

УДК 004.422

DOI: 10.25559/SITITO.14.201802.344-354

ВЫБОР ТЕХНОЛОГИЧЕСКИХ РЕШЕНИЙ ДЛЯ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ РАСПРЕДЕЛЕННЫХ ИНФОРМАЦИОННЫХ СИСТЕМ

Д.Ю. Ильин^{1,2}, Е.В. Никульчев^{1,2}, П.В. Колясников²¹ МИРЭА – Российский технологический университет, г. Москва, Россия² Российская академия образования, г. Москва, Россия

THE CHOICE OF TECHNOLOGICAL SOLUTIONS FOR SOFTWARE DEVELOPMENT FOR DISTRIBUTED INFORMATION SYSTEMS

Dmitry Yu. Ilyin^{1,2}, Evgeny V. Nikulchev^{1,2}, Pavel V. Kolyasnikov²¹ MIREA - Russian Technological University, Moscow, Russia² Russian Academy of Education, Moscow, Russia

© Ильин Д.Ю., Никульчев Е.В., Колясников П.В., 2018

Ключевые слова

Технологии разработки программ; цифровая психологическая платформа; информационная поддержка психологических исследований; выбор компонентов программ; проектирование информационных систем; программная инженерия.

Аннотация

Выбор инструментов и средств и технологий программирования для создания информационных систем представляет собой актуальное направление. Для каждой проектируемой системы необходимо определить набор критериев, предъявляемым к среде разработки, используемым библиотекам и технологиям. В статье рассматривается выбор технологических решений на примере разрабатываемой цифровой вычислительной веб-платформы Российской академии образования для обеспечения информационной поддержки деятельности психологов по проведению исследований, в том числе популяционных и лонгитюдных. Особенностью системы являются: масштабность и значительное время разработки, требующей реализации и обеспечение гарантированной вычислительной надежности широкого спектра применяемых цифровых инструментов психологических исследований; обеспечение функционирования в различных условиях, в частности при проведении массовых исследований в школах, имеющих различные характеристики вычислительных ресурсов и каналов связи; возможность масштабирования сервисов; обеспечение безопасности и закрытость данных, а также использование технологий и инструментов программирования, которые бы обеспечивали совместимость и конвертацию данных с другими средствами обработки психологических исследований. Для разрабатываемой системы были введены критерии оценки программных технологий, которые учитывают особенности функционирования и жизненного цикла продукта, на конкретном примере показан выбор соответствующих технологических решений. Важной составляющей при разработке является обучение команды разработчиков используемым технологиям. Приведена система, реализующая программу обучения выбранных технологий. Разработанная система обучения используется для уменьшения времени включения в программную разработку новых исполнителей, по мере расширения задач. Ключевые слова

Теоретическая механика; провисающая цепь; цепная функция; производная; интеграл; потенциальная энергия; кинетическая энергия; принцип Лагранжа-Дирихле; закон Ньютона; оптимизации с ограничениями; система алгебраических уравнений; компьютерная графика; анимация; Mathcad.

Об авторах:

Ильин Дмитрий Юрьевич, аспирант, МИРЭА — Российский технологический университет (119454, Россия, г. Москва, пр. Вернадского, д. 78); главный аналитик лаборатории мониторинга эффективности и качества научных исследований информационно-аналитического центра, Российская академия образования (119121, Россия, г. Москва, ул. Погодинская, д. 8), ORCID: <http://orcid.org/0000-0002-0241-2733>, [i@dmityriilin.com](mailto:dmityriilin.com)

Никульчев Евгений Витальевич, доктор технических наук, профессор, профессор кафедры управления и моделирования систем, МИРЭА — Российский технологический университет (119454, Россия, г. Москва, пр. Вернадского, д. 78); главный аналитик лаборатории мониторинга эффективности и качества научных исследований информационно-аналитического центра, Российская академия образования (119121, Россия, г. Москва, ул. Погодинская, д. 8), ORCID: <http://orcid.org/0000-0003-1254-9132>, nikulchev@mail.ru

Колясников Павел Владимирович, ведущий аналитик лаборатории мониторинга эффективности и качества научных исследований информационно-аналитического центра, Российская академия образования (119121, Россия, г. Москва, ул. Погодинская, д. 8), ORCID: <http://orcid.org/0000-0003-3633-5913>, pavelkolyasnikov@gmail.com



Keywords

Technology of program development; digital psychological platform; information support of psychological research; selection of program components; design of information systems; software engineering.

Abstract

The choice of tools and programming technologies for creating information systems is a current trend. Each projected system needs you to define a set of criteria for the development environment used by libraries and technologies. This article considers a choice of technological solutions using an example of the developed digital computing web platform of the Russian Academy of Education to provide information support to the activities of psychologists doing research, including population and longitudinal studies. The peculiarity of the system are: magnitude and significant development time requiring the implementation and assurance of the guaranteed computing reliability of a wide range of digital tools used in psychological research; maintenance of functioning in various conditions, in particular when carrying out mass research in schools with different characteristics of computing resources and communication channels; the possibility of scaling services; security and data privacy, as well as the use of technologies and programming tools that would ensure the compatibility and conversion of data with other means of processing psychological research. For the developed system, criteria were introduced for evaluating software technologies that take into account the features of the functioning and life cycle of a product, a specific example shows the selection of appropriate technological solutions. An important component in the development is the training of the development team for the technologies used. A system that implements the training program for selected technologies is given. The developed training system is used to reduce the time of inclusion in the development of new software developers, as the tasks are expanded.

1. Введение

В настоящее время для разработки программных систем можно использовать большой набор инструментов, отличающихся и своими функциональными возможностями и удобствами программирования, также и не лишенных недостатков, которые часто возникают только на стадии разработки, расширения системы или при интеграции модулей в единый комплекс. Поэтому важной задачей уже на этапе проектирования выбрать набор инструментов и технологий программирования, отвечающей как требованиям самого изготавливаемого программного продукта, так и процесса программирования. При этом необходимо рассматривать и параметры используемых технологий [1], гарантированное качество обработки данных при использовании сетей [2], надежность при расширении [3] и другие.

Цель статьи — рассмотреть выбор технологических решений на примере разрабатываемой цифровой вычислительной веб-платформы Российской академии образования [4, 5] для обеспечения информационной поддержки деятельности психологов по проведению исследований, в том числе популяционных и лонгитюдных [6, 7].

