уровня и выдачи рекомендаций по траектории изучения материала.

2.Самотестирование учащихся по ходу изучения ими отдельных глав теоретического материала.

3.Тестирование перед началом выполнения лабораторных и практических работ с целью проверки наличия необходимых начальных теоретических знаний.

4.Тестирование знаний по итогам изучения главы (раздела) теоретического материала, результатам выполнения лабораторной работы.

5.Тестирование по итогам изучения дисциплины с целью получения оценки полученных знаний либо выдачи допуска на прохождение промежуточной аттестации в иной форме.

В общем случае тест – это инструмент, состоящий из квалиметрически выверенной системы тестовых заданий, стандартизированной системы проведения и заранее спроектированной технологии обработки и анализа результатов, предназначенный для измерения качеств, знаний или навыков личности, изменение которых возможно в процессе систематического обучения.

Тесты включают в себя вопросы, отражающие содержание дисциплины или ее части, которые выносятся на контроль. При составлении вопросов для тестирования необходимо придерживаться следующих правил:

- в вопросе должна быть ясно выражена только одна мысль;
- мысль, выраженная в вопросе, должна быть записана лаконично, но содержательно;
- вопрос должен представлять важную часть пройденной темы;
- вопрос по сложности должен быть доступен студенту, а по содержанию – соответствовать критериям будущей профессиональной деятельности студента или потребностям обучения по другим дисциплинам;
- при формулировании вопросов и ответов следует исключать подсказки к правильным ответам;

- задания в тесте следует располагать в порядке постепенного возрастания трудности, что способствует снижению эмоционального стресса в процессе тестирования.

Для получения максимальной эффективности от тестирования знаний в процессе изучения дисциплины рекомендуется использовать два вида тестов:

- тест для самоконтроля (для каждого раздела или темы),
- итоговый тест (для всей дисциплины в целом).

Вопросы для самоконтроля предполагают возможность просмотреть теоретический материал и проработать ошибки, допущенные при ответах на данные вопросы. Они предназначены для получения обучающимся адекватной оценки своих знаний. Для каждого раздела рекомендуется 10-15 вопросов.

Работа с тестовой системой начинается с подготовки базы вопросов. При использовании тестирования в учебном процессе важно помнить, что каждый вопрос не должен иметь многоцелевую направленность, он призван выявлять лишь один определенный аспект. Критериями отбора содержания тестового материала служат:

- значимость. Этот принцип указывает на необходимость включить в тест только те элементы знания, которые можно отнести к наиболее важным и ключевым, без которых знания становятся неполными;
- научная достоверность. Суть тестовых заданий заключается в том, что они требуют четкого, заранее известного преподавателям ответа. Все спорные точки зрения, вполне допустимые в науке, не рекомендуется включать в тестовые задания;
- соответствие содержания теста уровню современного состояния науки. Этот принцип базируется на естественной необходимости подготовки специалистов и проверки их знаний на современном материале;
- репрезентативность. Следует обращать внимание не только на включение значимых элементов, но и на полноту охвата пройденного материала;

- возрастающая трудность учебного материала. Знание последующих элементов курса зависит от знания предыдущих учебных элементов, то есть дисциплину можно изучать только с самого начала и без пробелов;
- вариативность содержания. По мере изменения содержания учебной дисциплины должно варьироваться и содержание теста;
- системность содержания. Это означает, что подбор содержания и количества тестовых заданий должен отражать все разделы теоретических материалов соответственно трудоемкости по рабочей программе;
- комплектность и сбалансированность содержания теста. Тест, разработанный для итогового контроля знаний, не может состоять из материалов только одной темы;
- взаимосвязь содержания и формы. Содержание теста должно быть выражено в наилучшей, с точки зрения наглядности и обучающего потенциала, форме.

Существует множество видов (форматов) тестовых заданий, однако на практике используется от четырех до шести основных видов, которые делятся на два основных типа: закрытого и открытого. Вопросы закрытого типа предлагают выбрать ответ (один или несколько) из многих предложенных. При этом подразумевается, что все предложенные варианты ответа являются равнопривлекательными. Задания открытой формы не предлагают вариантов ответа, а требуют ввода символов в пустое поле какого-то утверждения, причем предполагается, что заполнить этот пропуск можно строго однозначно. К открытому типу относятся вопросы, предполагающие развернутый ответ на естественном языке. Данный вид вопросов является единственным, не допускающим возможность проверки ответа компьютером, а требует проверки и оценки ответа преподавателем.

Рассмотрим основные виды тестовых заданий, схематически классификация представлена на рис. 1

1. Выбор единственного ответа. Это простейшим вид тестовых заданий закрытого типа и представляет собой вопрос с множеством предложенных ответов, из которых требуется выбрать один верный. Разновидностью данного вида заданий являются вопросы с активной областью, в которых выбор ответа с помощью кнопок заменен выбором места на графическом изображении.



Рисунок 1.

Преимущества данного вида вопросов:

- данный вид заданий интуитивно понятен обучающимся,
- ввод ответа требует минимального времени,
- процедура обработки ответа предельно проста.

Недостатки простой выборки:

- существенная вероятность угадывания правильного ответа,
- возможность запоминания неверных ответов.

2. Множественный выбор. Второй из наиболее распространенных типов вопросов.

Представляет собой вопрос с множеством предложенных ответов, из которых требуется выбрать несколько верных. Кнопки выбора в нем заменены на окошки метки и обеспечивают возможность выбора

произвольной комбинации ответов (от одного ответа до всех возможных вариантов).

Преимущества данного вида вопросов:

- этот тип заданий информативен,
- дает возможность учесть частично правильные ответы.

Недостатки множественной выбора:

- существенная вероятность угадывания правильного ответа,
- возможность запоминания неверных ответов,
- отсутствие общепризнанной процедуры обработки ответа.

3. Установление соответствия. Представляет собой два списка в виде двух колонок и обучающийся должен сопоставить данные из разных колонок друг другу.

Преимущества данного вида вопросов:

- вероятность угадывания минимальна;
- можно подобрать вопросы достаточно сложные по содержанию, требующие усвоения знаний на уровнях анализа и синтеза.

Недостаток таких вопросов - это сложность выполнения теста при достаточно большом списке для сопоставления.

4. Установление последовательности. В таком типе вопросов обучающемуся задается вопрос и дается набор готовых элементов. В его задачу входит расстановка этих элементов в правильной последовательности.

Преимущество данного вида вопросов – это вероятность угадывания (при числе элементов более трех) - незначительна.

Недостаток такого вида вопросов – это не всегда возможно подобрать только один вариант правильного алгоритма.

5. Ответ на естественном языке. Данный вид близок к традиционной форме контроля тип заданий.

Преимущества данного вида вопросов:

- вероятность угадывания минимальна;

- методически ценно то, что реализуется самостоятельная формулировка ответа.

Недостатки свободного ввода:

- сложность синтаксического (тем более - семантического) анализа ответа;
- невозможность автоматического контроля ответов;
- наличие субъективного фактора в оценке ответов.

6. Заполнение пропуска. Иногда этот вид называют "Ввод символа", когда вводу подлежит один символ – буква или цифра. Данный вид задания представляет собой фразу или выражение, в котором пропущено слово или дата. Предполагается единственно возможный вариант ввода правильного ответа. Данный вид заданий наиболее эффективен при проверке разного рода терминов, констант, дат и правописания.

Преимущества:

- вероятность угадывания минимальна;
- данный вид заданий интуитивно понятен обучающимся.

Недостатки:

- возможность запоминания неверных ответов;
- невозможность в ряде случаев предусмотреть ввод учащимся различных синонимов.

Все вышеперечисленные типы возможно реализовать в многопользовательском тренажере.

Для формирования наибольшей наглядности тестового контроля рекомендуется максимально использовать графический материал. По завершении составления тестовых вопросов автору (авторам) необходимо указать следующие обязательные параметры:

- название теста (для какой дисциплины он предназначен) возможно с датой разработки или номером версии;
- общее число тестовых вопросов и рекомендуемое число вопросов теста, выдаваемых одному студенту;

- шкалу оценки знаний и соответствующее ей процентное содержание правильных ответов.
- время, отведенное на выполнение теста, но не более 45 – 90 минут (среднее время до момента утомления).
- иногда, помимо общего времени на выполнение теста, указывается и максимальное время для ответа на один вопрос.
- порядок прохождения теста – с возможностью пропуска тестовых вопросов и последующего возврата к ним, или без таковой.
- ключи к тестовым вопросам, т.е. правильные ответы к тестам, которые необходимы для создания электронной системы тестирования.

При подборе материалов и составлении ТЗ следует придерживаться следующих правил:

- содержание ТЗ должно быть ориентировано на проверку значимых понятий и элементов содержания конкретного учебного курса;
- основные термины тестового задания должны быть ясно определены;
- тестовые задания должны быть сформулированы в виде кратких суждений и четко поставленных вопросов;
- следует избегать контрольных заданий, которые требуют от испытуемых развернутых заключений при выполнении контрольных заданий;
- при разработке ТЗ можно применять графические и мультимедийные компоненты не только с целью представления содержания учебного материала, но и при постановке контрольного задания, требующего графической формы ответа;
- содержание задания должно быть выражено предельно простой синтаксической конструкцией без повторов и двойных отрицаний;
- в тексте задания не должно быть непреднамеренных подсказок и сленга;
- не следует включать в текст ТЗ прямые цитаты из книг;

- недопустимы заключения типа: все выше перечисленное верно, все указанные ответы неверны и т.д.;
- в ТЗ не должна использоваться терминология, выходящая за рамки учебной дисциплины;
- задание должно быть составлено с учетом того, что среднее время формирования заключения тестируемого со средним уровнем подготовки не должна превышать 2-х минут.
- количество ТЗ для начала нормальной работы желательно иметь порядка 200-300 заданий по конкретному учебному курсу, т.е. в расчете на ЭОР по данному курсу;
- при подборе материала для ТЗ следует учитывать, что все задание должно размещаться на экране без прокрутки;
- при разработке задания не следует делать ответ регистрозависимым.

Среднее время ответа студента на ТЗ определяется установкой преподавателя – автора контрольного задания. Оно выполнения задания определяется эмпирически при апробации теста. Кроме того, при тестировании часто вводят временное ограничение на тест в целом, т.е. время в течение которого обучаемый при среднем уровне подготовки должен ответить на все ТЗ.

Важным этапом подготовки ТЗ является их экспериментальная проверка – апробация. Эту процедуру следует начинать с тщательного анализа формулировок ТЗ в соответствии с изложенными выше правилами и выстраивание ТЗ в порядке предполагаемой трудности. В процессе апробации необходимо предусмотреть возможность пропуска сложных ТЗ, чтобы обратиться к ним после окончания работы над другими ТЗ.

На этом этапе желательно иметь максимальный набор различных тестовых заданий. Задания различного типа уменьшают утомляемость и позволяют включить в процедуру тестирования большего числа ТЗ. Задания различного типа более объективны для контроля знаний и для различных

разделов ЭОР могут быть полезны различные типы заданий. В то же время, если тестирование имеет локальные цели, следует включать в тест задания одного типа.

Результаты апробации порой приводят к отбраковке до 40 –60 % ТЗ, поэтому банк заданий должен быть достаточно большим. Целями апробации тестовых заданий являются определение трудности задания и оценка их пригодности; определение заданий, которые имеют существенные недостатки; определение временных параметров выполнения заданий; анализ норм тестовых заданий; выявление случайных ошибок и неточностей.

В общем виде качественную картину теста в целом поясняет рис.2 где в графическом виде отражены понятия простого теста (1), нормального (2), сложного (3) и неудачного (4).

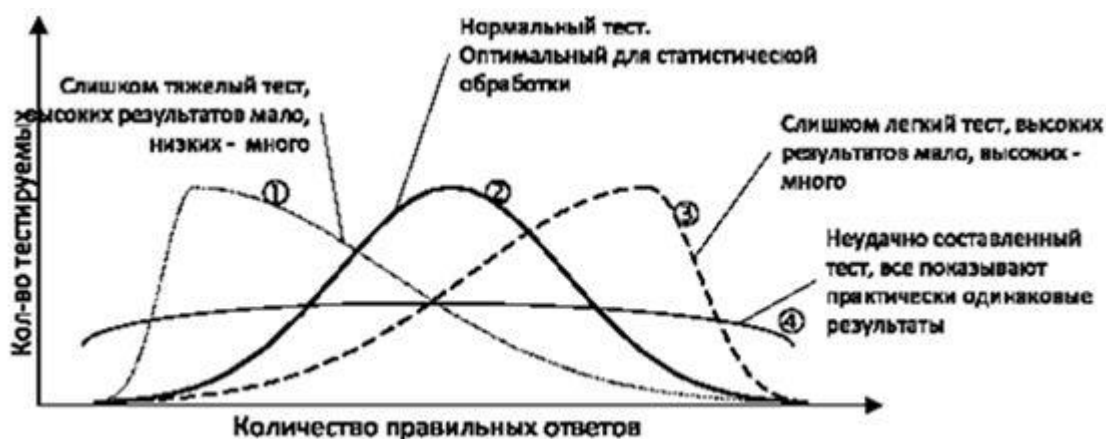


Рисунок 2.

Совершенствование разработанного теста осуществляется на этапе апробации в следующей последовательности: задания, выполненные всеми испытуемыми, исключаются из теста (не более одного задания); задания, с которыми не справился ни один из испытуемых, также исключаются.

Обычно, трудность задания определяется в процентах лиц (или в условных единицах от 0 до 1), давших на него правильный ответ. Чем легче задание, тем выше процент справившихся с ним. Если же ни один из тестируемых в выборке не справляется с тестовым заданием, то подобное

задание следует исключить из теста, так как оно не несет никакой информации об индивидуальных различиях. В результате в тесте должны остаться ТЗ, с которыми справились один или несколько человек.

Потом проводится анализ с целью определения не корректно сформулированных ТЗ - это легкие задания, с которыми не справился сильный учащийся. Скорее всего, формулировка таких ТЗ не совсем понятна обучаемым и их также следует исключить из состава ТЗ.

Чем ближе трудность задания к 1,00 (100%) или к 0, тем меньшую информацию несет оно о дифференциации учащихся и студентов в учебных учреждениях. Чем ближе уровень трудности задания к 0,50 (50%), тем большими различительными возможностями обладает ТЗ. Трудность (сложность) является особенностью не только тестового задания, но и теста в целом. Тест может состоять как из простых ТЗ, так и из сложных. В общем виде трудность (сложность) является статистической категорией.

Системы оценивания знаний учащихся должны вытекать из целей тестирования. Для тестов, ориентированных на критерий, считается важным, что испытуемый, превысил заданный критерий и в этом случае результат считается успешным. Для нормативно - ориентированных тестов основанием для сравнения тестовых показателей являются статистические нормы. Особенно важно подобное сравнение при текущем, итоговом и рубежном контроле. Тестовые оценки могут формироваться по сто - или пятибалльной системам.

Как было сказано выше для организации многопользовательского режима тестирования необходимо введение дополнительных компонентов, таких как база данных и сервер базы данных, а также программного обеспечения для функционирования этой информационной системы.

Содержание многопользовательского тренажера должно быть ориентировано на содержание профессионального модуля. В нашем случае социальный заказчик сформулировал техническое задание: разработать многопользовательский тренажер по профессиональному модулю: «ПМ.09

Проектирование, разработка и оптимизация веб-приложений», разработанного для основной образовательной программы в соответствии с ФГОС СПО по специальности 09.02.04 «Информационные системы и программирование» [9] для языка программирования Python для изучения по разделу модуля.

Тематический фрагмент раздела приведен в таблице 1.

Таблица 1

Фрагмент раздела профессионального модуля

<i>Формируемая компетенция</i>	<i>Трудовые действия</i>	<i>Умения</i>	<i>Знания</i>
<i>ПК 9.3. Разрабатывать интерфейс пользователя веб-приложений в соответствии с техническим заданием.</i>	<p>Разрабатывать интерфейс пользователя.</p> <p>Разрабатывать анимационные эффекты</p>	<p>Разрабатывать программный код клиентской части Веб-приложений.</p> <p>Оформлять код программы в соответствии со стандартом кодирования.</p> <p>Использовать объектные модели Веб-приложений и браузера.</p> <p>Разрабатывать анимацию для Веб-приложений для повышения его доступности и визуальной привлекательности (Canvas).</p>	<p>Языки программирования и разметки для разработки клиентской части веб-приложений.</p> <p>Принципы работы объектной модели Веб-приложений и браузера.</p> <p>Технологии для разработки анимации.</p> <p>Способы манипуляции элементами страницы веб-приложения. Виды анимации и способы применения ее.</p>

Выводы по 1 главе

Актуальным становится использование интерактивных электронных тренажеров по разным темам учебных предметов, при профессиональной подготовке и при повышении квалификации с учетом индивидуальных особенностей обучающихся. В образовательных организациях оптимальным и эффективным является разработка и внедрение многопользовательских электронных тренажеров.