Архитектура информационной системы является не только структурной основой для описания компонентов системы и их связей, но и определяет подходы к разработке и окружения. Описание архитектуры должно включать в себя ответы на вопросы, которые возникли в ходе проектирования системы.

Разрабатываемая платформа является веб-ориентированной и будет состоять из серверной и клиентской части, поэтому одно из основных задач является выбор языков и технологий, подходящих для разработки этих компонентов [8].

Платформа должна работать в большинстве браузеров, в том числе и мобильных, без установки дополнительных плагинов и расширений. Поэтому необходимо было выбрать решение, которое не будет накладывать какие-либо специфические ограничения или требовать от пользователя установки дополнительных плагинов и библиотек, кроме наличия самого браузера.

Требования к серверным компонентам менее строгие. Тем не менее, следует учитывать особенности тех или иных техноло-

гических решений в отношении кривой обучения и сложности поддержки результирующего программного продукта.

Таким образом, особенностью системы являются:

- значительное время разработки, требующей реализации и обеспечение гарантированной вычислительной надежности широкого спектра применяемых цифровых инструментов психологических исследований;
- обеспечение функционирования в различных условиях, в частности, при проведении массовых исследований в различных регионах страны, имеющих различные характеристики вычислительных ресурсов и каналов связи;
- возможность масштабирования сервисов;
- обеспечение безопасности и закрытость данных;
- использование технологий и инструментов программирования, которые бы обеспечивали совместимость и конвертацию данных с другими средствами обработки психологических исследований.

Для формирования адекватного задаче архитектурного решения применен ряд методов. Первичный этап включал в себя architecture requirement analysis (анализ требований архитектуры), с целью выявления основных вариантов использования, функциональных и нефункциональных требований к платформе [9].

На основании полученной информации был проведен architecture synthesis (синтез архитектуры) с целью определить набор слабосвязанных компонентов системы, их связи, наиболее эффективные способы обмена данными.

Для выбора языков программирования и технологий, подходящих для разработки платформы, было проведено их исследование и сравнение. Оно проводилось в контексте сформированной архитектуры, требований и имеющихся ограничений. В отношении языков программирования для браузерных приложений оцениваются возможности доставки приложений без необходимости установки дополнительного ПО. Фреймворки рассматриваются на предмет их активного применения в проектах, размера сообщества разработчиков, соответствия поставленной



задаче и времени существования на рынке. Стоит отметить, что прямое сравнение фреймворков для разработки не даст результатов, поскольку каждый из них позволит достичь конечного результата. Тем не менее, ряд из них следует считать более пригодными ввиду лучшей масштабируемости, меньших затрат на обучение и большего числа готовых модулей.

Статья состоит из трех разделов. В первом, в теоретической части изложены особенности архитектуры разрабатываемой системы и ее основные компоненты. Во втором, экспериментальном, введены критерии для клиентской и серверной части приложения, на конкретном примере показан выбор соответствующих технологических решений. В части реализации приведена система, реализующая программу обучения выбранным технологиям.

2. Теоретическая часть: методы и объект исследования

С точки зрения конечного пользователя проект будет состоять из двух основных компонентов: частной учетной записи исследователя и приложений для онлайн автономных (оффлайн) экспериментов. Учитывая, что с течением времени количество пользователей будет расти, то веб-сервис должен быть масштабируемым по горизонтали. Каждый узел API должен включать многоуровневую архитектуру. В сочетании с Object-DocumentMapper (ODM) он обеспечит обслуживание большей гибкостью по сравнению с монолитной архитектурой.

Экспериментальные и интенсивные алгоритмы данных для популяционных исследований должны быть отделены от основного сервиса, а также от административных функций. С точки зрения безопасности панель администрирования может использоваться как отдельная сервис в интрасети. Помимо масштабирования алгоритмической части, хранение данных также должно быть масштабируемым [8, 9]. Наилучшим подходом для проекта является сочетание шардирования (sharding) и репликации (replication) (рисунок 1.5). Шардирование (sharding) позволяет распределить данные между разными физическими серверами (шардами) на основе значения некоторого ключа, благодаря чему сущности группируются в набор данных по этому ключу. Репликация (replication) позволяет копировать данные между несколькими серверами, среди которых один сервер (master) – для сохранения данных, а другие (slave) – для считывания. Тем самым, шардирование может обеспечить систему высокой производительностью ввода-вывода, в то время как репликация может помочь обеспечить доступность службы.

Архитектура разрабатываемой платформы для психологических исследований была выбрана многокомпонентной, что обеспечивает больше гибкости по сравнению с монолитной.

Монолитные архитектуры имеют ряд недостатков:

- чем больше система, тем сложнее поддерживать ее и вносить изменения;
- при большой системе изменение небольшой части кода может стать причиной появления ошибок у всей системы;
- после каждого изменения кода приходится тестировать всю систему на наличие ошибок.

В отличие от монолитной, использование многокомпонентной архитектуры дает следующие преимущества:

- писать и поддерживать небольшие части проще, чем одну большую систему;
- легче распределить разработчиков на написание кон-

кретной части системы;

- система может быть гетерогенной, так как для каждого из компонентов можно использовать свои языки и технологии, в зависимости от задачи;
- более легкое обновление, так как затрагивается только нужный компонент;
- система становится более отказоустойчивой, так как в случае выхода из строя одного из компонентов другие еще могут оказаться рабочими.

Тем самым выбор в пользу многокомпонентной архитектуры является обоснованным ввиду ряда преимуществ перед монолитной и наиболее подходящей с учетом требований к разрабатываемой платформе.

На рис. 1 изображена схема архитектуры платформы для психологических исследований. Архитектура разделена на отдельные компоненты, которые могут работать независимо и общаться между собой с помощью REST API.

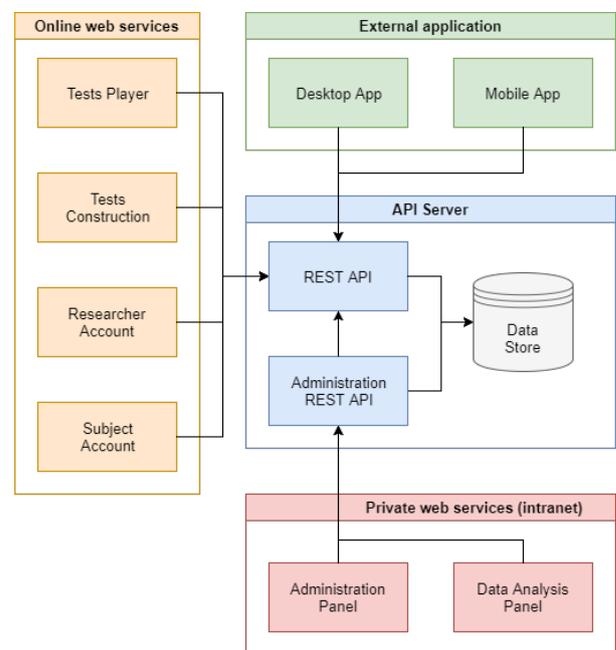


Рис. 1. Схема многокомпонентной архитектуры платформы

Fig. 1. Schema of a multi-component platform architecture

«API Server» является основным ядром системы, которое представляет собой REST API сервер и отвечает за работу с хранилищем данных, а также за выполнение различных сервисных функций.

«Online web services» являются компонентами, которые должны быть доступны из сети интернет. Они представляют основную онлайн часть разрабатываемой платформы, среди которых:

- онлайн плеер тестов;
- онлайн конструктор тестов;
- личный кабинет исследователя
- личный кабинет испытуемого.

«External application» являются отдельными приложениями, такие как десктопные и мобильные. В отличие от онлайн версии плеера тестов, особенностью приложений является то,



что процесс прохождения этих тестов должен работать без подключения к интернету. В связи с этим данные для тестов должны быть загружены заранее, а после прохождения тестов субьектом загружены обратно на сервер.

«Privatewebservices (intranet)» являются отдельными сервисами, состоящими из Панели администрирования платформы и Панели анализа данных. Особенность этих сервисов в том, что они должны быть изолированы от прямого доступа из сети интернет для обеспечения гарантии безопасности. Стоит также обратить внимание, что эти сервисы общаются со своим отдельным RESTAPI, включающим в себя административные методы, которые также не должны быть доступны из сети интернет.

3. Экспериментальная часть: выбор технологических решений на основе заданных критериев оценки

3.1. Выбор технологических решений для разработки клиентской части приложения

В результате рассмотрения JavaScript, Java апплетов и платформы Adobe Flash с точки зрения применимости для исполнения кода в браузере было выяснено, что применимым можно считать только JavaScript. Это связано не только с тем, что JavaScript используется во многих областях: клиентские браузеры, серверная часть, мобильные платформы, а также десктоп-приложения. Технология Java-апплетов, как и технология Adobe Flash, требуют установки дополнительных компонентов в систему пользователя. Более того, в зависимости от операционной системы и браузера может разниться процесс установки и

настройки. Ввиду большой вариативности аппаратного и программного обеспечения в образовательных учреждениях, в которых будут проходить исследования, применение этих двух технологий нецелесообразно, так как это может затруднить процесс проведения массовых исследований. Нужно учитывать, что в браузерах на мобильных устройствах Adobe Flash и Java апплеты не поддерживаются. Стоит также обратить внимание на то, что от Adobe Flash отказываются в пользу HTML5, который может работать с мультимедиа (видео и аудио).

Таким образом, выбор в пользу языка JavaScript для разработки клиентской части становится очевидным и альтернативного решения, в рамках заданных условий, на текущий момент нет. JavaScript поддерживается всеми распространенными браузерами и включен в них по умолчанию.

Разработка на стороне клиента крупных Single Page Applications (SPA) на базе чистого JavaScript является трудным и неэффективным процессом, поэтому необходимо использовать фреймворки, которые задают структуру приложения и имеют базовый набор компонентов. Почти все современные фреймворки обладают схожим функционалом и способны решить поставленную задачу. Тем самым выбор в первую очередь должен быть основан не на функционале фреймворка [12-14].

Для рассмотрения были выбраны наиболее известные и популярные фреймворки, среди которых Backbone.js [15], AngularJS 1 [16], AngularJS 2 [17], React.js [18], Ember.js [19, 20], Vue.js [18] и Polymer [21]. В таблице 1 показаны достоинства и недостатки этих фреймворков с учетом применимости к разрабатываемой платформе.

Таблица 1. Достоинства и недостатки Frontend фреймворков
Table 1. Advantages and disadvantages of Frontend frameworks

Фреймворк	Достоинства	Недостатки
Backbone.js	Компактный Простая структура Низкий порог вхождения Богатая документация (Rich documentation) Поддерживает REST	Не поддерживает двустороннюю привязку данных Требует дополнительных компонентов для реализации сложного функционала Плохо подходит для крупных проектов
AngularJS 1	Высокая популярность Низкий порог вхождения Богатая документация Большое сообщество Много готовых решений Является частью стека MEAN (MongoDB, Express, JS, AngularJS, NodeJS) Поддерживает REST Высокая скорость разработки Поддерживает двустороннюю привязку данных	Считается, что устарел, так как есть AngularJS 2 Не совместим с AngularJS 2 Скорость работы падает при достаточно большом количестве данных
AngularJS 2	Богатая документация Большое сообщество Имеет большое количество функций Поддерживает RESTwЕсть Angular Universal для решения проблем поисковой оптимизации (рендер страниц на сервере) Поддерживает двустороннюю привязку данных	Использует TypeScript для компиляции в JavaScript Более сложный порог вхождения, чем AngularJS 1 Необходимо сделать много действий для решения даже небольшого функционала



React	Компактный Высокая производительность Хорошая документация Подойдет для крупных и сложных проектов с высокой степенью нагрузки	Требует дополнительную реализацию на сервере для работы с данными (например, Flux или Redux) Не поддерживается REST Не совместим с библиотеками, которые модифицируют DOM Большой порог вхождения Сложный подход к разработке, непривычный для начинающих разработчиков
Ember.js	Богатая документация Большая экосистема Подходит для сложных и больших приложений Поддерживает REST Поддерживает двустороннюю привязку данных	Считается монолитным по сравнению с другими фреймворками Отсутствует повторное использование компонентов на уровне контроллера Больше порог вхождения, чем в других фреймворках Тяжелая структура Слишком большой для небольших проектов
Vue.js	Очень быстро растущая популярность Низкий порог вхождения Мало зависимостей Хорошая производительность Богатая документация Хорошая экосистема Поддерживает двустороннюю привязку данных	Достаточно молодой фреймворк Разрабатывается в основном одним человеком Сделано не очень много проектов Не поддерживается REST "из коробки" (есть библиотека Axios)
Polymer	Новая и перспективная технология Web Components Высокая скорость работы	Слишком молодое решение Большие риски при использовании Мало готовых решений и примеров Большой порог вхождения для начинающих

Backbone.js плохо подходит для разработки крупных проектов, так как отсутствуют необходимые компоненты для реализации сложного функционала. Тем самым, по мнению авторов статьи, использование данного фреймворка нецелесообразно ввиду того, что он не имеет достаточного функционала, а также имеются альтернативные решения.