Многопользовательский электронный тренажер - это программно-методический комплекс, в основу которого положена система диагностики и интерпретации полученных ответов, алгоритмы целенаправленной тренировки обучающегося в процессе многократного повторного выполнения имитации технологического процесса, а также тестовых заданий разного формата.

Существенным отличием электронного тренажера от электронного учебного пособия является наличие сценария. Под сценарием понимается некоторая предопределенная последовательность команд, способных выполняться в автоматическом режиме. Сценарий тренажера - это покадровое распределение содержания учебного курса и его процессуальной части в рамках программных структур разного уровня и назначения. В общем случае сценарий представляет собой два взаимосвязанных руководства по реализации конкретного проекта педагогический и технологический сценарий.

Для организации многопользовательского режима тестирования необходимо введение дополнительных компонентов, таких как база данных и сервер базы данных, а также программного обеспечения для функционирования этой информационной системы.

ГЛАВА 2. ОБЕСПЕЧЕНИЕ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ ЭЛЕКТРОННЫХ ОБРАЗОВАТЕЛЬНЫХ РЕСУРСОВ

2.1. Состояние защищенности электронных образовательных ресурсов в ЮУрГТК

ГБПОУ «ЮУрГТК» - крупнейшее учебное заведение среднего профессионального образования в г. Челябинске. В Южно-Уральском государственном техническом колледже сегодня учится более 4000 студентов. Директор колледжа Тубер Игорь Иосифович – заслуженный учитель РФ, кандидат педагогических наук, почетный строитель России.

В колледже 300 преподавателей, среди которых кандидаты педагогических наук, заслуженные учителя РФ, почетные работники среднего профессионального образования, лауреаты Всероссийских конкурсов и премий, лауреаты премии губернатора, лауреаты премии Законодательного собрания Челябинской области в сфере образования, преподаватели высшей и первой квалификационных категорий. Студенты ЮУрГТК учатся по 19 образовательным программам базового и повышенного уровней среднего профессионального образования на бюджетной основе и коммерческой основе.

Материально-техническая база колледжа состоит из 4 учебных корпусов, 125 учебных кабинетов и лабораторий, учебно-производственных мастерских; учебно-производственных полигонов; благоустроенных общежитий. Библиотека колледжа оснащена уникальным архивным оборудованием и обеспечивает доступ студентов более чем к 80 тысячам экземпляров учебной, технической, научной литературы и периодики. Здесь же оборудована зона для самостоятельной работы студентов с электронными носителями информации с выходом в Интернет.

В основу успехов колледжа на современном этапе заложена сертифицированная в соответствии со стандартом ГОСТ Р ИСО 9001-2001

система менеджмента качества. Она стимулирует педагогический коллектив к постоянному совершенствованию, внедрению инновационных технологий, нацеливает на успех. Высокий уровень подготовки специалистов в колледже подтверждается победами на олимпиадах, конкурсах и выставках регионального и российского значения.

Сегодня в своей деятельности Южно-Уральский государственный технический колледж опирается на современные образовательные технологии – их внедрению уделяется большое внимание, а также на требования работодателей – заказчиков кадров квалифицированных специалистов.

Информатизация колледжа вышла на новый уровень, когда появилась компьютерная сеть, информационный портал, единое информационное пространство, электронная библиотека, система библиотечного обслуживания Ирбис, система электронного обучения Moodle, новые лицензионные программные продукты.

Программное обеспечение, используемое в колледже: Windows XP, Linux, AdobePhotoshop CS3, MicrosoftVisualStudio 2010, MS SQL, MicrosoftVisio 2007, BPWin 4/1, ERWin 7.0, Opera, AdobeDreamweaver CS3, MicrosoftOffice 2007, AdobeFlash CS3, CorelDraw, 3D Studio MAX, 1C 8.0, MathCAD, Компас.

Благодаря рациональной политике использования лицензионного и свободного программного обеспечения, доступности компьютеров для студентов. как в учебное так и в не урочное время, возможности работать в сети Интернет и сети электронной библиотеки федерального уровня с доступом к полнотекстовому содержанию, можно считать, что информационно-коммуникационная база колледжа отвечает современным требованиям.

Содержание образовательного процесса определяется на основе Государственных образовательных стандартов среднего профессионального

образования (ГОС СПО) в части Государственных требований к минимуму содержания и уровню подготовки выпускников.

Реализуемые ГБПОУ «ЮУрГТК» профессиональные образовательные программы по всем лицензированным специальностям разработаны с учетом требований соответствующих ФГОС СПО Федеральный государственный образовательный стандарт среднего профессионального образования по специальности 09.02.03 Программирование в компьютерных системах (утв. приказом Министерства образования и науки РФ от 28 июля 2014 г. № 804) [8] и Федеральный государственный образовательный стандарт среднего профессионального образования по специальности 09.02.04 Информационные системы (по отраслям) (утв. приказом Министерства образования и науки РФ от 14 мая 2014 г. № 525) [9], Федеральный государственный образовательный стандарт среднего профессионального образования по специальности 09.02.05 Прикладная информатика (по отраслям) (утв. приказом Министерства образования и науки РФ от 13 августа 2014 г. N 1001) [10] полностью соответствуют уровню подготовки специалистов, формам обучения, нормативному сроку обучения.

При разработке рабочих учебных программ и календарно-тематических планов особое внимание обращается на осуществление междисциплинарных связей, своевременно вносятся коррективы, отражающие изменения в экономике страны, внедрение в производство новой техники и технологий. Все программы имеют внутреннюю и внешнюю рецензии.

На основе рабочих программ преподавателями «ЮУрГТК» составлены календарно-тематические планы (КТП), которые рассмотрены на заседаниях цикловых комиссий и утверждены заместителем директора по учебной работе. Подготовка специалистов осуществляется на базовом уровне. Максимальный объём учебной нагрузки (включая все виды аудиторной и внеаудиторной учебной работы) и объём обязательных учебных занятий соответствует установленным нормативам.

Соотношение объёмов теоретической и практической подготовки

(включая лабораторные и практические занятия, выполнение курсовых проектов, производственную (профессиональную) практику) соответствует рекомендуемым нормативам.

Современные тенденции информатизации образования предполагают активное создание и использование электронных образовательных ресурсов (далее - ЭОР) в образовательных организациях.

Согласно ГОСТ Р 53620 - 2009 «Информационно-коммуникационные технологии в образовании. Электронные образовательные ресурсы. Общие положения» [3] электронный образовательный ресурс - это образовательный ресурс, представленный в электронной цифровой форме и включающий в себя структуру, предметное содержание и метаданные о них.

С использованием ЭОР возникают проблемы закупки современной техники, структурирования информации, защиты авторского права, защиты ЭОР от несанкционированного доступа (НСД), а также проблема подготовки кадров [18,33], способных использовать ЭОР как полноценное дидактическое средство. Традиционный подход к обеспечению безопасности ЭОР в образовательных организациях сводится к тому, чтобы обеспечить целостность и доступность ЭОР на каком-либо общем информационном портале в составе информационно-образовательной среды (далее – ИОС). В ЮУрГТК большая часть ЭОР сконцентрирована в системе дистанционного обучения - Moodle

Основные угрозы безопасности ЭОР связаны с уязвимостью веб-приложений, так как большинство ЭОР реализуются именно как веб-приложения, а также с используемыми механизмами проверки идентификации при доступе к ЭОР.

Уязвимости веб-приложений, как правило, приводят к выполнению кода на удаленном сервере и вследствие этого позволяют вносить несанкционированные изменения в приложения. Если при разработке ЭОР не учитываются требования безопасности, то у злоумышленника имеется возможность модифицировать исполняемые команды. Нарушителем может стать обучающийся, желающий, например, видоизменить результаты

тестирования или какого-либо контрольного мероприятия. Во избежание такого несанкционированного доступа на стадии разработки ЭОР и на стадии его размещения в информационную систему колледжа необходимо на программном уровне планировать ограничение доступа к редактированию и видоизменению данных [41].

Далее, ИОС и ЭОР в его составе должны рассматриваться как информационная система конкретного класса согласно руководящим документам ФСТЭК.

Очевидно, что нарушение целостности ЭОР при несанкционированном доступе внешнего нарушителя приведет к негативным последствиям для субъектов персональных данных: обучающиеся не получают объективную оценку своим знаниям, преподаватели получают искаженную картину достижения учебных целей, кроме того, вероятность нарушения интеллектуальных прав создателей ЭОР привносит в такую ситуацию компрометирующие риски. По этой причине ЭОР по степени защищенности необходимо приравнять к классу 2Б в соответствии с [41] и обеспечить для нее соответствующий программно-аппаратный и физический уровни защищенности.

В таком случае ЭОР должна быть обеспечена подсистемой управления доступом, подсистемой регистрации и учета, подсистемой обеспечения целостности. При этом целостность будет проверяться при загрузке системы, что актуально при противодействии сетевым атакам, помимо этого будет осуществлена физическая охрана устройств и носителей информации, предусматривающая контроль доступа в помещения сервера посторонних лиц. Кроме того, будет необходимо проводить периодическое тестирование функций средств защиты информации от НСД при изменении программной среды.

Данные требования отчасти усложнят ЭОР, однако, это необходимость для обеспечения безопасности ЭОР и защиты правообладателей и авторов от рисков компрометации.

Помимо программно-аппаратных и физических мер защищенности ЭОР как информационной системы класса 2Б администрация образовательной организации должна реализовать также меры организационно – правового характера. К таким мерам мы относим разработку концепции информационной безопасности образовательной организации, а в рамках этой концепции – разработку положения об использовании электронных образовательных ресурсах, в котором детализировано указать обязанности всех пользователей системы и ответственность инженерного состава. Многие образовательные организации уже разработали и активно используют такие положения [61-64].

В «ЮУрГТК» на момент проведения нашего исследования имелись следующие локальные правовые акты, так или иначе затрагивающие аспекты информационной безопасности: «Положение об официальном сайте ГБПОУ «ЮУрГТК», «Положение об обработке и защите персональных данных в ГБПОУ «ЮУрГТК», «Политика в отношении обработки персональных данных в ГБПОУ «ЮУрГТК», «Положение об организации работы по охране труда, обеспечению безопасности образовательного процесса в ГБПОУ «ЮУрГТК», «Правила пользования библиотекой». Все документы размещены на сайте колледжа - <http://sustec.ru/>.

Анализ литературных источников [11-13,17-19] показывает, что исследование защищенности является важным этапом в решении задач связанных с контролем над безопасностью системы, моделированием и оптимизацией архитектуры системы защиты. В общем случае защищенность системы оценивается двумя основными способами: самостоятельная оценка, которая выполняется собственником, администраторами системы и/или службой внутреннего аудита, независимой от подразделений информационных технологий и информационной безопасности организации; оценка независимым внешним исполнителем, которая, в свою очередь, имеет два распространенных варианта реализации - аудит и экспертная оценка.

Самооценка или самоконтроль - самый распространенный тип оценки защищенности, который осуществляется практически непрерывно с

использованием не описанных формально процедур. Для проведения самостоятельной оценки защищенности системы, ответственное за безопасность лицо может воспользоваться стандартом ГОСТ Р ИСО/МЭК 17799–2005 [5]. Согласно данному стандарту рекомендуется проводить периодические проверки соответствия текущей защищённости, требуемому уровню, политике безопасности и техническим требованиям, однако в нем ничего не говорится о рекомендуемых методах реализации проверок. В общем случае, процедуру оценки, проводят с применением различных методов опроса. Основными этапами данного подхода являются: использование опросных листов на основе требований стандарта; формирование на основе требований и рекомендаций стандарта и/или на основе мнений привлеченных специалистов – экспертов базы знаний; нахождение правил по анализу ответов на опросные листы. Необходимо также учитывать специфику системы, информационные активы и статистики по угрозам ИБ или инцидентам ИБ, собранных в процессе функционирования системы.

Опрос был проведен среди преподавателей ИТ- дисциплин и были сформулированы наиболее существенные угрозы информационной безопасности для ЭОР - это несанкционированное внесение изменений в персональную информацию студентов при проведении тестирования, а именно – в результаты учебной успеваемости, а также возможность внесения корректив в программные коды и настройки операционной системы. Таким образом, обнаружили основные уязвимости в использовании ЭОР в корпоративной информационной системе колледжа.

2.2. Направления совершенствования информационной защиты электронных образовательных ресурсов в ЮУрГТК

Предпринимаемые меры защиты должны быть адекватны вероятности осуществления данного типа угрозы и потенциальному ущербу, который может быть нанесен в том случае, если угроза осуществится (включая

затраты на защиту от нее). Необходимо иметь в виду, что многие меры защиты требуют достаточно больших вычислительных ресурсов, что в свою очередь существенно влияет на процесс обработки информации. Поэтому современный подход к решению этой проблемы заключается в применении принципов ситуационного управления защищенностью информационных ресурсов. Суть такого подхода заключается в том, что требуемый уровень безопасности информации устанавливается в соответствии с ситуацией, определяющей соотношение между ценностью перерабатываемой информации, затратами - снижением производительности программно-аппаратной части информационной системы организации, дополнительным расходом оперативной памяти и др., которые необходимы для достижения этого уровня, и возможными суммарными потерями - материальными, моральными, компроментационными и др.) от искажения и несанкционированного использования информации.

Необходимые характеристики защиты информационных ресурсов определяются в ходе ситуационного планирования при непосредственной подготовке технологического процесса защищенной обработки информации с учетом сложившейся ситуации, а также (в сокращенном объеме) во время процесса обработки. Выбирая защитные меры, приходится учитывать не только прямые расходы на закупку оборудования и программ, но и расходы на внедрение новинки, на обучение и переподготовку персонала. Важным обстоятельством является совместимость нового средства со сложившейся аппаратно-программной структурой объекта.

Зарубежный опыт в области защиты интеллектуальной собственности и отечественный опыт по защите государственных секретов показывают, что эффективной может быть только комплексная защита, сочетающая в себе такие направления защиты, как правовое, организационное и инженерно-техническое [15].

Правовое направление предусматривает формирование совокупности законодательных актов, нормативно-правовых документов, положений,

инструкций, руководств, требования которых являются обязательными в рамках сферы их деятельности в системе защиты информации.

Организационное направление – это регламентация производственной деятельности и взаимоотношений исполнителей на нормативно-правовой основе таким образом, что разглашение, утечка и несанкционированный доступ к конфиденциальной информации становятся невозможными или существенно затрудняются за счет проведения организационных мероприятий.

По мнению специалистов, организационные мероприятия играют большую роль в создании надежного механизма защиты информации, так как возможности несанкционированного использования конфиденциальных сведений в значительной мере обусловлены не техническими аспектами, а злоумышленными действиями, нерадивостью, небрежностью и халатностью пользователей или персонала защиты.

К организационным мероприятиям относятся:

- мероприятия, осуществляемые при проектировании, строительстве и оборудовании служебных и производственных зданий и помещений
- мероприятия, осуществляемые при подборе персонала;
- организация и поддержание надежного пропускного режима, охраны помещений и территории, контроля за посетителями;
- организация хранения и использования документов и носителей конфиденциальной информации;
- организация защиты информации;
- организация регулярного обучения сотрудников.

Одним из основных компонентов организационного обеспечения информационной безопасности организации является Служба информационной безопасности – в ГПБОУ «ЮУрГТК» этот функционал выполняет Информатизационный центр (далее – ИЦ). Именно от профессиональной подготовленности сотрудников ИЦ, наличия в их

арсенале современных средств управления безопасностью во многом зависит эффективность мер по защите информации.

Ядром инженерно-технического направления являются программно-аппаратные средства защиты информации, к которым относятся механические, электромеханические, электронные, оптические, лазерные, радио- и радиотехнические, радиолокационные и другие устройства, системы и сооружения, предназначенные для обеспечения безопасности и защиты информации. Под программным обеспечением безопасности информации понимается совокупность специальных программ, реализующих функции защиты информации и режима функционирования.

Сформированная совокупность правовых, организационных и инженерно-технических мероприятий выливается в соответствующую политику безопасности.

Политика безопасности определяет облик системы защиты информации в виде совокупности правовых норм, организационных (правовых) мер, комплекса программно-технических средств и процедурных решений, направленных на противодействие угрозам для исключения или минимизации возможных последствий проявления информационных воздействий. После принятия того или иного варианта политики безопасности необходимо оценить уровень безопасности информационной системы. Естественно, что оценка защищенности производится по совокупности показателей, основными из которых являются стоимость, эффективность, реализуемость.

Сформированный возможный сценарий действий нарушителя требует проверки системы защиты информации. Такая проверка называется «тестированием на проникновение». Цель – предоставление гарантий того, что для неавторизованного пользователя не существует простых путей обойти механизмы защиты.

Один из возможных способов аттестации безопасности системы – приглашение хакеров для взлома без предварительного уведомления

персонала сети. Для этого выделяется группа из двух-трех человек, имеющих высокую профессиональную подготовку. Хакерам предоставляется в распоряжение автоматизированная система в защищенном исполнении, и группа в течение 1–3 месяцев пытается найти уязвимые места и разработать на их основе тестовые средства для обхода механизмов защиты. Наемные хакеры представляют конфиденциальный доклад по результатам работы с оценкой уровня доступности информации и рекомендациями по улучшению защиты. Наряду с таким способом используются программные средства тестирования.

На этапе составления плана защиты в соответствии с выбранной политикой безопасности разрабатывается план его реализации. План защиты является документом, вводящим в действие систему защиты информации, который утверждается руководителем организации. Планирование связано не только с наилучшим использованием всех возможностей, которыми располагает компания, в том числе выделенных ресурсов, но и с предотвращением ошибочных действий, могущих привести к снижению эффективности предпринятых мер по защите информации.

Управление информационной безопасностью должно быть:

- устойчивым к активным вмешательствам нарушителя;
- непрерывным, обеспечивающим постоянное воздействие на процесс защиты;
- скрытым, не позволяющим выявлять организацию управления защитой информации;
- оперативным, обеспечивающим возможность своевременно и адекватно реагировать на действия злоумышленников и реализовывать управленческие решения к заданному сроку.

Кроме того, решения по защите информации должны быть обоснованными с точки зрения всестороннего учета условий выполнения поставленной задачи, применения различных моделей, расчетных и информационных задач, экспертных систем, опыта и любых других данных,

повышающих достоверность исходной информации и принимаемых решений.

Как показывает разработка реальных информационных систем, ни один из способов (мер, средств и мероприятий) обеспечения безопасности информации не является абсолютно надежным, а максимальный эффект достигается при объединении всех их в целостную систему защиты информации. Только оптимальное сочетание организационных, технических и программных мероприятий, а также постоянное внимание и контроль над поддержанием системы защиты в актуальном состоянии позволят с наибольшей эффективностью обеспечить решение постоянной задачи.

Методологические основы обеспечения информационной безопасности являются достаточно общими рекомендациями, базирующимися на мировом опыте создания подобных систем. Задача каждого специалиста по защите информации – адаптировать абстрактные положения к своей конкретной области – образовательной организаций, в которой всегда найдутся свои особенности и тонкости.

Анализ опытных разработок показывает, что одним из эффективных направлений обеспечения информационной безопасности является виртуализация информационных ресурсов, а также создание рубежа аутентификации/идентификации [11], [12], [25-27], [48], [50].

Выводы по 2 главе

ГБПОУ «ЮУрГТК» - крупнейшее учебное заведение среднего профессионального образования в г. Челябинске, обеспеченное мощной программно-аппаратной базой, имеющее собственную разветвленную корпоративную сеть и электронные образовательные ресурсы различной направленности. В ГБПОУ «ЮУрГТК» уделяется достаточно внимания обеспечению безопасности образовательного процесса.

Процедура оценки, состоящая в проведении опросов на основе требований стандарта по учету усредненного экспертного мнения специалистов, показал наличие таких угроз, как возможность несанкционированного внесения изменений в персональную информацию студентов при проведении тестирования, а именно – в результаты учебной успеваемости, а также возможность внесения корректив в программные коды и настройки операционной системы.

При внедрении многопользовательского тренажера имеющиеся уязвимости могут усилиться. В связи с чем, необходимо учитывать эти особенности функционирования данной информационной системы и внедрять средства защиты информации, соответствующие основным направлениям и подходам повышения защищенности информационных ресурсов в ГБПОУ «ЮУрГТК».

Анализ опытных разработок показывает, что одним из эффективных направлений обеспечения информационной безопасности является виртуализация информационных ресурсов и создание надежного рубежа аутентификации/идентификации.

ГЛАВА 3 РАЗРАБОТКА И АПРОБАЦИЯ МНОГОПОЛЬЗОВАТЕЛЬСКОГО ТРЕНАЖЕРА В ГПБОУ «ЮУрГТК»

3.1. Реализация многопользовательского тренажера для профессионального модуля на базе Moodle

С учетом проведенного анализа защищенности электронных образовательных ресурсов, выявленных тенденций обеспечения информационной безопасности информационных ресурсов мы реализовали многопользовательский тренажер для профессионального модуля, используя плагин CodeRunner (V3.3.0) для Moodle, организовав, так называемую «песочницу».

Песочница — специально выделенная среда для безопасного исполнения компьютерных программ. Обычно представляет собой жёстко контролируемый набор ресурсов для исполнения гостевой программы — например, место на диске или в памяти. Доступ к сети, возможность общаться с главной операционной системой или считывать информацию с устройств ввода обычно либо частично эмулируют, либо сильно ограничивают. Песочницы представляют собой пример виртуализации [48].

Повышенная безопасность исполнения кода в песочнице зачастую связана с большой нагрузкой на систему — именно поэтому некоторые виды песочниц используют только для неотлаженного или подозрительного кода.

Общие сведения о плагине CodeRunner. CodeRunner - это плагин для Moodle, который позволяет преподавателям запускать программу, чтобы оценить ответ обучающегося. Наибольшее распространение CodeRunner получила в курсах программирования, где студентов просят написать программный код в соответствии с какой-либо спецификацией, и этот код затем оценивается путем запуска его в серии тестов. Вопросы CodeRunner также использовались в других областях компьютерной науки и техники для

оценки вопросов, в которых возможно много разных правильных ответов, и для оценки правильности должна использоваться программа.

Независимо от направления, выбранного для теста, вопросы CodeRunner всегда выполняются в адаптивном режиме, в котором обучающиеся могут нажать кнопку «Проверить», чтобы проверить, проходит ли их код тесты, определенные в вопросе. Если код не проходит, студенты имеют возможность подать код на проверку повторно, хотя и, как правило, с небольшим штрафом. В типичном режиме «все или ничего» все тестовые примеры должны пройти, если за представление присуждаются какие-либо оценки. Возможно настроить вопросы CodeRunner таким образом, чтобы оценка определялась тем, сколько тестов код успешно прошел.

CodeRunner и его предшественники *pycode* и *ccode* используются в университетской практике, выполняя более миллиона заявок на вопросы студентов на Python, C, JavaScript, PHP, Octave и Matlab. Другие курсы с использованием Moodle / CodeRunner включают в себя:

1. EMTH171 Математическое моделирование и вычисления
2. SENG02 Software Engineering I
3. COSC261 Формальные языки и компиляторы
4. COSC367 Вычислительный интеллект
5. ENCE360 Операционные системы
6. SENG365 Web Computing Architecture

CodeRunner в настоящее время поддерживает Python2 (считается устаревшим), Python3, C, C ++, Java, PHP, JavaScript (NodeJS), Octave и Matlab. Архитектура позволяет легко модифицировать ее и на другие языки.

CodeRunner можно безопасно использовать на установленном сервере Moodle, при условии, что программное обеспечение песочницы, в котором выполняется код («Jobe»), установлено на отдельном компьютере с соответствующей защитой и межсетевым экраном. Однако, если тесты на основе CodeRunner будут использоваться для тестов и итоговых экзаменов,

разработчики рекомендуют использовать отдельный сервер Moodle, как по причинам нагрузки, так и для того, чтобы различные средства связи Moodle, такие как чат и обмен сообщениями, могли быть отключены, не влияя на другие информационные потоки.

Один 4-ядерный сервер Moodle может обрабатывать среднюю частоту отправки вопросов теста около 60 вопросов в минуту, поддерживая время ответа менее 3-4 секунд, при условии, что сам код студента выполняется за доли секунды. Тип вопроса CodeRunner может быть установлен в любой современной системе Moodle (версии 2.6 или более поздней, включая версию 3.0), в Linux, Windows и Mac. В целях безопасности представленные задания обычно выполняются на отдельном компьютере, который называется «сервер Jobe» или «компьютер песочницы Jobe».

Мы произвели настройку песочницы на отдельно развернутом сервере. Мы развернули сервер Jobe на основе собственных вычислительных ресурсах, следуя инструкциям на <https://github.com/trampgeek/jobee>. Затем мы использовали интерфейс администратора Moodle для плагина CodeRunner, чтобы указать имя хоста Jobe и, также номер порта. Хотя CodeRunner имеет гибкую архитектуру, которая поддерживает различные способы выполнения задач обучающегося в защищенной среде (песочнице). В этой изолированной программной среде используется отдельный сервер, разработанный специально для использования CodeRunner, который в спецификации называется Jobe . Основные этапы настройки зафиксированы на рисунках 2-10.

Поскольку мы создали Jobe на отдельном сервере, JobeSandbox полностью изолирует код отдельного пользователя (студента) от сервера Moodle. Существует малая вероятность того, что некий злонамеренный программный код может отключить сервер песочницы и появится возможность нарушения безопасности на сервере Moodle, например, для взлома базы данных Moodle. Однако, Moodle ведет подробный журнал всех

событий, поэтому студент, сознательно нарушающий безопасность, принимает на себя риск быть уличенным нарушителем.

CodeRunner требует двух отдельных плагинов, один предназначен для конкретных типов вопросов, а другой - для специализированного адаптивного поведения. Плагины находятся в двух разных репозиториях github: github.com/trampgeek/moodle-qbehaviour_adaptive_adapted_for_coderunner и github.com/trampgeek/moodle-qtype_coderunner.

Используя стандартный веб-интерфейс Moodle в качестве преподавателя на созданном нами курсе создаем новый вопрос CodeRunner. Например, простой тестовый вопрос Python3: «*Напишите функцию $sqr(n)$, которая возвращает квадрат своего параметра n* ». Вводное краткое руководство в неполном руководстве по составлению вопросов содержит пошаговые инструкции по созданию банка вопросов. В качестве альтернативы возможно просто создать вопрос, используя интерактивную справку в форме создания вопроса.

Мы запустили модульные тесты для установки правильного хоста и необходимой конфигурации для сервера Jobe и использовали встроенные типы вопросов CodeRunner.

Приведенная ниже блок-схема показывает компоненты CodeRunner.

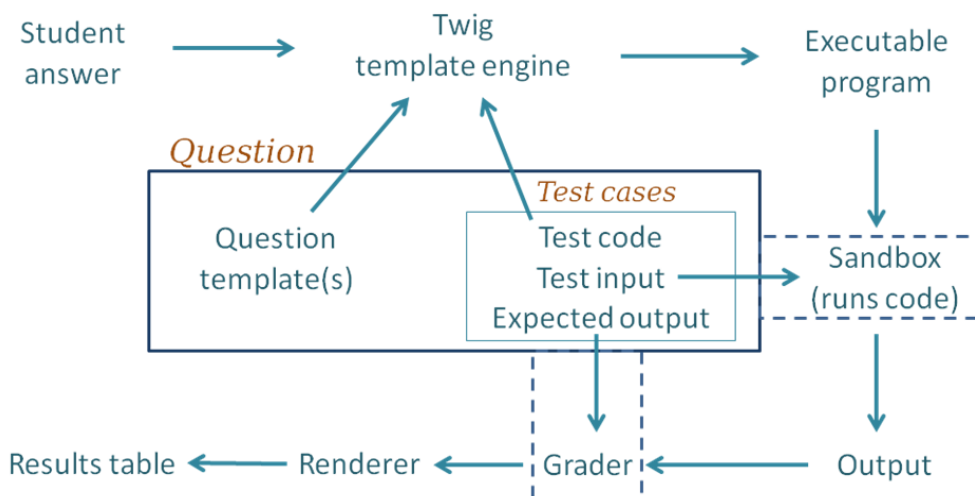


Рисунок3 - Архитектура CodeRunner

Процедура оценки ответов следующая:

1. Для каждого из тестовых случаев шаблонизатор Twig (<https://translate.google.com/translate?depth=1&hl=ru&prev=search&rurl=translate.google.ru&sl=en&sp=nmt4&u=http://twig.sensiolabs.org/&id=17259,15700022,15700043,15700124,15700149,15700186,15700191,15700201,15700237,15700242>) объединяет представленный студентом ответ с шаблоном вопроса вместе с кодом для этого конкретного тестового примера, чтобы получить исполняемую программу.

2. Исполняемая программа передается в песочницу. Песочница компилирует программу и запускает ее, используя стандартный ввод, предоставленный тестовым набором.

3. Вывод из прогона передается в любой настроенный компонент Grader (См. Рис...Архитектура), как и ожидаемый вывод, указанный для тестового примера. Наиболее распространенным грейдером является грейдер с «точным соответствием», но доступны и другие типы.

4. Выходные данные грейдера - это «объект результата теста», который содержит (помимо прочего) атрибуты «Ожидаемый» и «Полученный».

5. Вышеупомянутые шаги повторяются для всех тестовых случаев, давая массив объектов результатов теста (не показано явно на рисунке).

6. Все результаты теста передаются в средство визуализации вопросов CodeRunner, которое представляет их пользователю в виде таблицы результатов. Прошедшие тесты отмечены зеленой галочкой, а проваленные - красным крестиком. Обычно вся таблица окрашивается в красный цвет, если какой-либо тест не пройден, или в зеленый цвет, если все тесты пройдены.

Приведенное выше описание несколько упрощено.

Во-первых, не всегда необходимо запускать разные задания в песочнице для каждого теста. Вместо этого все тесты часто можно объединить в одну исполняемую программу. Это достигается использованием так называемого «шаблона комбинатора», а не более простого «шаблона для каждого теста», описанного выше. Шаблоны комбинатора полезны для вопросов разнообразия функций записи и записи . Они не часто используются с вопросами о написании программы , которые обычно тестируются с различными стандартными входными данными, поэтому требуется многократное выполнение. Кроме того, даже с вопросами о записи функции, которые имеют шаблон комбинатора, CodeRunner вернется к выполнению тестов по одному (все еще используя шаблон комбинатора), если выполнение всех тестов в одной программе дает некоторую форму времени выполнения ошибка, чтобы учащиеся могли быть представлены со всеми результатами теста вплоть до того, который не прошел.

Шаблоны комбинатора описаны в разделе «Шаблоны».

Во-вторых, приведенное выше описание процесса оценки игнорирует шаблонные грейдеры, которые выполняют оценку, а также тестирование. Они поддерживают более продвинутые стратегии тестирования, такие как проведение тысяч тестов или присвоение оценок более сложными способами, чем это возможно при стандартной опции «все или ничего» или линейного суммирования отдельных тестовых оценок.

Шаблонный грайдер для каждого тестового случая может использоваться для определения каждой строки таблицы результатов, или шаблонный грайдер комбинатора может использоваться для определения всей панели обратной связи, с таблицей результатов или без нее.

Типы вопросов. CodeRunner поддерживает широкий спектр типов вопросов и может быть легко расширен для поддержки других. Тип вопроса CodeRunner определяется прототипом вопроса, который задает параметры времени выполнения, такие как язык выполнения и песочница, а также

шаблон, который определяет, как программа тестирования строится из контрольных примеров вопроса плюс представления студента. Прототип также определяет, оценивается ли правильность представления учащегося с использованием *EqualityGrader*, *NearEqualityGrader* или *RegexGrader*. *EqualityGrader* ожидает, что результат выполнения теста будет точно соответствовать ожидаемому результату для тестового случая. *NearEqualityGrader* похож, но нечувствителен к регистру и допускает изменения в количестве пробелов (например, пропущенные или лишние пустые строки или несколько пробелов, где ожидался только один). Вместо этого *RegexGrader* ожидает совпадения с регулярным выражением. *EqualityGrader* рекомендуется для любого обычного использования, так как он поощряет студентов к тому, чтобы их результаты были в точности правильными; они должны быть в состоянии повторно представить почти правильные ответы для небольшого штрафа, что, как правило, лучше, чем пытаться присваивать знаки детали на основе совпадений регулярных выражений.

Тестовые случаи определяются преподавателем для проверки кода студента. Каждый тестовый пример определяет фрагмент тестового кода, стандартный ввод, который будет использоваться при запуске тестовой программы, и ожидаемый результат этого запуска.

Тестовая программа составлена из информации о тестовом наборе и представления студента с использованием шаблона, определенного прототипом. Шаблон может быть либо шаблоном для каждого теста, который определяет разные программы для каждого теста, либо шаблоном комбинатора, который может определять программу, которая объединяет несколько тестов в один прогон. Шаблоны описаны в разделе «Шаблоны».