Polymer является библиотекой, которая основывается на достаточно новой технологии Web Components. Спецификация W3C данной технологии еще не завершена. Могут быть проблемы с поддержкой браузеров, проблемы в стабильности работы, а также получается большой порог вхождения для разработчиков. В связи с этим использование этого фреймворка было решено отказать ввиду возможных рисков.

React, в отличие от других, является библиотекой и не позволяет создать веб приложение, так как он предназначен для создания View части и должен работать с данными на сервере, например, совместно с Flux или Redux. Поэтому React является сложным для понимания, имеет непривычную структуру, что усложняет понимание приложения в целом, а также имеет большой порог вхождения для начинающих разработчиков. По мнению авторов, на React сложнее сделать быстрый прототип и поддерживать решение, чем на другом фреймворке.

AngularJS 1, AngularJS 2, Ember.js и Vue.js имеют двустороннюю привязку данных, возможность построения крупных систем, хорошую документацию и сообщество. Основным выбором будет проводиться между данными фреймворками.

Ember.js имеет сложную структуру проекта и большой порог вхождения для начинающих разработчиков, а в случае выхода за пределы стандартного использования является громоздким и не гибким. Кроме того, фреймворк является менее популярным, чем AngularJS и Vue.js.

Vue.js версии 2 на текущий момент является самым быстрорастущим по популярности фреймворком, он взял лучшие решения от Ember.js, React и AngularJS, а также имеет хорошую производительность. Немаловажным фактором также является

то, что Vue.js не поддерживает REST и требует дополнительной библиотеки Axios для этого. Кроме того, фреймворк является молодым и разрабатывается в основном одним человеком, поэтому его использование может привести к большим рискам.

В результате, наиболее подходящими для разработки платформы для психологических исследований является AngularJS 1 и AngularJS 2. AngularJS 1 является достаточно простым фреймворком для освоения и понимания, обладает низким порогом вхождения при богатом наборе функций. AngularJS 2 является параллельным проектом с AngularJS 1 и разрабатывается отдельно. AngularJS 2 сильно усложнили, для написания простейшего приложения требуется гораздо больше действий. Кроме того, он написан на TypeScript, что потребует дополнительных знаний от разработчиков.

С учетом того, что разрабатываемая платформа для психологических исследований имеет ограничение в ресурсах, то наиболее подходящим решением на текущий момент является AngularJS 1. Кроме того AngularJS 1 имеет большую популярность и прогноз на использование, согласно данным GitHub и патентному анализу [22].

3.2. Выбор технологических решений для разработки серверной части приложения

Разработка серверной части платформы позволяет выбирать из достаточно широкого круга технологий, по сравнению с клиентской частью. Связано это с тем, что серверные технологии зависят от предпочтений разработчиков, оборудования и требований, предъявляемых к проекту. Выбор технологических решений для разработки серверных компонентов лучше начать не с языков программирования, а с рассмотрения фреймворков в связи с тем, что они задают основную структуру для разработки приложения. В таблице 2 представлены характерные особенности, достоинства и недостатки наиболее подходящих фреймворков для разработки серверной части платформы.



Таблица 2. Достоинства и недостатки Server-side фреймворков
Table 2. Advantages and disadvantages of Server-side Frameworks

Фреймворк	Используемый язык	Достоинства	Недостатки
Laravel, Symfony [23]	PHP	Низкий порог вхождения Большое количество разработчиков на PHP	Блокирующие IO вызовы Интерпретатор PHP имеет низкую производительность Нет платной поддержки
Django [24]	Python	Низкий порог вхождения Генерация панели администрирования для реляционных БД	Блокирующие IO вызовы Не поддерживает NoSQL-решения "из коробки"
Ruby on Rails [25]	Ruby	Низкий порог вхождения	В сообществе разработчиков есть упоминания проблем масштабирования при растущей нагрузке Блокирующие IO вызовы
Express.js [22, 26]	JavaScript (Node.js)	Не блокирующий по умолчанию (асинхронный) Плавная кривая обучения	Долгосрочная поддержка проекта имеет затруднения (сложность рефакторинга) Разработка большими группами может быть затруднительна
Loopback [27]	JavaScript (Node.js)	Не блокирующий по умолчанию (асинхронный) Генерация панели просмотра и работы с REST API Декларативный подход к генерации REST API	Сгенерированный API не содержит методов для массового обновления связанных между собой сущностей
Play [28]	Scala / Java	Не блокирующий по умолчанию (асинхронный) Хорошо масштабируем даже при блокирующем коде Строгая типизация упрощает рефакторинг	Медленная компиляция Новые версии фреймворка требуют доработок в конечном ПО
Vaadin [29]	Java	Содержит библиотеку готовых UI элементов Frontend-код генерируется на основе серверного Строгая типизация упрощает рефакторинг	Блокирующий по умолчанию Медленная компиляция Высокий порог вхождения Разработка новых элементов UI трудоемкая Нет полноценного контроля за Frontend-кодом
ASP .NET MVC [30]	C#	Строгая типизация упрощает рефакторинг	Lock-in на платформу Windows Необходимость покупки лицензий Windows Server для развертывания

Так как было определено, что требуется высокая степень масштабируемости проекта, следует акцентировать внимание на non-blocking I/O фреймворках. В связи с этим стоит исключить из рассмотрения Laravel, Symfony, Django и Ruby on Rails. Также, ввиду сложностей по реализации non-blocking I/O и custom интерфейсов, для проекта не подходит фреймворк Vaadin.

ASP .NET MVC накладывает дополнительные ограничения на инфраструктуру при отсутствии значимых достоинств, поэтому фреймворк следует исключить из дальнейшего рассмотрения. Таким образом, основной выбор будет проводится между фреймворками Express.js, Loopback и Play.

Немаловажным фактором является язык программирования, на котором написан фреймворк. Express.js и Loopback написаны на Node.js (JavaScript), в том время как Play на Java. В случае с JavaScript, как для клиентской, так и для серверной части будет применяться единый синтаксис. Это повысит эффективность разработки платформы, так как разработчику потребуется знать не два, а только один язык программирования, что явля-

ется плюсом в условиях небольшого количества разработчиков. Кроме того, это позволяет совместить часть процесса обучения и снизить общий порог вхождения, что в лучшую сторону повлияет на время подготовки новых специалистов, которые будут участвовать в разработке платформы. JavaScript является самым популярным языком в мире по данным статистики таких крупных сервисов, как GitHub и StackOverflow. В связи с этим целесообразнее применение фреймворков Express.js и Loopback, чем Play.