Пример типа вопроса. Тип вопроса «С-функция» предполагает, что учащиеся должны предоставить функцию «С», а также возможные дополнительные вспомогательные функции в соответствии со спецификацией. В качестве тривиального примера можно задать вопрос:

«Напишите функцию C с сигнатурой `int sqr(int n)` которая возвращает квадрат своего параметра `n`». Затем автор предоставит несколько тестовых примеров в форме `printf("%d\n", sqr(-11));` и дать ожидаемый результат этого теста.

Шаблон для теста для такого типа вопроса затем оборачивает отправку и тестовый код в одну программу, такую как:

```
#include <stdio.h> // --- Student's answer is inserted here ---- int main() {  
printf("%d\n", sqr(-11)); return 0; },
```

 который будет скомпилирован и запущен для каждого теста. Выходные данные прогона будут затем сравниваться с указанным ожидаемым выходным значением, и контрольный пример будет помечен как правильный или неправильный соответственно.

В этом примере предполагается использование шаблона для каждого теста, а не более сложного шаблона комбинатора, который фактически используется типом вопроса встроенной функции C. Смотрите раздел о *шаблонах* для получения дополнительной информации.

Встроенные типы вопросов. Файл
<moodlehome>/question/type/coderunner/db/builtin_PROTOTYPES.xml

представляет собой файл формата экспорта moodle-xml, содержащий определения всех встроенных типов вопросов. Во время установки и в конце любого обновления версии все вопросы прототипа из этого файла загружаются в категорию CR_PROTOTYPES в контексте системы. Системный администратор может редактировать эти прототипы, но это не рекомендуется, так как измененные версии будут потеряны при каждом обновлении. Вместо этого следует создать категорию LOCAL_PROTOTYPES (или другое такое имя по нашему выбору) и сохранить в ней копии всех вопросов-прототипов, требующих редактирования, с соответствующим изменением имени типа вопроса. В этой категории также можно создавать новые типы вопросов-прототипов. Редактирование прототипов обсуждается далее в этом документе.

Встроенные типы вопросов включают в себя следующее:

1. c_function . Это тип вопроса, рассмотренный в приведенном выше примере, за исключением того, что он использует шаблон комбинатора. Студент предоставляет только функцию (плюс возможные функции поддержки), и каждый тест (как правило) имеет форму `printf(format_string, func(arg1, arg2, ..))`

Шаблон для этого типа вопроса генерирует некоторые стандартные включения, за которыми следует код студента, за которым следует основная функция, которая выполняет тесты один за другим. Однако, если в каком-либо из тестовых случаев определены стандартные входные данные, шаблон раскрывается и выполняется отдельно для каждого тестового случая.

Способ выполнения C (или любой другой) программы не является частью определения типа вопроса: он определяется конкретной песочницей, в которую передается выполнение. Песочница `Jobe` использует компилятор `gcc` с языком, установленным для принятия C99, и с параметрами `-Wall` и `-Werror` , установленными в командной строке, чтобы выдавать все предупреждения и отклонять код, если есть какие-либо предупреждения.

2. cpp_function . Это версия C ++ предыдущего типа вопроса. Студент предоставляет только функцию (плюс возможные функции поддержки), и каждый тест (как правило) имеет форму `cout << func(arg1, arg2, ..)`.

Шаблон для этого типа вопроса генерирует некоторые стандартные включения, за которыми следует строка использования пространства имен `std`; сопровождаемый кодом студента, сопровождаемым основной функцией, которая выполняет тесты один за другим.

3. c_program и **cpp_program**. Эти два очень простых типа вопросов требуют, чтобы студент предоставил полную рабочую программу. Для каждого теста автор обычно предоставляет

стандартный `stdin` и указывает ожидаемый стандартный `stdout` . Программа компилируется и запускается «как есть», и в стандартном режиме оценки «все или ничего» должен выводиться правильный вывод для всех тестовых случаев, которые должны быть помечены как правильные.

4. **python3** . Используется для большинства вопросов по Python3. Для каждого теста сначала выполняется код студента, а затем код теста.
5. **python3_w_input** . Вариант вопроса *python3*, в котором функция `input` переопределяется в начале программы, так что используемые ею стандартные символы ввода выводятся на стандартный вывод, как при вводе на клавиатуре во время интерактивного тестирования. Небольшой недостаток этого типа вопроса по сравнению с типом *python3* состоит в том, что код студента смещен вниз в файле, так что номера строк, присутствующие в любом синтаксисе или сообщениях об ошибках во время выполнения, не совпадают с номерами в исходном коде студента.
6. **python2** . Используется для большинства вопросов по Python2. Что касается *python3*, сначала выполняется код студента, а затем последовательность тестов. Этот тип вопроса следует считать устаревшим из-за широко распространенного перехода на Python3 через образовательное сообщество.
7. **java_method** . Это предназначено для раннего обучения Java, где студенты все еще учатся писать индивидуальные методы. Код студента - это отдельный метод плюс возможные вспомогательные методы, который обернут в классе вместе со статическим основным методом, содержащим предоставленные тесты (которые обычно вызывают метод студента и выводят результаты).

- 8. java_class** . Здесь студент пишет целый класс (или, возможно, несколько классов в одном файле). Затем тестовые примеры помещаются в метод `main` для отдельного общедоступного тестового класса, который добавляется в класс студентов и затем выполняется целое. Класс, который пишет студент, может быть как частным, так и публичным; шаблон заменяет любые вхождения `public class` в представлении просто `class` . В то время как студенты могут создавать программы, которые не будут корректно обрабатываться этой упрощенной заменой, в результате они просто не пройдут тесты. Вскоре они научатся писать свои классы ожидаемым образом (то есть с `public` и `class` в одной строке, разделенными одним пробелом)!
- 9. java_program** . Здесь студент пишет полную программу, которая компилируется, затем выполняется один раз для каждого теста, чтобы увидеть, генерирует ли он ожидаемый результат для этого теста. Имя основного класса, которое необходимо для именования исходного файла, извлекается из представления с помощью поиска по регулярному выражению для открытого класса с помощью `public static void main` метода `public static void main` .
- 10.octave_function** . При этом используется система Octave с открытым исходным кодом для обработки заявок, похожих на `matlab`.
- 11.Php** . Вопрос `php`, в котором студенческая работа представляет собой обычный `php`-файл, с PHP-кодом, заключенным в теги, а выводом является обычный вывод PHP, включающий все содержимое HTML вне тегов `php`

Другие менее часто используемые встроенные типы вопросов: *c_full_main_tests* , *python3_w_input* , *nodejs* , *pascal_program* и *pascal_function*

Как будет обсуждаться позже, этот базовый набор типов вопросов можно настраивать или расширять различными способами.

В разработанном МТ мы использовали типы вопросов с 1 по 9.

Помимо перечисленных имеются и другие некоторые более специализированные типы вопросов

Типы вопросов UOC включают:

1. **python3_cosc121** . Это сложный тип вопросов Python3, который используется в Кентерберийском университете почти для всех вопросов курса COSC121. Вариант ответа студента сначала проходит через анализатор исходного кода Pylint, и оно отклоняется, если Pylint дает какие-либо ошибки. В противном случае тестирование продолжается как обычно, то есть Pylint должен быть установлен на сервере песочницы. Мы ограничились типовыми встроенными вопросами, однако, имеется возможность для использования и специализированных типов, для этого можно настроить так, чтобы он требовал или запрещал определенные языковые конструкции (например, когда требуется, чтобы студенты переписывали цикл *for* как цикл *while*), или ограничивал размер функции заданным значением, или удалял *основную* функцию из кода студента, так что функции поддержки могут быть проверены в изоляции.

2. **Функция Matlab** . Используется для вопросов о функциях Matlab. Код студента должен быть объявлением функции, которое тестируется с каждым тестовым набором. Название на самом деле является ложью, так как этот тип вопроса теперь использует Octave, что гораздо эффективнее и проще для автора вопроса программировать в контексте CodeRunner. Однако у Octave много тонких отличий от Matlab, и некоторые проблемы неизбежны. Пусть покупатель будет бдителен.

3. **matlab_script** . Как и `matlab_function`, это ложь, поскольку на самом деле она использует Octave. Сначала выполняется тестовый

код (который обычно устанавливает контекст), а затем выполняется код студента, который может генерировать или не генерировать выходные данные в зависимости от контекста. Наконец выполняется код в Extra Template Data (если есть). Функция `disp` Octave заменена на ту, которая более близко имитирует Matlab, но, как указано выше: `caveat emptor`.

Шаблоны - это ключ к пониманию того, как проверяется представление. У каждого вопроса есть шаблон, либо импортированный из типа вопроса, либо явно настроенный, который определяет, как исполняемая программа создается из ответа учащегося, тестового кода и другого пользовательского кода внутри самого шаблона.

Шаблон вопроса может быть шаблоном для каждого теста или шаблоном комбинатора. Первый проще; он применяется один раз для каждого теста в вопросе, чтобы получить исполняемую программу, которая отправляется в песочницу. Каждое такое выполнение определяет одну строку таблицы результатов. Шаблоны комбинатора, как следует из названия, могут объединять несколько тестовых наборов в одно выполнение при условии, что для любого из тестовых примеров нет стандартного ввода. Сначала мы обсудим более простой шаблон для каждого теста.

Шаблоны для каждого теста имеются в *Per_test_template* - это, по сути, программа с «заполнителями», в которую вставляются ответ учащегося и тестовый код для запускаемого тестового примера. Затем расширенный шаблон отправляется в песочницу, где он компилируется (при необходимости) и запускается со стандартным вводом, определенным в тестовом примере. Выходные данные, возвращаемые из песочницы, затем сопоставляются с ожидаемым выходным сигналом для тестового примера, где выбранный валидатор определяет «совпадение»: точное совпадение, почти точное совпадение или совпадение регулярного выражения.

Синтаксис Twig `{{STUDENT_ANSWER | e ('py')}}` приводит к тому, что представление студента фильтруется с помощью функции `escape`,

соответствующей языку Python, которая экранирует все символы двойной кавычки и обратной косой черты с добавленной обратной косой чертой. Любой вывод, записанный в *stderr*, интерпретируется CodeRunner как ошибка времени выполнения, которая прерывает последовательность тестов, поэтому студент видит вывод ошибок только в первом тестовом примере.

Некоторые другие сложные типы вопросов, которые мы использовали, включают в себя:

1. Вопрос Matlab, в котором код шаблона (также Matlab) разбивает код студента на функции, проверяя длину каждого из них, чтобы убедиться, что он не слишком длинный, прежде чем приступить к маркировке.

2. Другой сложный вопрос Matlab, в котором код шаблона, написанный на Python, выполняет код Matlab учащегося, затем запускает пример ответа, предоставленный в вопросе, извлекает все числа с плавающей запятой и сравнивает числа равенства с некоторым заданным допуском.

3. Вопрос на Python, где код студента на самом деле является компилятором для простого языка. Код шаблона запускает компилятор студента, передает его вывод через ассемблер, который генерирует файл класса JVM, а затем запускает этот класс с JVM, чтобы проверить его правильность.

4. Вопрос на Python, где представление студентов вовсе не является кодом, а представляет собой текстовое описание конечного автомата для данной диаграммы перехода; код шаблона оценивает правильность предоставленного автомата.

Во втором примере, приведенном выше, используются две дополнительные функции CodeRunner, не упомянутые до сих пор:

- возможность настройки редактора кода Ace, который используется для выделения полей ввода кода с подсветкой синтаксиса, для использования в поле ответа ученика языка, отличного

от того, который использовался для запуска представления в песочнице.

- использование переменной шаблона QUESTION, которая содержит все атрибуты вопроса, включая текст вопроса, пример ответа и параметры шаблона (см. ниже).

Как сказано выше, синтаксис Twig `{{STUDENT_ANSWER | e ('py')}}` приводит к тому, что отправка студента фильтруется с помощью escape-функции Python, которая экранирует все символы двойной кавычки и обратной косой черты с добавленной обратной косой чертой. Escthon `e ('py')` - только один из доступных escape-агентов. Другие являются:

1. `e ('Java')`. Этот префикс ставит одинарные и двойные кавычки с обратной косой чертой и заменяет символы новой строки, возвратов, трансляции форм, возврата и табуляции их обычной экранированной формой (`\ n`, `\ r` и т. Д.).

2. `e ('c')`. Это псевдоним для `e («Java»)`.

3. `e ('MATLAB')`. Это позволяет избежать одинарных кавычек, процентов и символов новой строки. Он должен использоваться в контексте `sprintf Matlab`, например

```
student_answer = sprintf('{{ STUDENT_ANSWER | e('matlab')}}')
```

На рисунках 4,5 показана таблица результатов студента после отправки полностью правильного ответа и частично правильного ответа соответственно.

Question 1
Correct
Mark 5.00 out of 5.00

Paste the 5 lines of output from your simulator run into the answer box below. You get 1 mark for each correct line.

Answer:

```
Thing: 7
Number: 23
Blah: 11
Twaddle this is
Last line
```

Check

	Expected	Got	Comment	Mark	
✓	Thing: 7 Number: 23 Blah: 11 Twaddle this is Last line	Thing: 7 Number: 23 Blah: 11 Twaddle this is Last line	Line 0 right Line 1 right Line 2 right Line 3 right Line 4 right	5	✓

Passed all tests! ✓

Correct
Marks for this submission: 5.00/5.00.

Рисунок 4 - Таблица результатов студента после отправки полностью правильного ответа

Question 1
Partially correct
Mark 5.00 out of 5.00

Paste the 5 lines of output from your simulator run into the answer box below. You get 1 mark for each correct line.

Answer:

```
Thing: 77
Number: 23
Blah: 11
Twaddle this is
Lost line
```

Check

	Expected	Got	Comment	Mark	
✓	Thing: 7 Number: 23 Blah: 11 Twaddle this is Last line	Thing: 77 Number: 23 Blah: 11 Twaddle this is Lost line	Line 0 wrong Line 1 right Line 2 right Line 3 right Line 4 wrong	3	✓

Partially correct
Marks for this submission: 3.00/5.00.

Рисунок 5 - Таблица результатов студента частично правильного ответа

Шаблон-грейдер также можно использовать для оценки вопросов программирования, когда обычные грейдеры (например, точное или регулярное выражение соответствия результатов программы) неадекватны.

В качестве простого примера, предположим, что студент должен написать свою собственную функцию квадратного корня, так что его ответ в квадрате находится в пределах абсолютного допуска 0,000001 от правильного ответа. Чтобы запретить им использовать математический модуль, любое использование оператора `import` должно быть запрещено, но мы проигнорируем этот аспект, чтобы сосредоточиться на аспекте оценки.

Самый простой способ решить эту проблему - написать серию тестовых примеров в форме:

```
approx = student_sqrt(2)
right_answer = math.sqrt(2)
if math.abs(approx - right_answer) < 0.00001:
    print("OK")
else:
    print("Fail (got {}, expected {})".format(approx, right_answer))
```

где ожидаемый результат «OK».

Шаблонные грейдеры, которые выполняют код, предоставленный студентом, несколько сложны для правильной записи, поскольку им нужно вывести корректную запись JSON во всех ситуациях, обрабатывая такие проблемы, как посторонний вывод из кода студента, ошибки во время выполнения или синтаксические ошибки. Самый безопасный подход, как правило, состоит в том, чтобы запустить код студента в подпроцессе, а затем оценить результат.

Шаблонный грейдер для теста на вопрос ученика с квадратным корнем, который проверяет функцию `student_sqrt` студента с 1000 случайными числами в диапазоне от 0 до 1000 и выглядит следующим образом:

```
import subprocess, json, sys
student_func = """{{ STUDENT_ANSWER | e('py') }}"""
```

```

if 'import' in student_func:
    output = "The word \"import\" was found in your code!"
    result = {'got': output, 'fraction': 0}
    print(json.dumps(result))
    sys.exit(0)

test_program = """import math
from random import uniform
TOLERANCE = 0.000001
NUM_TESTS = 1000
{{ STUDENT_ANSWER | e('py') }}
ok = True
for i in range(NUM_TESTS):
    x = uniform(0, 1000)
    stud_answer = student_sqrt(n)
    right = math.sqrt(x)
    if abs(right - stud_answer) > TOLERANCE:
        print("Wrong sqrt for {}. Expected {}, got {}".format(x, right,
stud_answer))
        ok = False
        break

if ok:
    print("All good!")
"""

try:
    with open('code.py', 'w') as fout:
        fout.write(test_program)
    output = subprocess.check_output(['python3', 'code.py'],

```

```

stderr=subprocess.STDOUT, universal_newlines=True)
except subprocess.CalledProcessError as e:
    output = e.output

mark = 1 if output.strip() == 'All good!' else 0
result = {'got': output, 'fraction': mark}
print(json.dumps(result)).

```

На следующих рисунках 6, 7 показаны варианты результатов тестов.

You are to write a Python3 function `my_sqrt(x)` that takes a floating point number `x` in the range 0, 1000 and computes an approximation to the square root of `x` to within an absolute accuracy of 0.000001. Your function is not permitted to import any other modules.

Answer:

```

1 def my_sqrt(x):
2     NUM_REFINEMENTS = 8
3     approx = 0.5 * x
4     for i in range(NUM_REFINEMENTS):
5         better = 0.5 * (approx + x/approx)
6         approx = better
7     return better

```

Check

	Test	Got	
✓	Testing with 1000 random numbers	All good!	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Рис.6 Вариант результата теста1

You are to write a Python3 function `my_sqrt(x)` that takes a floating point number `x` in the range 0, 1000 and computes an approximation to the square root of `x` to within an absolute accuracy of 0.000001. Your function is not permitted to import any other modules.

Answer:

```

1 def my_sqrt(x):
2     NUM_REFINEMENTS = 8
3     approx = 0.5 * x
4     for i in range(NUM_REFINEMENTS)
5         better = 0.5 * (approx + x/approx)
6         approx = better
7     return better

```

Check

Test	Got
✗ Testing with 1000 random numbers	File "code.py", line 8 <pre> for i in range(NUM_REFINEMENTS) ^ SyntaxError: invalid syntax </pre>

Your code must pass all tests to earn any marks. Try again.

Incorrect
Marks for this submission: 0.00/1.00. Accounting for previous tries, this gives 1.00/1.00.

Рисунок 7 - Вариант результата теста 2

Очевидно, что написание вопросов с использованием шаблонных грейдеров гораздо сложнее, чем с использованием обычного встроенного грейдера, основанного на равенстве. Мы использовали встроенный грайдер.

3.2. Апробация многопользовательского тренажера на базе ЮУрГТК

Мы произвели настройку таблицы результатов следующим образом. Выходные данные стандартных грейдеров представляют собой список так называемых объектов *TestResult*, каждый из которых имеет следующие поля (которые включают в себя фактические данные тестового примера):

- testcode // The test that was run (trimmed, snipped)
- incorrect // True iff test passed fully (100%)
- expected // Expected output (trimmed, snipped)
- mark // The max mark awardable for this test
- awarded // The mark actually awarded.

```
got      // What the student's code gave (trimmed, snipped)
stdin    // The standard input data (trimmed, snipped)
extra    // Extra data for use by some templates
```

Поле с именем *result_columns* в форме создания вопроса было использовано нами в таблице результатов.

Каждый спецификатор столбца сам по себе является списком, обычно с двумя или тремя элементами. Первый элемент - это заголовок столбца, второй элемент - это обычно поле из объекта *TestResult*, отображаемого в столбце (одно из значений, перечисленных выше), а необязательный третий элемент - это *sprintf* строка формата, используемая для отображения поля. Грейдеры шаблонов для каждого теста могут добавлять свои собственные поля, которые также могут быть выбраны для отображения. Также возможно объединить несколько полей в столбец, добавив дополнительные поля к спецификатору: они должны предшествовать *sprintf* спецификатору формата, который затем становится обязательным. Например, для отображения *Mark Fraction* столбца в форме можно использовать *0.74 out of 1.00* спецификатор формата столбца ["Mark Fraction", "awarded", "mark", "%.2f out of %.2f"].

В качестве особого случая формат *%h* означает, что поле результата теста должно быть взято как готовый к выводу HTML и не должно подвергаться дальнейшей обработке; это может быть полезно для шаблонов пользовательских грейдеров, которые генерируют вывод HTML, таких как графика SVG, и мы также использовали его в вопросах, где вывод из программы студента был HTML.

Мы сделали небольшое упрощение. Синтаксис фактически допускает выражения в форме:

```
filter(testResultField [,testResultField]... )
```

где *filter*- имя встроенной функции фильтра, которая каким-либо образом фильтрует заданное поле (поля) *testResult*. На данный момент единственной такой встроенной функцией фильтра является *diff*. Это (или

было) функция, принимающая два поля результатов теста в качестве параметров и возвращающая строку HTML, представляющую первое поле теста со встроенными элементами HTML `<ins>` и ``, которые показывают вставки и удаления, необходимые для преобразования первого поля во второй. Это было использовано для обеспечения поддержки кнопки.

Показ различий автоматически отображается, если ответ помечен как неправильный, и если используется грейдер с точным соответствием. Следовательно, функция *diff* фильтра больше не функционирует, но остается синтаксически поддерживаемой для поддержки устаревших вопросов, которые ее используют.

Начиная с версии 3.3.0, CodeRunner поддерживает подключаемые пользовательские интерфейсы, хотя администратор должен установить плагин. В настоящее время в CodeRunner встроены два пользовательских интерфейса: Ace и Graph. Мы выбрали необходимый пользовательский интерфейс через раскрывающееся меню в разделе настроек формы автора вопроса. Выбор управляет редактированием полей предварительного ответа и предварительного ответа формы авторинга и ответа учащегося в реальном тесте. Редактор Ace всегда используется для редактирования самого шаблона, если только он не отключен с помощью *шаблона*, в поле создания *используется* флажок *ace*.

Плагин Graph UI, который на данный момент следует рассматривать как экспериментальный, предоставляет простые возможности рисования графиков для поддержки вопросов, когда ученика просят нарисовать или отредактировать график. По умолчанию пользовательский интерфейс Graph, разработанный для конечных автоматов, рисует ориентированные графы, позволяет узлам помечаться как состояния *Accept* и позволяет входящим начальным ребрам. Например:

Другие параметры шаблона могут быть добавлены по мере необходимости.

Все активные плагины пользовательского интерфейса CodeRunner как в форме создания вопроса, так и на странице тестирования обучающегося можно включать и выключать нажатием клавиши CTRL-ALT-M, поочередно открывая и скрывая элемент textarea.

Авторы вопросов могут написать свои собственные плагины пользовательского интерфейса; файл JavaScript с именем формы ui_something.jsv папке <moodlehome>/question/type/coderunner/amd/srcпредполагается, что это плагин пользовательского интерфейса и автоматически добавляется в выпадающее меню доступных плагинов. Такие файлы плагинов должны быть модулями AMD и должны реализовывать интерфейс, определенный в файле <moodlehome>/question/type/coderunner/amd/src/userinterfacewrapper.js

Каждый тип вопроса определяется вопросом-прототипом, который является еще одним вопросом в базе данных, от которого могут наследоваться новые вопросы. На панели имеется и вариант сохранения текущего вопроса в качестве прототипа. Необходимо ввести имя для нового типа вопроса. По умолчанию необходимо начинать имя вопроса со строки PROTOTYPE_, за которой следует имя типа. Например, PROTOTYPE_python3_OOP.

Текст вопроса вопроса-прототипа отображается на панели «Сведения о типе вопроса» в форме создания вопроса.

CodeRunner ищет вопросы-прототипы только в контексте текущего курса. Поиск включает родительские контексты, обычно видимые только администратору, такие как системный контекст; все встроенные прототипы находятся в контексте этой системы. Таким образом, если преподаватель в одном курсе создает новый тип вопроса, он сразу же появится в списке типов вопросов для всех авторов, редактирующих вопросы в этом курсе, но он не будет виден авторам в других курсах.

Когда создается вопрос определенного типа, включая определяемые пользователем типы, все так называемые настраиваемые поля наследуются

от прототипа. Это означает, что изменения в прототипе затронут все «наследные» вопросы. Чтобы уменьшить путаницу в пользовательском интерфейсе, настраиваемые поля подразделяются на базовые (для каждого шаблона теста, сортировщика, селекторов столбцов таблицы результатов и т.д), а также «расширенные». К последним относятся язык, песочница, время ожидания, ограничение памяти и функция «сделать этот вопрос прототипом».

Покажем, как прототип вопроса Python используется для определения нового типа вопроса *c_via_python*, который имитирует встроенный *min* вопроса *c_program*, и при этом обеспечивает большую гибкость. Чтобы создать новый тип вопроса, используем шаблон:

1. Создать новый вопрос CodeRunner.
2. Выбрать тип вопроса *python3*
3. Нажать *Настроить*
4. Заменить содержимое *шаблона* текстовой области с кодом шаблона ниже.
5. Снять флажок *Комбинатор Is*
6. Ввести `DEMO_PROTOTYPE_C_using_python` в качестве имени вопроса
7. Ввести любой текст для описания типа вопроса в текстовой области *Вопрос*. Этот текст будет показан всем авторам, использующим этот новый тип вопроса, если они откроют раздел *сведений* о *типе* вопроса в форме создания вопроса.
8. Открыть расширенную настройку
9. Set *Является ли прототип?* на *Yes* (*определяется пользователем*)
10. Установить *тип вопроса* в *c_via_python* .
11. Установить для *языка Ace* значение *c* , чтобы код учащихся был отредактирован как C, даже если прототип находится на языке Python.

12. Сохранить вопрос.

Теперь имеется новый тип вопроса `c_via_python` появляется в *вопросе* *типа* выпадающего автора формы редактирования для нового CodeRunner вопроса. Этот новый тип вопроса ведет себя как встроенный тип вопроса `c_program`, но он более гибкий; например, он может быть легко расширен для выполнения проверок переданного кода C перед компиляцией.

Полный вопрос Прототипом `c_via_python` типа вопроса входит в *образцах* папки распределения CodeRunner.

```
""" The template for a question type that compiles and runs a student-
submitted
```

```
    C program.
```

```
"""
```

```
import subprocess, sys
```

```
# Write the student code to a file prog.c
```

```
student_answer = """{{ STUDENT_ANSWER | e('py') }}"""
```

```
with open("prog.c", "w") as src:
```

```
    print(student_answer, file=src)
```

```
# Compile
```

```
{% if QUESTION.parameters.cflags is defined %}
```

```
cflags = """{{ QUESTION.parameters.cflags | e('py') }}"""
```

```
{% else %}
```

```
cflags = "-std=c99 -Wall -Werror"
```

```
{% endif %}
```

```
return_code = subprocess.call("gcc {0} -o prog
prog.c".format(cflags).split())
```

```
if return_code != 0:
```

```
    print("** Compilation failed. Testing aborted **", file=sys.stderr)
```

```
# If compile succeeded, run the code. Since this is a per-test template,
```

```
# stdin is already set up for the stdin text specified in the test case,
```

```
# so we can run the compiled program directly.
```

```

if return_code == 0:
    try:
        output = subprocess.check_output(["./prog"], universal_newlines=True)
        print(output)
    except subprocess.CalledProcessError as e:
        if e.returncode > 0:
            # Ignore non-zero positive return codes
            if e.output:
                print(e.output)
        else:
            # But negative return codes are signals - abort
            if e.output:
                print(e.output, file=sys.stderr)
            if e.returncode < 0:
                print("Task failed with signal", -e.returncode, file=sys.stderr)
                print("*** Further testing aborted ***", file=sys.stderr).

```

В текущих настройках доступны три служебных скрипта, связанных с CodeRunner. Первоначально предназначенные только для использования администратором, они оказываются полезными и для преподавателей.

Три сценария:

1. <moodle_home> /question/type/coderunner/bulktestindex.php

Этот сценарий отображает список всех категорий вопросов, доступных пользователю, который в данный момент вошел в Moodle на компьютере, на котором выполняется сценарий. Каждая категория отображается в виде интерактивной ссылки, которая затем запускает сценарий, который проверяет примеры ответов на все вопросы в этой категории, сообщая обо всех успехах и неудачах.

2. <moodle_home>

/question/type/coderunner/prototypeusageindex.php В этих сценариях отображается индекс, подобный приведенному выше, за исключением

того, что теперь по интерактивным ссылкам теперь запускается сценарий, который сообщает об использовании прототипа вопроса в этой категории.

3. <moodle_home>

/question/type/coderunner/downloadquizattempts.php Этот сценарий, который все еще является экспериментальным, отображает список всех тестов, доступных для вошедшего в систему пользователя, что позволяет им загружать электронную таблицу всех представлений в выбранный тест всех обучающихся. Загруженная электронная таблица подходит для автономного анализа и содержит информацию, отсутствующую в экспортированном файле ответов Moodle, такую как все промежуточные подпункты и предварительные проверки и время каждого такого действия.

Загрузка может быть в формате CSV или Excel; последнее рекомендуется для большинства случаев, потому что давняя ошибка в `frutcsv` функции PHP может привести к повреждению выходных файлов csv. Экспортированная электронная таблица Excel может быть открыта в Excel или Open Office и сохранена как CSV.

Формат загрузки сложен и требует хорошего понимания схемы базы данных Moodle. Используется следующий запрос, где каждая строка результата выводится как одна строка электронной таблицы.

```
SELECT
```

```
concat(quiza.uniqueid, qasd.attemptstepid, qasd.id) as uniquekey,  
quiza.uniqueid as quizattemptid,  
timestart,  
timefinish,  
u.firstname,  
u.lastname,  
u.email,  
qatt.slot,
```

```

    qatt.questionid,
    quest.name as qname,
    slot.maxmark as mark,
    qattsteps.timecreated as timestamp,
    FROM_UNIXTIME(qattsteps.timecreated,'% Y/%m/%d
%H:%i:%s') as datetime,
    qattsteps.fraction,
    qattsteps.state,
    qasd.attemptstepid,
    qasd.name as qasdname,
    qasd.value as value
FROM {user} u
JOIN {quiz_attempts} quiza ON quiza.userid = u.id AND quiza.quiz
= :quizid
JOIN {question_attempts} qatt ON qatt.questionusageid =
quiza.uniqueid
JOIN {question_attempt_steps} qattsteps ON
qattsteps.questionattemptid = qatt.id
JOIN {question_attempt_step_data} qasd on qasd.attemptstepid =
qattsteps.id
JOIN {question} quest ON quest.id = qatt.questionid
JOIN {quiz_slots} slot ON qatt.slot = slot.slot AND slot.quizid =
quiza.quiz
WHERE quiza.preview = 0
AND (qasd.name NOT RLIKE '^_' OR qasd.name = '-_rawfraction')
AND qasd.name NOT RLIKE '^_'
AND quest.length > 0
ORDER BY quiza.uniqueid, timestamp;

```

Этот запрос приводит к появлению нескольких почти идентичных строк для одного действия студента (например, отправка

ответа) с разными парами (qasdname, value). ['qasdname' является name столбцом таблицы вопрос-попытка-шаг-данных и value является значением, связанным с этим именем. Набор таких пар (ключ, значение) зависит от типа вопроса и конкретного записываемого действия пользователя.] В настоящее время так называемые «переменные поведения» - те, которые содержат подчеркивание - не включаются в экспорт, за исключением Переменная -_rawfraction. Это должно ограничить объем данных, но решение может измениться в будущем.

Обработка исходной электронной таблицы несколько сложна. Файл, quizsubmissions.py включенный в репозиторий git, определяет классы Python для упрощения процесса. Заявления

```
from quizsubmissions import QuizSubmissions
submission_data = QuizSubmissions(csvfilename)
```

импортировать экспортированную электронную таблицу и дает легкий доступ к данным о каком-либо конкретном ученике и их деятельности в викторине.

Например: submission_data['rjl83'].submissions[2].get_answer() возвращает окончательный ответ, представленный студентом rjl83 на вопрос 2.

Доступно много другой статистической информации, такой как имя студента, все промежуточные материалы и предварительные проверки, их время и т.д.

Оригинальный rucode был взят из аналога [CodingBat](#), сайта, который реализует простую функцию или метод, например, функцию, которая возвращает двойной квадрат своего параметра плюс 1. Код студента выполняется с помощью серии тестовых примеров, и результаты отображаются сразу после отправки в простой табличной форме, показывая каждый тестовый пример, ожидаемый ответ и фактический ответ. Ряды, в которых ответ, рассчитанный по коду ученика, верен, получают большую

зеленую галочку; неправильные строки получают большой красный крест. Код считается правильным, только если все тесты отмечены галочкой. Если код неверен, студенты могут просто исправить его и отправить повторно.

Соответственно, вопросы CodeRunner всегда используют адаптивное поведение Moodle, независимо от поведения, заданного для теста, в котором проводятся вопросы. Студенты могут проверить свой код на правильность, как только он был введен, и, если их ответ неверен, могут повторно отправить, как правило, с небольшим штрафом. Таким образом, оценка, полученная в тестировании по стилям программирования, определяется в первую очередь тем, сколько проблем студент может решить за данное время, и, во-вторых, тем, сколько кодов студент должен сделать по каждому вопросу.

Полная версия составленных вопросов представлена в Приложении 1.

3.3. Меры информационной защиты многопользовательского тренажера по профессиональному модулю в ГБПОУ ЮУрГТК

Современные системы обнаружения нарушения информационной безопасности включают в себя системы виртуализации, песочницы со встроенными системами антивирусной защиты и системы управления знаниями о киберугрозах и уязвимостях (Threat Intelligence) [38].