Из оставшихся двух фреймворков выбор в пользу Loopback более целесообразен по ряду причин:

- Loopback предлагает ряд паттернов, что поможет сохранить должный уровень поддерживаемости кодовой базы при ее увеличении;
- фреймворк основывается на Express.js, что позволит задействовать все его функциональные компоненты;
- Loopback предлагает функционал для упрощенной генерации API, что значительно снижает трудозатраты при разработке.



Перечисленное выше, по мнению авторов статьи, является более значимым, чем наиболее плавная кривая обучения. Таким образом, выбор остановлен на фреймворке Loopback.

4. Реализация: программа обучения для подготовки разработчиков

В процессе работы над проектом могут появляться новые разработчики, которым необходимо освоить стек технологий и инструментов или закрепить уже имеющиеся навыки, которые применяются разработке веб-ориентированной платформы.

Сокращение времени на подготовку каждого сотрудника в отдельности в условиях ограниченного количества опытных разработчиков является важной задачей. В связи с этим было принято решение разработать программу обучения, которая затрагивает основные аспекты разработки и применяемого набора технологий.

Программа обучения включает набор задач, над которыми обучаемый ведет самостоятельную работу согласно описанному в ней заданию. Все вопросы и обсуждения ведутся с ментором

(наставником). После завершения каждой задачи она сдается ментору, который проверяет правильность ее выполнения и задает контрольные вопросы. Вопросы, которые возникают в процессе работы над задачей рекомендуется записывать, а по достижению достаточного количества допускается задавать их ментору. На момент сдачи материала и ответов на контрольные вопросы предполагается, что сотрудник подготовлен и будет отвечать на вопросы самостоятельно, что позволит объективно оценить понимание материала.

Разработанная программа обучения включает в себя 2 основных части и разбита на отдельные задачи в Redmine (рисунки 2 и 3):

- **базовая часть**, включающая в себя основные навыки по веб-разработке (система управления проектами, система контроля версий, HTML5, CSS3, верстка, Semantic-UI, JavaScript и jQuery);
- **практическая часть**, включающая в себя разработку приложения по управлению списком задач (AngularJS, Node.js, NPM и Loopback).

Task #246: Задача #1.1. Система управления проектами Redmine	New	<input type="text"/>
Task #247: Задача #1.2. Система контроля версий Git	New	<input type="text"/>
Task #248: Задача #1.3. Правильная организация разработки с GitFlow, SemVer и GitLab	New	<input type="text"/>
Task #249: Задача #1.4. Основы веба, изучение HTML и CSS	New	<input type="text"/>
Task #250: Задача #1.5. Верстка тестовой страницы	New	<input type="text"/>
Task #251: Задача #1.6. Изучение CSS-фреймворка	New	<input type="text"/>
Task #252: Задача #1.7. Верстка страницы с использованием CSS-фреймворка Semantic UI	New	<input type="text"/>
Task #253: Задача #1.8. Изучение JavaScript	New	<input type="text"/>
Task #254: Задача #1.9. Разработка калькулятора на JavaScript	New	<input type="text"/>
Task #255: Задача #1.10. Расширенный JavaScript	New	<input type="text"/>
Task #256: Задача #1.11. Изучение jQuery	New	<input type="text"/>
Task #257: Задача #1.12. Стандарты кодирования и документирование в JavaScript	New	<input type="text"/>

Рис. 2. Первая часть программы обучения (базовая часть)

Fig. 2. The first part of the training program (basic part)

Задачи в Redmine копируются для каждого из обучаемых сотрудников из задач-шаблонов. По мере прохождения задач они отмечаются соответствующими статусами, указывается процент их выполнения, пишутся комментарии и производится оценка затраченного времени, что позволяет ментору отслеживать — на каком именно этапе освоения материала находится обучаемый и каков его прогресс.

Каждая задача включает в себя несколько пунктов:

- **пункт «Задание»** — это основное задание, которое необходимо выполнить;
- **пункт «Дополнительно»** — как правило, содержит дополнительные ссылки и материалы;
- **пункт «Контрольные вопросы»** — контрольные вопросы, по которым необходимо будет отчитываться после завершения задачи.

Разработанная программа обучения уже применяется на практике и хорошо показала себя: обучаемый самостоятельно проходит материал и привыкает к работе с системой управления проектами Redmine, а ментор сокращает время на обучение. Для дополнительного контроля и ведения статистики был создан опросник с помощью GoogleForms, которые заводится на каждого из обучаемых сотрудников. Ментор в определенный промежуток времени заполняет опросник на обучаемого, в результате чего получается отчет, пример которого показан на рисунке 5. Это также позволяет оценить, насколько разработчик подготовлен для выполнения задач по разработке веб-ориентированной платформы.



Support #494: Задача #2.1. Разработка ToDo. Подготовка и настройка рабочего окружения (Frontend)	New	<input type="checkbox"/>
Task #495: Задача #2.2. Теоретическая подготовка по AngularJS 1.x	New	<input type="checkbox"/>
Support #496: Задача #2.3. Создание репозитория проекта (Angular)	New	<input type="checkbox"/>
Feature #497: Задача #2.4. Применение пакетного менеджера для загрузки зависимостей проекта	New	<input type="checkbox"/>
Feature #498: Задача #2.5. Инициализация Angular на веб-странице	New	<input type="checkbox"/>
Feature #499: Задача #2.6. Роутинг, контроллеры и ссылки в Angular	New	<input type="checkbox"/>
Feature #500: Задача #2.7. Верстка страницы ToDo с применением Semantic UI	New	<input type="checkbox"/>
Feature #501: Задача #2.8. Сервисы и директивы в Angular	New	<input type="checkbox"/>
Support #502: Задача #2.9. Релиз готового функционала	New	<input type="checkbox"/>
Feature #503: Задача #2.10. Работа с Local Storage	New	<input type="checkbox"/>
Support #504: Задача #2.11. Подготовка виртуального рабочего окружения (Backend)	New	<input type="checkbox"/>
Task #505: Задача #2.12. Теоретическая подготовка по Loopback 3.x	New	<input type="checkbox"/>
Support #506: Задача #2.13. Создание репозитория проекта (Loopback)	New	<input type="checkbox"/>
Feature #507: Задача #2.14. Инициализация приложения Loopback, использование Explorer'a для просмотра API	New	<input type="checkbox"/>
Feature #508: Задача #2.15. Data Source, Модели, Boot-скрипты и генерация CRUD в Loopback	New	<input type="checkbox"/>
Feature #509: Задача #2.16. Создание дополнительных удаленных методов в Loopback	New	<input type="checkbox"/>
Feature #510: Задача #2.17. Экспорт и встраивание Angular SDK в проект	New	<input type="checkbox"/>
Feature #511: Задача #2.18. Наследование и связи между моделями в Loopback. Встроенные модели.	New	<input type="checkbox"/>
Feature #512: Задача #2.19. Список контроля доступа в Loopback. Авторизация.	New	<input type="checkbox"/>

Рис. 3. Вторая часть программы обучения (практическая часть)
Fig. 3. The second part of the training program (practical part)

Задание:

1. Прочитать про модель ветвления GitFlow - <http://nvie.com/posts/a-successful-git-branching-model/> (перевод - <https://habrahabr.ru/post/106912/>)
2. Посмотреть спецификацию SemVer - <http://semver.org/>
3. Прочитать материал в Базе Знаний - [Git - Правила работы](#)
4. Посмотреть GitLab, его отличия от GitHub и Bitbucket
5. Сделать тестовое задание - [Тестовое задание по Git, Git Flow и SemVer](#)

Дополнительно:

1. Установить Atom с плагином Git-Plus и попробовать работу с Git
2. Установить и посмотреть SourceTree [только Win и Mac] - <https://www.sourcetreeapp.com/>

Контрольные вопросы:

1. На каком языке писать комментарии в коде, названия коммитов, описание в GitLab и задачи в Redmine?
2. Как указать в коммите привязку к задаче в Redmine?
3. Может ли быть коммит без указания номера задачи в Redmine?
4. Зачем нужен GitFlow?
5. Какие ветви являются самыми главными, а какие вспомогательными в GitFlow?
6. Как используются ветви функциональности (feature)? Куда вливаются эти ветви и зачем?
7. Как используются ветви релиза? Куда вливаются эти ветви и зачем?
8. Как используются ветви исправлений? Куда вливаются эти ветви и зачем?
9. Как правильно именовать функциональные ветки (feature)?
10. Что дает флаг --no-ff при слиянии веток? Можно ли сделать так, чтобы не указывать его каждый раз?
11. Может ли одна функциональная ветка (feature) быть вложена в другую и почему?
12. Какие правила слияния веток (merge) указаны в Базе Знаний?
13. Как указываются номера версий по спецификации SemVer?
14. Что такое GitLab, зачем он нужен и какие есть альтернативы?

Рис. 4. Пример задания 1.3 «Правильная организация разработки с GitFlow, SemVer и GitLab»
Fig. 4. Sample task 1.3 «Proper organization of development with GitFlow, SemVer and GitLab»



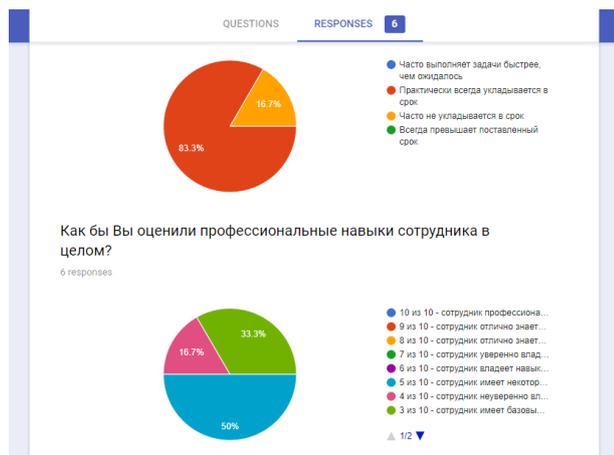


Рис. 5. Пример части отчета по одному из сотрудников
Fig. 5. Example of part of the report for one of the employees

5. Заключение

Гарантированное качество функционирования больших распределенных систем, их успешная и эффективная работа в течении ряда лет, возможности масштабирования и взаимодействия с различными платформами закладывается на этапе проектирования. Сейчас отходит в прошлое характерная еще десятилетием назад хаотическая разработка, приводившая к тому, что к процессу изготовления продукта и запуска его в эксплуатацию информационные технологии уже устаревали. Новые подходы требуют тщательного и всестороннего документирования процесса проектирования и реализацию. Широко используется понятия экосистемы языков программирования, автоматизированные системы и инструменты разработки, позволяющие проектировать информационную систему, способную к развитию и обеспечению качеству и надежности решаемых задач.

В статье на конкретном примере цифровой платформы психологических исследований показан процесс выбора технологий для разработки клиентской и серверной части. В настоящее время ядро системы создано и проходит опытную эксплуатацию в образовательном учреждении для информационной поддержки психологических исследований. Созданная реализация показала адекватный выбор программных технологий, система успешно функционирует в тестовом режиме, одновременно расширяя свои функциональные возможности.

Созданная программа обучения также показала свою эффективность, уменьшив затраты на привлечение новых программистов. Система обучения также была опробована на студентах и аспирантах кафедры управления и моделирования систем МИРЭА, что позволило оценить методические результаты и усилить качество подготовки за счет обучения реальным технологиях программной инженерии.

Благодарности

Исследование выполнено в рамках государственного заказа Министерства образования и науки Российской Федерации «Разработка технологической концепции Дата-центра междисциплинарных исследований в образовании» (проект 25.13253.2018/12.1).