Многие ведущие производители антивирусного программного обеспечения внедрили песочницы в свои продукты как средство проактивной защиты, помимо этого появились специализированные программы, позволяющие запускать приложения в изолированной среде исполнения. Как правило, песочницы используют для запуска непроверенного кода из неизвестных источников как средство проактивной защиты от вредоносного кода, а также для обнаружения и анализа вредоносных программ. Также зачастую песочницы используются в процессе разработки программного

обеспечения для запуска «сырого» кода, который может случайно повредить систему или испортить сложную конфигурацию. Такие песочницы копируют основные элементы среды, для которой пишется код, и позволяют разработчикам быстро и безболезненно экспериментировать с неотлаженным кодом.

При настройке песочницы на отдельно выделенном сервере при разработке многопользовательского тренажера для ГБПОУ «ЮУрГТК» мы использовали изоляцию на основе полной виртуализации. Использование любой виртуальной машины в качестве защитного слоя над гостевой операционной системой, где установлен браузер и иные потенциально опасные программы, через которые пользователь может заразиться, дает достаточно высокий уровень защиты основной рабочей системы. Попытка модификации каких-либо объектов приведет к изменению лишь их копий внутри песочницы, реальные данные не пострадают. Контроль прав не дает возможности атаковать основную систему изнутри песочницы через интерфейсы операционной системы.

Кроме того, в качестве основной меры защиты нами была применена двухфакторная аутентификация.

Аутентификация (англ. authentication < греч. αὐθεντικός [authentikos] «реальный, подлинный» < αὐτός [authos] «сам; он самый») — процедура проверки подлинности пользователя путём сравнения введённого им пароля (для указанного логина) с паролем, сохранённым в базе данных пользовательских логинов, либо подтверждение подлинности электронного письма путём проверки цифровой подписи письма по открытому ключу отправителя, либо проверка контрольной суммы файла на соответствие сумме, заявленной автором этого файла [12]. Аутентификация является одним из первых рубежей информационной защиты, но, как и другие услуги защиты, может быть предложен только в контексте определенной стратегии защиты.

Проведя анализ существующих механизмов аутентификации пользователей нами выделено три основных характеристики, которыми обладает каждый из них – рисунок 8.

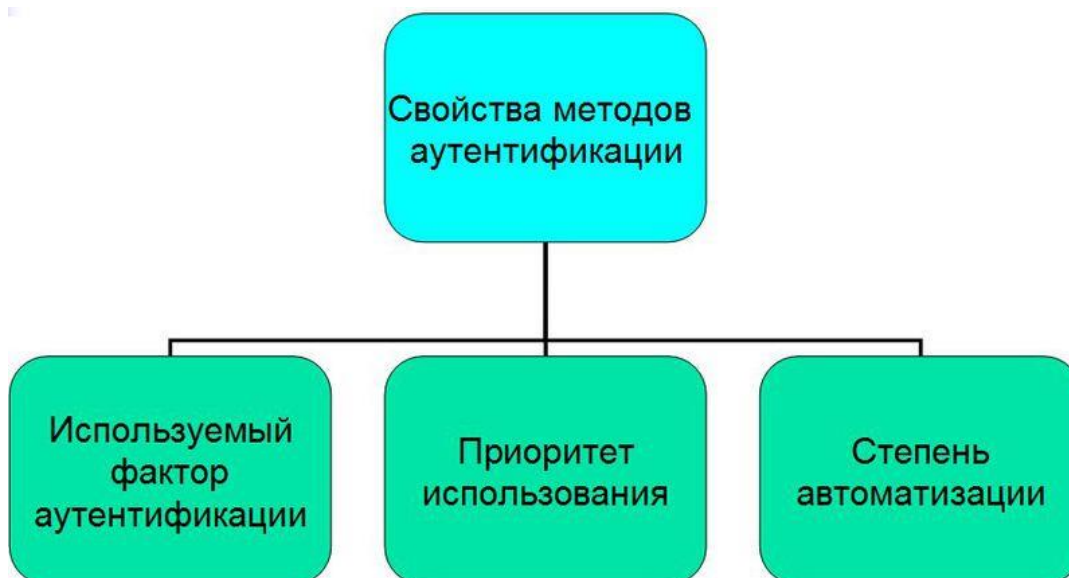


Рисунок 8 - Классификация механизмов аутентификации

Степень автоматизации может быть *полной*, либо *неполной*. Имеется ввиду автоматизация аутентификации со стороны системы, а не пользователя.

Приоритет использования — это порядок, в каком порядке пользователь пользуется данным способом аутентификации.

Как правило выделяют *Основной и резервный методы аутентификации*. Как видно из названия этот метод используется для штатного входа в систему. Самый распространённый из них — вход по паролю, использующийся в подавляющем большинстве компьютерных систем. Менее распространённым способом является использование аппаратных идентификаторов, на которые записываются ключи доступа или пользовательские пароли. Также в корпоративном секторе популярна двухфакторная аутентификация. Как правило, под этим понимается связка из е-токена и пин-кода вводимого пользователем, но встречаются и более

экзотические сочетания, состоящие, например, из биометрического сканера и аппаратного идентификатора или пользовательского пароля и ряд других.

Резервный метод аутентификации. В случае потери пароля или токена, либо взлома учётной записи в силу вступают резервные методы аутентификации. Наиболее распространены два метода: ответ на «секретный вопрос» и отправка пароля на доверенный почтовый ящик, указанный при регистрации. Есть много интересных модификаций данного способа аутентификации. Например, одним из первых было предложение использовать собственные «секретные вопросы». Что было, практически немедленно реализовано у ведущих провайдеров/ Так же можно привести пример продвинутой системы резервной аутентификации основанной на вопросах из базы онлайн знакомств. Таких вопросов много, они просты сами по себе, но в совокупности по ним можно достаточно чётко представить себе характер человека. При обращении система задаёт часть вопросов, из тех на которые пользователь ответил при регистрации, если он сумеет правильно ответить на большинство из них, то аутентификация считается успешной.

***last-resort* механизм («последней инстанции»).** Несмотря на все минусы и слабости таких механизмов резервного восстановления доступа к учётным записям, ведущие интернет-компании вынуждены ими пользоваться, ведь альтернативой этому является использование «last-resort» аутентификации, что можно перевести как аутентификация «последней инстанции», то есть механизма, к которому прибегают в самых крайних случаях, когда все остальные способы оказались бессильны. В данный момент это означает обращение к администраторам информационных систем, либо в специальные отделы поддержки клиентов, а это непомерно дорого, если учесть, что количество активных пользователей с каждым годом экспоненциально растёт [11, 12].

Используемый фактор аутентификации — аутентификация представляет из себя процесс сравнения информации, предоставляемой

пользователем, с эталонной. В зависимости от типа информации её можно отнести к одному из четырёх основных факторов, либо к их комбинации.

Фактор знания (Парольная аутентификация) — «то, что ты знаешь». Первый и самый распространённый на данный момент механизм аутентификации, ввод чего-либо, что известно только пользователю, например, пароля или ответа на секретный вопрос. Теоретически, это самый простой и безопасный метод проверки подлинности, так как он обладает достаточной криптостойкостью, его просто и дешево реализовать, а всё что нужно от пользователя, так это запомнить 8-12-ти символьную комбинацию из букв, цифр и различных знаков. Однако, на практике всё совершенно иначе.

Вещественный фактор (Аппаратная аутентификация) — «то, чем ты владеешь». Второй по популярности фактор аутентификации. В первую очередь под этим понимаются аппаратно-программные системы идентификации и аутентификации или устройства ввода идентификационных признаков [12]. В их состав входят аппаратные идентификаторы, устройства ввода-вывода (считыватели, контактные устройства, адаптеры, разъемы системной платы и др.) и соответствующее ПО. Идентификаторы предназначены для хранения уникальных идентификационных признаков. Кроме этого они могут хранить и обрабатывать конфиденциальные данные. Устройства ввода-вывода и ПО осуществляют обмен данными между идентификатором и защищаемой системой.

В электронных системах идентификационные признаки представляются в виде цифрового кода, хранящегося в памяти идентификатора. По способу обмена данными между идентификатором и устройством ввода-вывода электронные системы подразделяются на контактные (к ним относятся: iButton — information button — информационная «таблетка»; о смарт-карты — интеллектуальные карты; USB-ключи или USB-токены; (token — опознавательный признак, маркер) и

бесконтактные (RFID-идентификаторы — radio-frequency identification — радиочастотные идентификаторы; смарткарты).

Контактное считывание подразумевает непосредственное соприкосновение идентификатора с устройством ввода-вывода. Бесконтактный (дистанционный) способ обмена не требует четкого позиционирования идентификатора и устройства ввода-вывода. Чтение или запись данных происходит при поднесении идентификатора на определенное расстояние к устройству ввода-вывода. Системы на базе смарт-карт и радиочастотных идентификаторов можно отнести по времени их создания к старшему, iButton — к среднему, а USB-ключей — к младшему поколению.

При обсуждении надежности обычно рассматривают самое важное и в то же время самое слабое звено системы — идентификатор. В свою очередь, надежность идентификаторов связывают со степенью их защищенности от механических воздействий, влияния температуры, внешних электромагнитных полей, агрессивных сред, пыли, влаги, а также от атак, направленных на вскрытие чипов, хранящих секретные данные. Разработчики идентификаторов iButton обеспечивают сохранность характеристик своих изделий при механическом ударе 500g, падении с высоты 1,5 м на бетонный пол, рабочем диапазоне температур от -40 до 70°C, воздействии электромагнитных полей и атмосферы. Этому способствует герметичный стальной корпус идентификатора, сохраняющий прочность при миллионе контактов с устройством ввода-вывода. Память некоторых идентификаторов (DS1991, DS1963S) защищена от доступа. Срок эксплуатации идентификатора iButton составляет 10 лет. К недостаткам систем на базе iButton следует отнести отсутствие встроенных в идентификаторы криптографических средств, реализующих шифрование данных при их хранении и передаче в компьютер, поэтому iButton обычно используется совместно с другими системами, на которые возлагаются функции шифрования.

Конечно, по степени механической надежности радиочастотные идентификаторы, смарт-карты и USB-ключи уступают iButton. Выход из строя карты вследствие механических повреждений является не таким уж редким событием. «Узким» местом USB-ключей является и ресурс их USB-разъемов. Разработчики данных идентификаторов включают этот показатель в технические спецификации изделий. Например, для идентификаторов семейства eToken гарантированное число подключений составляет не менее 5000 раз.

Достоинство радиочастотных идентификаторов, смарт-карт и USB-ключей состоит в том, что в их состав входят защищенная энергонезависимая память и криптографический процессор, позволяющие повысить уровень защиты устройств.

Опубликовано множество работ, в которых описываются разнообразные атаки на чипы идентификаторов. Эти исследования носят как теоретический, так и практический характер. К теоретическим методам вскрытия относят, в частности, атаки Bellcore, дифференциальный анализ искажений DFA (Differential Fault Analysis) и питания DPA (Differential Power Analysis). К практическим методам можно отнести глитчинг (glitching) и физические атаки, направленные на распаковку чипа и извлечение необходимой информации.

Разработчики криптографических процессоров стремятся по мере возможности адекватно реагировать на атаки с помощью разнообразных механизмов внешней и внутренней защиты. К механизмам внешней защиты относят установку датчиков (емкостный либо оптический сенсор), покрытие чипа металлическим слоем, специальными клеями и т. д., к внутренним — шифрование шины, случайное тактирование, проведение повторных вычислений, генерирование шума. В общем, из-за стоимости аппаратных идентификаторов, они применяются в основном в бизнесе, там где требуются удобство, надёжность и высокая криптостойкость. Основных минуса всего два: их можно отнять или потерять и они могут сломаться.

Биофактор (Биометрическая аутентификация) — «то, что является частью тебя». Биометрические данные, для снятия которых, как правило, необходимы специальные программно-аппаратные средства — так называемые, биометрические сканеры, которые различаются по характеру считываемых данных. *Биометрические сканеры, основанные на статических методах:*

- Распознавание по отпечаткам пальцев. Это — самый распространенный статический метод биометрической идентификации, в основе которого лежит уникальность для каждого человека рисунка папиллярных узоров на пальцах. Изображение отпечатка пальца, полученное с помощью специального сканера, преобразуется в цифровой код (свертку) и сравнивается с ранее введенным шаблоном (эталоном) или набором шаблонов;
- Распознавание по геометрии руки. Данный статический метод построен на распознавании геометрии кисти руки, также являющейся уникальной биометрической характеристикой человека. С помощью специального устройства, позволяющего получать трехмерный образ кисти руки (некоторые производители сканируют форму нескольких пальцев), получают измерения, необходимые для получения уникальной цифровой свертки, идентифицирующей человека;
- Распознавание по радужной оболочке глаза. Этот метод распознавания основан на уникальности рисунка радужной оболочки глаза. Для реализации метода необходима камера, позволяющая получить изображение глаза человека с достаточным разрешением, и специализированное программное обеспечение, позволяющее выделить из полученного изображения рисунок радужной оболочки глаза, по которому строится цифровой код для идентификации человека.

Биометрические сканеры, основанные на динамических методах:

- Распознавание по рукописному почерку. Как правило, для этого

динамического метода идентификации человека используется его подпись (иногда написание кодового слова). Цифровой код идентификации формируется по динамическим характеристикам написания, то есть для идентификации строится свертка, в которую входит информация по графическим параметрам подписи, временным характеристикам нанесения подписи и динамике нажима на поверхность в зависимости от возможностей оборудования (графический планшет, и т. д.);

- Распознавание по клавиатурному почерку. Метод в целом аналогичен вышеописанному, однако вместо подписи в нем используется некое кодовое слово, а из оборудования требуется только стандартная клавиатура. Основная характеристика, по которой строится свертка для идентификации, — динамика набора кодового слова;
- Распознавание по голосу. В настоящее время развитие этой одной из старейших технологий ускорилось, так как предполагается ее широкое использование при сооружении интеллектуальных зданий. Существует достаточно много способов построения кода идентификации по голосу: как правило, это различные сочетания частотных и статистических характеристик последнего.

В целом для многих из перечисленных методов необходимо достаточно дорогое оборудование и не менее дорогое ПО.

Социальный фактор (Социальная аутентификация) — «те, кто тебя знают». В качестве последнего фактора можно использовать легального пользователя. Такую систему можно назвать основанной на доверенной аутентификации. Microsoft долго использовала форму восстановления учетной записи, основанную на доверенной стороне, для её собственных сотрудников: если сотрудник забывал свои учетные данные, его менеджер или коллеги могли запросить временный пароль от его имени [27]. Такой механизм, по определению, требует вмешательства другого человека, и применим, зачастую, в небольших системах малых и средних компаний, где администратор системы может выкроить время для генерации нового пароля.

В больших компаниях и корпорациях, существуют целые отделы, занимающиеся подобными проблемами, и поэтому такой способ аутентификации является в настоящее время одним из самых затратных.

Для защиты изолированной среды, в которой функционирует разработанный нами МТ, была использована двухфакторная аутентификация. Двухфакторная аутентификация (англ. two-factor authentication, также известна как двухэтапная верификация) (далее – ДФА), является типом многофакторной аутентификации и представляет собой технологию, обеспечивающую идентификацию пользователей с помощью комбинации двух различных компонентов.

Примером двухфакторной аутентификации является авторизация Google и Microsoft. Когда пользователь заходит с нового устройства, помимо аутентификации по имени-паролю, его просят ввести шестизначный (Google) или восьмизначный (Microsoft) код подтверждения. Абонент может получить его по SMS, с помощью голосового звонка на его телефон, код подтверждения может быть взят из заранее составленного реестра разовых кодов или новый одноразовый пароль может быть сгенерирован приложением-аутентификатором за короткие промежутки времени. Выбор метода осуществляется в настройках аккаунта Google или Microsoft соответственно. Пример приведен на рисунке 9.

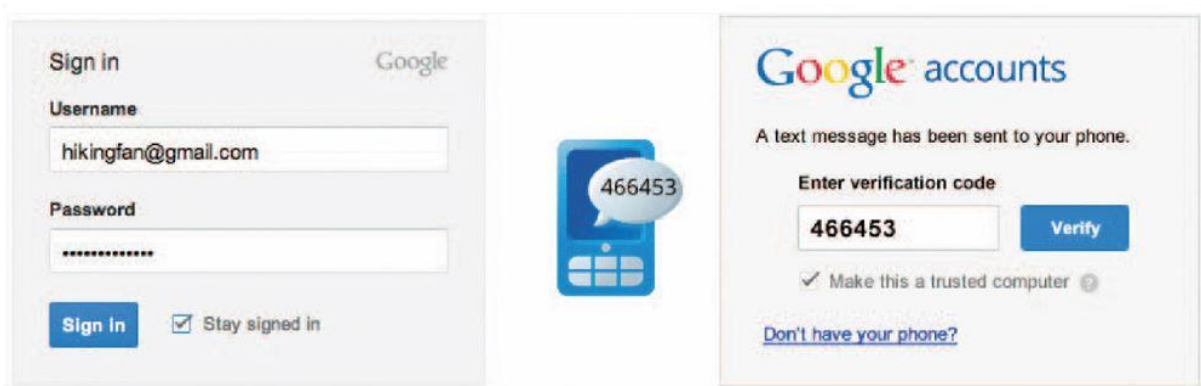


Рисунок 9 - Вход по двухфакторной аутентификации.