Список использованных источников

- [1] Venkatesan D., Sridhar S. A novel programming framework for architecting next generation enterprise scale information systems // Information Systems and e-Business Management. 2017. Vol. 15, issue 2. Pp. 489-534. DOI: 10.1007/s10257-016-0330-y
- [2] Chen Y., Huang J., Lin C., Hu J. A partial selection methodology for efficient QoS-aware service composition // IEEE Transactions on Services Computing. 2015. Vol. 8, issue 3. Pp. 384-397. DOI: 10.1109/TSC.2014.2381493
- [3] Zhang H., Lu M., Gu T. SOA software architecture extended modeling considering reliability information // Proceedings of 2017 IEEE Second International Conference on Reliability Systems Engineering (ICRSE), 2017. Pp. 1-6. DOI: 10.1109/ICRSE.2017.8030794
- [4] Zakharov I., Nikulchev E., Ilin D., Ismatullina V., Fenin A. Web-based Platform for Psychology Research // ITM Web of Conferences. 2017. Vol. 10 (04006). DOI: 10.1051/itm-conf/20171004006
- [5] Nikulchev E., Kolyasnikov P., Ilin D., Kasatonov S., Biryukov D., Zakharov I. Selection of Architectural Concept and Development Technologies for the Implementation of a Web-Based Platform for Psychology Research // Proceedings of IEEE 2018 Computing Conference (10–12 July, 2018). London, 2018. Pp. 370-376.
- [6] Malykh S.B., Tikhomirova T.N. Personality features and intellect: the nature of correlation // Voprosy psihologii. 2015. Vol. 2. Pp. 147–160.
- [7] Tikhomirova T., Lysenkova I., Ismatullina V., Malykh S. Cross-cultural study of mother-child interactions and child's intelligence // The European Proceedings of Social & Behavioural Sciences. 2017. Vol. 33. Pp. 353-366. DOI: 10.15405/epsbs.2017.12.38
- [8] Chattopadhyay S., Banerjee A., Banerjee N. A fast and scalable mechanism for Web service composition // ACM Transactions on the Web. 2017. Vol. 11, issue 4. Pp. 26. DOI: 10.1145/3098884
- [9] Barak M., Ziv S. Wandering: A Web-based platform for the creation of location-based interactive learning objects // Computers & Education. 2013. Vol. 62. Pp. 159-170. DOI: 10.1016/j.compedu.2012.10.015
- [10] Venkateswaran N., Changder S. Simplified data partitioning in a consistent hashing based sharding implementation // Proceedings of IEEE Region 10 Conference, TENCON 2017-2017 (5-8 November, 2017). Penang, Malaysia, 2017. Pp. 895-900. DOI: 10.1109/TENCON.2017.8227985
- [11] Venkateswaran N., Changder S. Handling workload skew in a consistent hashing based partitioning implementation // Proceedings of IEEE 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI) (13-16 September, 2017). Udupi, India, 2017. Pp. 1163-1169. DOI: 10.1109/ICACCI.2017.8125999
- [12] Kristensen E K., Møller A. Type test scripts for TypeScript testing // Proceedings of the ACM on Programming Languages (OOPSLA), Article 90 (October 2017), 25 pages. DOI: 10.1145/3133914
- [13] Dhok M., Ramanathan M. K., Sinha N. Type-aware concolic testing of JavaScript programs // Proceedings of the 38th International Conference on Software Engineering (ICSE



- '16). ACM, New York, NY, USA, 2016. Pp. 168-179. DOI: 10.1145/2884781.2884859
- [14] Bae S., Park J., Ryu S. Partition-based coverage metrics and type-guided search in concolic testing for JavaScript applications // Proceedings of 2017 IEEE/ACM 5th International FME Workshop on Formal Methods in Software Engineering (FormaliSE) (27-27 May 2017). Buenos Aires, Argentina, 2017. Pp. 72-78. DOI: 10.1109/FormaliSE.2017.10
- [15] Mardan A. Backbone. js and Parse. com. In Full Stack JavaScript. Apress, Berkeley, CA, 2015. Pp. 121-136. DOI: 10.1007/978-1-4842-1751-1_5
- [16] Chansuwath W., Senivongse T. A model-driven development of web applications using AngularJS framework // Proceedings of 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS) (26-29 June 2016). Okayama, Japan, 2016. Pp. 1-6. DOI: 10.1109/ICIS.2016.7550838
- [17] Fain Y., Moiseev A. Angular 2 Development with TypeScript (1st ed.). Manning Publications Co., Greenwich, CT, USA, 2016. 456 p.
- [18] Sheppard D. PWAs From the Start. In: Beginning Progressive Web App Development. Apress, Berkeley, CA, 2017. Pp. 209-240. DOI: 10.1007/978-1-4842-3090-9_11
- [19] Hamdy A., Ibrahim O., Hazem A. A Web Based Framework for Pre-release Testing of Mobile Applications // Proceedings of 20th International Conference on Circuits, Systems, Communications and Computers (CSCC 2016). MATEC Web of Conferences Vol. 76, (04041), 2016. DOI: 10.1051/mateconf/20167604041
- [20] Cravens J., Brady T.Q. Building Web Apps with Ember. js: Write Ambitious Javascript. O'Reilly Media, Inc., 2014. 188 p.
- [21] Kristensen E.K., Møller A. Inference and Evolution of TypeScript Declaration Files. In: Huisman M., Rubin J. (eds) Fundamental Approaches to Software Engineering. FASE 2017 // Lecture Notes in Computer Science. Vol. 10202. Springer, Berlin, Heidelberg, 2017. Pp. 99-115. DOI: 10.1007/978-3-662-54494-5_6
- [22] Nikulchev E., Ilin D., Bubnov G., Mateshuk E. Scalable service for predictive learning based on the professional social networking sites // International Journal of Advanced Computer Science and Applications. 2017. Vol. 8, issue 5. Pp. 9-15. DOI: 10.14569/IJACSA.2017.080502
- [23] Sinha S. How Request, Response Work in Laravel 5. Beginning Laravel. Apress, Berkeley, CA, 2017. Pp. 161-165. DOI: 10.1007/978-1-4842-2538-7_18
- [24] Lotfy K., Hale M.L. Assessing pairing and data exchange mechanism security in the wearable Internet of Things // Proceedings of 2016 IEEE International Conference on Mobile Services (MS) (27 June-2 July 2016). San Francisco, CA, USA, 2016. Pp. 25-32. DOI: 10.1109/MobServ.2016.15
- [25] Zhang Y., Yin G., Yu Y., Wang H. Investigating social media in GitHub's pull-requests: a case study on Ruby on Rails // Proceedings of the 1st International Workshop on Crowd-based Software Development Methods and Technologies. ACM, New York, NY, USA, 2014. Pp. 37-41. DOI: 10.1145/2666539.2666572
- [26] Chaniotis I.K., Kyriakou K.I. D., Tselikas N.D. Is Node. js a viable option for building modern web applications? A performance evaluation study // Computing. 2015. Vol. 97, issue 10. Pp. 1023-1044. DOI: 10.1007/s00607-014-0394-9
- [27] Rankovski G., Chorbev I. Improving Scalability of Web Applications by Utilizing Asynchronous I/O // Advances in Intelligent Systems and Computing. 2016. Vol. 665. Pp. 211-218. DOI: 10.1007/978-3-319-68855-8_21
- [28] Hunt J. Play framework // A Beginner's Guide to Scala, Object Orientation and Functional Programming. Springer, Cham, 2018. Pp. 431-446. DOI: 10.1007/978-3-319-75771-1_38
- [29] Alexopoulos K., Koukas S., Boli N., Mourtzis D. Resource Planning for the Installation of Industrial Product Service Systems. In: Lödding H., Riedel R., Thoben KD., von Cieminski G., Kiritsis D. (eds) // Advances in Production Management Systems. The Path to Intelligent, Collaborative and Sustainable Manufacturing. APMS 2017. IFIP Advances in Information and Communication Technology. Vol. 514. Springer, Cham, 2017. Pp. 205-213. DOI: 10.1007/978-3-319-66926-7_24
- [30] Troelsen A., Japikse P. Introducing ASP. NET MVC // Pro C# 7. Apress, Berkeley, CA, 2017. Pp. 1179-1221. DOI: 10.1007/978-1-4842-3018-3_29