Для организации доступа к изолированной среде (песочницы) мы применили и доработали плагин, реализующий двухфакторную аутентификацию. Листинг программы представлен в Приложении 2

Для оценки эффективности ДФА (двухфакторной аутентификации) сначала определим некоторые базовые понятия, которые будут использоваться в этой оценке. Привычные для специалистов по информационной безопасности (ИБ) свойства защищенности информационной системы (ИС) применительно к системам идентификации, аутентификации (далее – СИА) могут быть выстроены в порядке убывания приоритетов следующим образом: доступность, целостность, конфиденциальность. В работе [13] показано, что в понятие надежности кроме указанных свойств защищенности, входят такие свойства, как безотказность, сохранность (устойчивость к воздействиям) и ремонтпригодность.

Безотказность – свойство системы (объекта) непрерывно сохранять работоспособное состояние в течение некоторого времени (наработки). Под наработкой обычно понимается продолжительность времени работы системы или объем работы. Состоянием называется множество существенных свойств, которыми объект обладает в данный момент времени. Безотказность и доступность условно можно объединить в понятие минимизации простоев СИА, т.е. обеспечение непрерывности обслуживания заявок на аутентификацию.

Важнейшим понятием в теории надежности является понятие отказа. В ИС отказы происходят не всегда одинаково, различные способы отказа называются состояниями отказа. Состояния отказа отражают события ненадлежащего обслуживания. Применительно к СИА под отказом будем понимать отрицательный результат аутентификации и соответственно состояние отказа в авторизации легального пользователя.

Работоспособное состояние – это такое состояние объекта, при котором множество существенных свойств в полном объеме отвечает заданным требованиям.

Под опасным отказом будем понимать положительный результат прохождения процесса аутентификации злоумышленником.

Для формирования дерева отказов и дерева событий введем следующие предположения, основанные на опыте проектирования, построения и анализе функционирования ряда промышленных СИА [44]:

1. Основной поток заявок $\lambda_{л.п}$ на обслуживание СИА поступает от легальных пользователей системы, при этом заявки не содержат ошибок, а система и ее элементы не имеют отказов.

2. Среди массы заявок от легальных пользователей имеется некоторая часть некорректно оформленных заявок $\lambda_{ош.л.п} \in \lambda_{л.п}$. из-за непреднамеренных ошибок.

3. Из числа заявок от легальных пользователей имеется некоторая часть заведомо ложных заявок $\lambda_{з.л.л.п} \in \lambda_{л.п}$. с целью выдать себя за пользователя с более привилегированными правами доступа. Таким образом, имеем соотношение: $\lambda_{ош.л.п} + \lambda_{ош.л.п} + \lambda_{з.л.л.п} = \lambda_{л.п}$.

4. В СИА поступает некоторая часть заведомо ложных заявок от злоумышленников $\lambda_{з.л.зл}$, пытающихся выдать себя за легальных пользователей: $\lambda_{з.л.зл} \cap \lambda_{л.п}$.

Уточним ряд положений о работе СИА:

1. СИА состоит из серверной и клиентской частей, связанных устойчивым каналом (каналами) связи.

2. Серверная часть состоит из нескольких связанных защищенным образом серверов (напри-мер, по протоколу IPSec), отказоустойчивость OU которых (по SLA – Service Level Agreement, соглашение об уровне обслуживания) $OU \geq 99,95\%$. Системное, прикладное и специальное программное обеспечение (ПО) – лицензионное, как правило, вовремя обновляется и обслуживается производителями.

3. Клиентская часть может быть представлена в виде следующих модификаций: компьютер пользователя с необходимым набором ПО и аутентификационной информацией (АИ) пользователя:

а) код доступа (логин, пароль);

б) логин и пароль плюс одноразовый пароль или усиленный неквалифицированный сертификат и ключ неквалифицированной подписи;

в) квалифицированный сертификат доступа и ключ подписи.

Для определенности будем считать, что в вариантах (б) и (в) АИ пользователя находится в некоем устройстве, связанном с конкретным пользователем. При этом связь пользователя с устройством осуществлена ЦР в виде ЭУ при выдаче пользователю АИ и находится в БД учетных записей.

Проведем анализ видов и последствий отказов согласно рекомендациям [42-46]. Как известно, этот метод позволяет определить возможные причины отказа элементов системы и события, породившие отказ.

Составим дерево отказов СИА в соответствии с пятиуровневой схемой. Верхний (первый) уровень – отказ системы. Второй уровень – отказ составных частей. Третий уровень – отказ элементов. Следующий уровень определяет события, порождающие отказ. Пятый уровень определяет виды воздействий, приводящих к отказу СИА.

Построение всего дерева отказов СИА в виде графа событий и последствий представляет собой трудночитаемый рисунок с множеством мелких значков и надписей. Поэтому выделим наиболее существенные виды отказов СИА, не пропуская, по возможности, наиболее критичные с точки зрения безопасности и надежности.

Для примера сначала рассмотрим некоторые отказы СИА, связанные с событиями, породившими отказ, в процедурах регистрации и хранения (табл.2).

Таблица 2

Примеры дерева отказов, связанных с нарушениями ИБ, в процедурах регистрации и хранения аутентификационной информации

Уровень Системы	Отказ СИА	Отказ СИА	Опасный отказ СИА (в процедуре регистрации)	Опасный отказ СИА (в процедуре хранения)
Уровень Составных Частей	Отказ в регистрации	Отказ в регистрации Легальному Пользователю	Злоумышленник зарегистрирован под видом легального Пользователя	Злоумышленник владеет ИД и секретом (аутентификатором) легального пользователя
Уровень Элементов	Отказ в приеме ИД	Отказ в результате проверки ИД	Проверки ИД не выявили обмана	Потеря конфиденциальности секрета
События, порождающие Отказ	Неполный набор представленных пользователем ИД	В базах данных ведомств не найдены ИД, соответствующие Представленным Пользователем	Поддельные документы на имя Легального Пользователя	Нарушение условий хранения секрета
Виды воздействия	Ошибка заявителя, попытка злоумышленника	Неполная база, сбой, вирусная атака	Атака класса «маскарад»	Хищение, копирование ИД и секрета (аутентификатора)

Также можно рассмотреть дерево отказов СИА, связанных с нарушениями ИБ, и для других процедур аутентификации. В табл.3 приводятся примеры отказов, обусловленных нарушениями ИБ, в процедурах валидации, протоколах обмена и принятия решения.

Таблица 3

Примеры дерева отказов, связанных с нарушениями ИБ, в процедурах проверки валидности ЭУ, протоколах аутентификации и принятия решения

Уровень Системы	Отказ СИА	Отказ СИА	Отказ СИА	Отказ СИА
Уровень составных частей СИА	Отказ в валидации	Отказ в работе протокола обмена	Отказ в процедуре принятия решения	Отказ в процедуре принятия решения
Отказ элементов	ЭУ пользователя не валидно	Отказ в клиентской части	Несовпадение Предъявленного секрета с БД	Превышено время ожидания
События, порождающие Отказ	Нет цепочки проверки сертификата, не работает служба OCSP DVCS	Не установлен драйвер, не выполнено обновление системного ПО	Подмена ИД и ЭУ на имя легального Пользователя	Велика интенсивность потока заявок на аутентификацию для данной СИА, ошибки проектирования
Виды воздействия	Атака не сервер УЦ, выдавший ЭУ, сбой УЦ цепочки ЭУ	Вирусная атака, халатность администратора	Попытка Злоумышленника	DDoS-атака

Сформированное таким образом дерево отказов позволяет более четко идентифицировать вероятные события, которые могут привести к нарушениям ИБ при работе СИА. Рассмотрение отказов является одной из важных подготовительных процедур для идентификации рисков нарушения безопасности функционирования СИА. Следующей процедурой, согласно [46], является формирование модели дерева событий, где необходимо выделить наиболее вероятные опасные события и оценить частоту их реализации.

Существует вероятность ошибки первого рода (СИА не авторизовала легального пользователя ИС). Рассмотрим возможные причины такого события: 1) пользователь неверно ввел свою АИ (например, забыл пароль в случае «а»); рисков процессов аутентификации;

2) перегрузка СИА ввиду большого числа одновременных заявок и/или время ожидания превысило некий порог ожидания;

3) отказ клиентской части (аппаратный или программный сбой);

4) отказ канала связи (аппаратный и/или программный);

5) отказ серверной части.

Также существует вероятность ошибки второго рода, когда СИА признала ИА правильной и авторизовала злоумышленника под именем легального пользователя.

На основе анализа опыта построения и эксплуатации ряда промышленных СИА практикующих специалистов выделим ряд вероятных опасных событий [45] и оценим влияние ДФА.

$RNE_i, i = 1, n$. Перечислим эти события и приведем грубую оценку частоты их реализации для двух состояний ИС: без применения двухфакторной аутентификации и после реализации таковой.

RNE_1 . Целенаправленные действия злоумышленника при регистрации. Регистрация – одна из самых ответственных операций процессов аутентификации, существенно влияющая на безопасность, надежность и в

конечном счете на доверие работы СИА. Данную угрозу обозначают как «маскарад» при регистрации. Средняя частота такого события для государственных и корпоративных СИА по оценке [45] располагается в достаточно широких пределах: 10^{-7} – 10^{-5} в год, поскольку двухфакторная аутентификация является средством усиленной аутентификации прием нижний предел.

РНЕ2. Злоумышленник для доступа к интересующим его информационным ресурсам может воспользоваться уязвимостями СИА. Это опасное событие имеет вероятность осуществиться. Будем называть это событие «уязвимости СИА» и оценим частоту в пределах 10^{-5} – 10^{-3} . ДФА напрямую на аннигиляцию данной угрозы не влияет.

РНЕ3. Этот тип вероятного опасного события (ВОС) может быть связан с действиями инсайдера. Помочь злоумышленнику пройти все рубежи СИА может легальный пользователь. Еще больше возможностей у администратора. Кратко назовем это событие «помощь инсайдера». Средние оценки частоты: 10^{-6} – 10^{-4} . При применении ДФА частота снижается до нижнего предела.

РНЕ4. Завладение злоумышленником ИД и АИ легального пользователя. Это может быть кража, клонирование ИД и АИ, подсмотренный пароль, перехваченный PIN-код. Кратко назовем этот тип «кража ИД и АИ» и оценим частоту: 10^{-5} – 10^{-3} . ДФА напрямую на данную угрозу не влияет.

РНЕ5. Атака «вход по принуждению» встречается все реже и реже: 10^{-7} – 10^{-5} . Применение ДФА в этом случае снижает риск появления до 10^{-7}

РНЕ6. Ошибки и/или целенаправленные действия злоумышленника при смене пароля, замене цифрового сертификата доступа или сценарии «забыл дома смарт-карту». Коротко назовем этот тип «смена АИ» и оценим частоту в пределах 10^{-5} – 10^{-3} . Применение ДФА в этом случае снижает риск появления до 10^{-5}

РНЕ7. Данный тип ВОС связан с ошибками валидации ЭУ. Под валидацией будем понимать процесс проверки действительности сертификата доступа и цепочки сертификатов, для парольной защиты это процедура сличения хешей паролей (присланного претендентом и зарегистрированного в БД учетных записей). Короткое название – «ошибки валидации». Оценки частоты: 10^{-6} – 10^{-4} . ДФА напрямую на данную угрозу не влияет.

РНЕ8. Ошибки в принятии решения «свой–чужой». Процедура производится на серверах, вероятная частота подобного события 10^{-7} – 10^{-5} . Применение ДФА в этом случае снижает риск появления до 10^{-7}

РНЕ9. Имитация доверяющей стороны. Особенно актуален такой тип ВОС при предоставлении Web –доступа, который становится все более распространенным. Фишинг (подмена сайта) является одним из актуальных ВОС, оценки частоты колеблются в пределах 10^{-4} – 10^{-2} . Применение ДФА в этом случае снижает риск появления до 10^{-4}

РНЕ10. Подмена доверенной стороны или объекта (spoofing), оценим частоту 10^{-6} – 10^{-4} . Применение ДФА в этом случае снижает риск появления до 10^{-6}

РНЕ11. Риск добровольной передачи персонального средства ИА другому пользователю. Частоту можно оценить в пределах 10^{-4} – 10^{-2} . Средство борьбы – усиленная персонализация, в нашем случае двухфакторная аутентификация позволяет снизить частоту до 10^{-6} – 10^{-4} .

РНЕ12. Наконец, последним ВОС будем считать воздействие вредоносного программного обеспечения, вероятность заражения рабочих мест определяется политикой безопасности организации, в среднем может быть оценена как 10^{-4} – 10^{-2} . ДФА напрямую на данную угрозу не влияет.

Таким образом, анализируя влияние ДФА на вероятность реализации рисков ИБ в изолированной среде с ДФА – доступом, можем отметить, что вероятность их появления снижается в 9 случаях из рассмотренных 12 типовых, что свидетельствует о повышении уровня защищенности

информационных ресурсов. Технология песочниц является испытанной и надежной методикой выполнения рискованных приложений, усиление ее защиты двухфакторной аутентификацией способствует повышению информационной безопасности ИС и защищенности информационных ресурсов образовательной организации.

Проведенное исследование подтвердило выдвинутую гипотезу.

Выводы по 3 главе

Мы реализовали изолированную среду - песочницу, используя плагин CodeRunner (V3.3.0) для Moodle и специально выделенный сервер. Песочница (англ. sandbox) - это среда для безопасного исполнения компьютерных программ, представляет собой жёстко контролируемый набор ресурсов для исполнения гостевой программы. Песочницы являются реализацией виртуализации. CodeRunner - плагин для Moodle, который позволяет создавать виртуальные изолированные среды. Многопользовательский тренажер по языку программированию Python мы реализовали в изолированной среде – песочнице.

Для дополнительной защиты нами была апробирована двухфакторная аутентификация, для чего был доработан плагин для Moodle. Экспериментальная проверка доказала работоспособность спроектированной информационной системы: песочницы, многопользовательского тренажера и плагина двухфакторной аутентификации.

Анализ влияния двухфакторной аутентификации на общую защищенность изолированной среды показал повышение уровня безопасности вышеописанной информационной системы. Методика оценки влияния была основана на рисках вероятных опасных событий и аннигилирующего влияния двухфакторной аутентификации в этих опасных событиях.

Состояние информационной системы при применении двухфакторной аутентификации более защищено по 9 позициям опасных событий из 12, что доказывает ранее выдвинутую гипотезу.

Заключение

В ходе проведенного исследования были решены задачи, поставленные в начале исследования.

Была раскрыта специфика многопользовательских тренажеров как вида электронных образовательных ресурсов, проведен обзор сред, пригодных для разработки многопользовательских тренажеров, выявлена структура, содержание многопользовательских тренажеров.

Многопользовательский электронный тренажер - это программно-методический комплекс, в основу которого положена система диагностики и интерпретации полученных ответов, алгоритмы целенаправленной тренировки обучающегося в процессе многократного повторного выполнения имитации технологического процесса, а также тестовых заданий разного формата. Существенным отличием электронного тренажера от электронного учебного пособия является наличие сценария. Для организации многопользовательского режима тестирования необходимо введение дополнительных компонентов, таких как база данных и сервер базы данных, а также программного обеспечения для функционирования этой информационной системы.

На базе Moodle был разработан проект многопользовательского тренажера для раздела профессионального модуля по программированию.

На базе исследования – ГПБОУ «ЮУРГТК» сделан анализ защищенности информационных ресурсов и выявлены подходы к совершенствованию информационной безопасности образовательной организации. Процедура оценки, состоящая в проведении опросов на основе требований стандарта по учету усредненного экспертного мнения специалистов, показал наличие таких угроз, как возможность несанкционированного внесения изменений в персональную информацию студентов при проведении тестирования, а именно – в результаты учебной успеваемости, а также возможность внесения корректив в программные коды и настройки операционной системы.

При внедрении многопользовательского тренажера имеющиеся уязвимости очевидно усилятся, в связи с чем, нами были предложены меры информационной защиты и проведена экспериментальная проверка их эффективности. Для этого мы реализовали изолированную среду - песочницу, используя плагин CodeRunner (V3.3.0) для Moodle и специально выделенный сервер. CodeRunner - плагин для Moodle. Многопользовательский тренажер по языку программированию Python мы реализовали в данной изолированной среде.

Для дополнительной защиты нами была апробирована двухфакторная аутентификация, для чего был доработан плагин для Moodle. Экспериментальная проверка доказала работоспособность спроектированной информационной системы: песочницы, многопользовательского тренажера и плагина двухфакторной аутентификации.

Анализ влияния двухфакторной аутентификации на общую защищенность изолированной среды показал повышение уровня безопасности вышеописанной информационной системы. Методика оценки влияния была основана на рисках вероятных опасных событий и аннигилирующего влияния двухфакторной аутентификации в этих опасных событиях.

Таким образом, задачи исследования решены, гипотеза доказана, цель исследования достигнута.

Библиографический список

1. ГОСТ Р 51901.1-2002. Менеджмент риска. Анализ риска технологических систем. [Электронный ресурс]. – Режим доступа: <http://docs.cntd.ru/document/1200030153> свободный, дата проверки: 17.12.2013г.
2. ГОСТ Р 51901.12-2007. Метод анализа видов и последовательность отказов [Электронный ресурс]. – Режим доступа: http://www.opengost.ru/iso/13_gosty_iso/13110_gost_iso/4936-gost-r-51901.12-2007-mek-60812_2006-menedzhment-riska.-metod-analiza-vidov-i-posledstviy-otkazov.html свободный, дата проверки: 17.12.2013 г.
3. ГОСТ Р 53620 - 2009 «Информационно-коммуникационные технологии в образовании. Электронные образовательные ресурсы. Общие положения».
4. ГОСТ Р 53625-2009 (ИСО/МЭК 19796-1:2005) Информационная технология. Обучение, образование и подготовка. Менеджмент качества, обеспечение качества и метрики. Часть 1. Общий подход
5. ГОСТ Р ИСО/МЭК 17799–2005. Информационная технология. Практические правила управления информационной безопасностью.
6. ГОСТ Р ИСО/МЭК 27001 «Информационная технология. Методы и средства обеспечения безопасности. Системы менеджмента информационной безопасности. Требования».
7. ГОСТ Р ИСО/МЭК 9594-8-98 «Информационная технология (ИТ). Взаимосвязь открытых систем. Справочник. Часть 8. Основы аутентификации».
8. Федеральный государственный образовательный стандарт среднего профессионального образования по специальности 09.02.03 Программирование в компьютерных системах (утв. приказом Министерства образования и науки РФ от 28 июля 2014 г. № 804

9. Федеральный государственный образовательный стандарт среднего профессионального образования по специальности 09.02.04 Информационные системы (по отраслям) (утв. приказом Министерства образования и науки РФ от 14 мая 2014 г. № 525

10. Федеральный государственный образовательный стандарт среднего профессионального образования по специальности 09.02.05 Прикладная информатика (по отраслям) (утв. приказом Министерства образования и науки РФ от 13 августа 2014 г. N 1001

11. Абдулин А.А., Гафарова Е.А. Направления обеспечения информационной безопасности электронного документооборота в ГБОУ СПО (ССУЗ) «Челябинский профессиональный колледж» // Информационные технологии Сибири сборник материалов международной научно-практической конференции. Западно-Сибирский научный центр. 2016. С. 87-90.

12. Аутентификация. Теория и практика. Обеспечение безопасного доступа к информационным ресурсам / А.А. Афанасьев и др.; под ред. А.А. Шелупанова, С.Л. Груздева, Ю.С. Нахаева. – М.: Горячая линия-Телеком, 2009. – 552 с.

13. Баранова А. В., Ямпурин Н. П. Методы оценки надежности информационных систем // НиКа. 2014. №. URL: <https://cyberleninka.ru/article/n/metody-otsenki-nadezhnosti-informatsionnyh-sistem> (дата обращения: 19.01.2019).

14. Белевитин В.А. Гафарова Е.А., Корчемкина Ю.В., Шварцкоп О.Н. Влияние тернарности представления учебной информации на повышение креативности обучающихся [Текст] // European Social Science Journal. 2017. № 6. С. 194-200.

15. Белов, Е.Б. Основы информационной безопасности. // Учебное пособие для вузов / Е.Б. Белов, В.П. Лось, Р.В. Мещеряков, А.А. Шелупанов. – М.: Горячая линия – Телеком, 2006. – 544 с.

16. Бойко А.А., Бердник А.А. Методы защиты виртуальной среды. Программный комплекс для проведения автоматизированного аудита виртуальной среды на предмет наличия ошибок в конфигурации безопасности // Всероссийский журнал научных публикаций. 2013. №3 (18). URL: <https://cyberleninka.ru/article/n/metody-zaschity-virtualnoy-sredy-programmnyu-kompleks-dlya-provedeniya-avtomatizirovannogo-audita-virtualnoy-sredy-na-predmet> - (дата обращения: 09.01.2019)]

17. Гафарова Е.А. О возможности использования открытых лицензий для защиты интеллектуальных прав создателей научных и образовательных ресурсов.// Современное развитие науки: вопросы теории и практики Сборник материалов II-ой международной научно-практической конференции. 2016. С. 46-47.

18. Гафарова Е.А. Сеницын Ф.В. К вопросу проектирования онтологий предметной области при подготовке магистров по направлению информационная безопасность.//Иновационные технологии в подготовке современных профессиональных кадров: опыт, проблемы. Сборник научных трудов. 2016. С. 56-59.

19. Дровникова И.Г. Роль и место современных компьютерных технологий обучения в совершенствовании управления подготовкой специалистов для системы безопасности [Электронный ресурс]// URL: И.Г. Дровникова, Т.А. Буцынская, П.А. Орлов // Вестник Воронежского института МВД России - № 3, 2008 <http://cyberleninka.ru/article/n/rol-i-mesto-sovremennyh-kompyuternyh-tehnologiy-obucheniya-v-sovershenstvovanii-upravleniya-podgotovkoj-spetsialistov-dlya-sistemy> (дата обращения 04.12.2018).

20. Жукова Н.В. Возможности использования электронных тест – тренажеров при обучении физической химии // Фундаментальные исследования. – 2013. – № 10-12. – С. 2778-2781; URL: <http://www.fundamental-research.ru/ru/article/view?id=32871> (дата обращения: 02.12.2018).

21. Заводчикова Н.И., Плясунова У.В., Суворова М.А. Использование системы дистанционного обучения Moodle для организации самостоятельной работы студентов дневной формы обучения//Вестник Костромского государственного университета. Серия: Педагогика. Психология. Социокинетика. 2016. №4. [Электронный ресурс]//URL: <https://cyberleninka.ru/article/n/ispolzovanie-sistemy-distantsionnogo-obucheniya-moodle-dlya-organizatsii-samostoyatelnoy-raboty-studentov-dnevnoy-formy-obucheniya> (дата обращения: 06.12.2018).

22. Институт ЮНЕСКО по информационным технологиям в образовании: Основы разработки электронных образовательных ресурсов [Электронный ресурс] // URL: <https://www.intuit.ru/academies/companiesn/45/info> - (дата обращения: 06.12.2018).

23. Интернет-тренажеры [Электронный ресурс]//URL: <http://training.i-exam.ru/> - (дата обращения 03.10.2018).

24. Интернет-тренажеры в итоговой аттестации[Электронный ресурс] // <http://training.i-exam.ru/23456> (дата обращения 03.11.2018).

25. Капустин Ф.А. Информационная безопасность и защита информации в современном обществе // Актуальные проблемы авиации и космонавтики. 2016. №12. URL: <https://cyberleninka.ru/article/n/informatsionnaya-bezopasnost-i-zaschita-informatsii-v-sovremennom-obschestve> (дата обращения: 03.01.2019).

26. Конявский В.А. Методы и аппаратные средства защиты информационных технологий электронного документооборота : Дис. ... д-ра техн. наук : 05.13.19 : М., 2005 360 с. РГБ ОД, 71:05-5/526

27. Корнев В.М., Проскуряков А.В. Обеспечение комплексной безопасности образовательной организации [Электронный ресурс] // Актуальные проблемы авиации и космонавтики. 2017. №13. URL: <https://cyberleninka.ru/article/n/obespechenie-kompleksnoy-bezopasnosti-obrazovatelnoy-organizatsii> - (дата обращения: 10.01.2019).

28. Корчемкина Ю.В. Гафарова Е.А., Белоусова Н.А., Мальцев В.П. Применение информационных технологий для повышения эффективности и планирования образовательной траектории обучения математике студентов [Текст]// Современные наукоемкие технологии. 2017. № 7. С. 114-118.

29. Корчемкина Ю.В., Гафарова Е.А. Модель методической системы обучения линейной алгебре студентов экономико-информационных направлений подготовки [Текст].// Вестник Челябинского государственного педагогического университета. 2017. № 8. С. 41-48.

30. Кузнецов Н.А., Микрин Е.А., Кульба В.В. Информационная безопасность систем организационного управления. Теоретические основы. Издательство: Наука, 2006, 424 с.

31. Мальков Михаил Васильевич О надежности информационных систем // Труды Кольского научного центра РАН. 2012. №4. URL: <https://cyberleninka.ru/article/n/o-nadezhnosti-informatsionnyh-sistem> (дата обращения: 19.01.2019).

32. Мезенов А.С., Гафарова Е.А. О реализации конституционного права граждан на доступ к информации в условиях интенсификации сетевого взаимодействия/ / А.С. Мезенов, Е.А. Гафарова // Инновационные технологии в подготовке современных профессиональных кадров: опыт, проблемы : сборник научных трудов. — Челябинск: Челябинский филиал РАНХиГС, 2016. — С. 94–98. — 200 с. — ISBN: 978-591970-052-4.

33. Насс О.В. Формирование компетентности педагогов в проектировании электронных образовательных ресурсов в контексте обновления общего среднего и высшего образования: монография. М.: Изд-во МПГУ, 2010.

34. Основные направления научных исследований в области обеспечения информационной безопасности Российской Федерации. [Электронный ресурс] // URL: <http://www.scrf.gov.ru/security/information/document155/> - (дата обращения: 10.01.2019).

35. Плаксин М.А., Халимова М.Р., Хамадеева Д.О. Компьютерный тренажер для освоения основ финансовой грамотности. [Текст] // Вестник Пермского государственного университета (ПГУ), г. Пермь], 2016, с. 53-58
36. Полат Е.С. Телекоммуникации в системе образования. // ИНФО, М.: Информатика и образование - 1988, №5. С.110-113.
37. Поляк В.Е. Компьютерные тренажеры и интерактивные электронные технические руководства: как их использовать в учебном процессе? [Электронный ресурс]//URL: nito.rsvpu.ru/files/nito2013/presentations/Поляк.pps (дата обращения 07.12.2018).
38. Распоряжение Правительства Российской Федерации от 20 октября 2010 г. № 1815-р «О государственной программе Российской Федерации «Информационное общество (2011–2020 годы)» [Электронный ресурс]. – Режим доступа: <http://www.rg.ru/2010/11/16/infobschestvo-site-dok.html> свободный, дата проверки: 17.12.2018 г.
39. Роберт И.В. Современные информационные технологии в образовании: дидактические проблемы; перспективы использования. // М.: Школа-Пресс, - 1994. 205 с.
40. Роберт И.В. Экспертно-аналитическая оценка качества программных средств учебного назначения. // Педагогическая информатика. М., - 1993. №1. С.54-62.
41. Руководящий документ «Автоматизированные системы. Защита от несанкционированного доступа к информации. Классификация автоматизированных систем и требования по защите информации. Утверждено решением председателя Государственной технической комиссии при Президенте Российской Федерации от 30 марта 1992 г.
42. Сабанов А.Г. Аутентификация при электронном обмене документами // Доклады Том. гос. ун-та систем управления и радиоэлектроники. – 2011. – № 2(24). – С. 263–266.

43. Сабанов А.Г. Концепция моделирования процессов аутентификации // Доклады ТУСУРа. –2013. – № 3(29). – С. 71–75.
44. Сабанов А.Г. Методы исследования надежности удаленной аутентификации // Электросвязь. –2013. – № 4. – С. 20–24.
45. Сабанов А.Г. Об оценке рисков удаленной аутентификации как процесса // Электросвязь. –2012. – № 4. – С. 27–32.
46. Сабанов А.Г. Основные процессы аутентификации // Вопросы защиты информации. – 2012. –3. – С. 54–57.
47. Савин И.А., Батенькина О.В. Написание скриптов для трехмерного графического движка/Визуальная культура: дизайн, реклама, информационные технологии: сборник трудов XIII Международной научно-практической конференции. 2014. С. 91-95.
48. Силаков Д. В. Использование аппаратной виртуализации в контексте информационной безопасности // Труды ИСП РАН. 2011. №. URL: <https://cyberleninka.ru/article/n/ispolzovanie-apparatnoy-virtualizatsii-v-kontekste-informatsionnoy-bezopasnosti> (дата обращения: 09.01.2019)
49. Тараскина Я.В. Дидактические функции MOODLE в иноязычном вузовском образовании [Электронный ресурс] // Современные проблемы науки и образования.–2015.–№1-1 //URL: <http://www.science-education.ru/ru/article/view?id=19146> (дата обращения: 06.12.2018).
50. Хабрахабр [Электронный ресурс]//URL: <https://habrahabr.ru/post/158479/> - (дата обращения 03.10.2018).
51. Харитонов В.Ю., Бажин В.А., Рудельсон Л.Е. Компьютерное воспроизведение виртуальной реальности в современных авиационных тренажерах // Научный вестник МГТУ ГА. 2011. №171. URL: <https://cyberleninka.ru/article/n/kompyuternoe-voisproizvedenie-virtualnoy-realnosti-v-sovremennyh-aviatsionnyh-trenazherah> (дата обращения: 14.12.2018).
52. Харитонов В.Ю., Рудельсон Л.Е. Симуляция реальности в тренажерах при подготовке летчиков современных авиационных

[Электронный ресурс] // Научный вестник МГТУ ГА. 2011. №171. URL: <https://cyberleninka.ru/article/n/kompyuternoe-voisproizvedenie-virtualnoy-realnosti-v-sovremennyh-aviatsionnyh-trenazherah> (дата обращения: 14.11.2018).

53. Хеннер, Е. К. Формирование ИКТ-компетентности учащихся и преподавателей в системе непрерывного образования [Текст] // Е.К. Хеннер. - 2-е изд. (эл.). - М. : БИНОМ. Лаборатория знаний, 2012. - 188 с. : ил. - ISBN 978-5-9963-0883-5

54. Шишкин В. В., Гераськина С. Т., Шишкина О. Ю. Трехмерное моделирование в среде BLENDER / графика, виды графики, компьютерная графика, техника компьютерной графики. Ульяновск, 2010. С. 20–32.

55. Шубинский И.Б. Функциональная надежность информационных систем. Методы анализа. – Ульяновск: Печатный двор, 2012. – 296 с.

56. Юхин Е.Г., Кошелев Н.А., Хафизов А.М., Малышева О.С. Разработка виртуального тренажера – имитатора работы трубчатой печи для повышения профессиональных навыков сотрудников предприятий нефтегазовой отрасли. [Электронный ресурс] // Фундаментальные исследования. – 2015. – № 12-5. – С. 970-974; URL: <http://www.fundamental-research.ru/ru/article/view?id=39661> - (дата обращения: 14.11.2018)]

57. [Электронный ресурс] // URL: <http://www.your-study.ru/About> - (дата обращения: 02.12.2018).

58. [Электронный ресурс] // URL: <https://moodle.org> - (дата обращения: 02.12.2018).

59. [Электронный ресурс] // URL: <https://olat.org/> - (дата обращения: 02.12.2018)

60. [Электронный ресурс] // URL: <https://www.opigno.org/en> - (дата обращения: 02.12.2018).

61. [Электронный ресурс]//URL: <https://ffin.ru/market/review/82/50835/>- (дата обращения 03.10.2018).

62. [Электронный ресурс]: Положение об электронных образовательных ресурсах Краевого ГБПОУ «Профессиональное образовательное учреждение немецкого национального района»// URL:http://www.proflizei.ru/Akty/p_ob_ehlektronnykh_obraz_resursakh.pdf. - (дата обращения: 02.12.2018).

63. [Электронный ресурс]: Положение об электронных образовательных ресурсах ГОУ ВПО «Дагестанский государственный университет» //URL: <http://www.ndoc.dgu.ru/PDFF/8.01.pdf> - (дата обращения: 02.12.2018).

64. [Электронный ресурс]: Положение об электронных образовательных ресурсах ФГАОУ ВПО «Сибирский федеральный университет»//: URL: <http://about.sfu-kras.ru/docs/8733/pdf/413360> - (дата обращения: 02.12.2018).