Поступила 16.05.2018; принята в печать 10.06.2018;
опубликована онлайн 30.06.2018.

References

- [1] Venkatesan D., Sridhar S. A novel programming framework for architecting next generation enterprise scale information systems. *Information Systems and e-Business Management*. 2017; 15(2):489-534. DOI: 10.1007/s10257-016-0330-y
- [2] Chen Y., Huang J., Lin C., Hu J. A partial selection methodology for efficient QoS-aware service composition. *IEEE Transactions on Services Computing*. 2015; 8(3):384-397. DOI: 10.1109/TSC.2014.2381493
- [3] Zhang H., Lu M., Gu T. SOA software architecture extended modeling considering reliability information. *Proceedings of 2017 IEEE Second International Conference on Reliability Systems Engineering (ICRSE)*, pp. 1-6, 2017. DOI: 10.1109/ICRSE.2017.8030794
- [4] Zakharov I., Nikulchev E., Ilin D., Ismatullina V., Fenin A. Web-based Platform for Psychology Research. *ITM Web of Conferences*. 2017; 10(04006). DOI: 10.1051/itm-conf/20171004006
- [5] Nikulchev E., Kolyasnikov P., Ilin D., Kasatonov S., Biryukov D., Zakharov I. Selection of Architectural Concept and Development Technologies for the Implementation of a Web-Based Platform for Psychology Research. *Proceedings of IEEE 2018 Computing Conference (10-12 July, 2018)*. London, pp. 370-376, 2018.
- [6] Malykh S.B., Tikhomirova T.N. Personality features and intellect: the nature of correlation. *Voprosy psichologii*. 2015; 2:147-160.
- [7] Tikhomirova T., Lysenkova I., Ismatullina V., Malykh S. Cross-cultural study of mother-child interactions and child's intelligence. *The European Proceedings of Social & Behavioural Sciences*. 2017; 33:353-366. DOI: 10.15405/epsbs.2017.12.38
- [8] Chattopadhyay S., Banerjee A., Banerjee N. A fast and scalable mechanism for Web service composition. *ACM Transactions on the Web*. 2017; 11(4):26. DOI: 10.1145/3098884
- [9] Barak M., Ziv S. Wandering: A Web-based platform for the creation of location-based interactive learning objects. *Computers & Education*. 2013; 62:159-170. DOI: 10.1016/j.compedu.2012.10.015



- [10] Venkateswaran N., Changder S. Simplified data partitioning in a consistent hashing based sharding implementation. *Proceedings of IEEE Region 10 Conference, TENCON 2017-2017* (5-8 November, 2017). Penang, Malaysia, 2017. pp. 895-900. DOI: 10.1109/TENCON.2017.8227985
- [11] Venkateswaran N., Changder S. Handling workload skew in a consistent hashing based partitioning implementation. *Proceedings of IEEE 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)* (13-16 September, 2017). Udupi, India, pp. 1163-1169, 2017. DOI: 10.1109/ICACCI.2017.8125999
- [12] Kristensen E.K., Møller A. Type test scripts for TypeScript testing. *Proceedings of the ACM on Programming Languages (OOPSLA)*, Article 90 (October 2017), 25 pages. DOI: 10.1145/3133914
- [13] Dhok M., Ramanathan M. K., Sinha N. Type-aware concolic testing of JavaScript programs. *Proceedings of the 38th International Conference on Software Engineering (ICSE '16)*. ACM, New York, NY, USA, pp. 168-179, 2016. DOI: 10.1145/2884781.2884859
- [14] Bae S., Park J., Ryu S. Partition-based coverage metrics and type-guided search in concolic testing for JavaScript applications. *Proceedings of 2017 IEEE/ACM 5th International FME Workshop on Formal Methods in Software Engineering (FormalISE)* (27-27 May 2017). Buenos Aires, Argentina, pp. 72-78, 2017. DOI: 10.1109/FormalISE.2017.10
- [15] Mardan A. Backbone.js and Parse.com. In *Full Stack JavaScript*. Apress, Berkeley, CA, pp. 121-136, 2015. DOI: 10.1007/978-1-4842-1751-1_5
- [16] Chansuwath W., Senivongse T. A model-driven development of web applications using AngularJS framework. *Proceedings of 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS)* (26-29 June 2016). Okayama, Japan, pp. 1-6, 2016. DOI: 10.1109/ICIS.2016.7550838
- [17] Fain Y., Moiseev A. *Angular 2 Development with TypeScript* (1st ed.). Manning Publications Co., Greenwich, CT, USA, 2016. 456 p.
- [18] Sheppard D. PWAs From the Start. In: *Beginning Progressive Web App Development*. Apress, Berkeley, CA, 2017. pp. 209-240. DOI: 10.1007/978-1-4842-3090-9_11
- [19] Hamdy A., Ibrahim O., Hazem A. A Web Based Framework for Pre-release Testing of Mobile Applications. *Proceedings of 20th International Conference on Circuits, Systems, Communications and Computers (CSCC 2016)*. MATEC Web of Conferences Vol. 76, (04041), 2016. DOI: 10.1051/mateconf/20167604041
- [20] Cravens J., Brady T.Q. *Building Web Apps with Ember.js: Write Ambitious Javascript*. O'Reilly Media, Inc., 2014. 188 p.
- [21] Kristensen E.K., Møller A. Inference and Evolution of TypeScript Declaration Files. In: Huisman M., Rubin J. (eds) *Fundamental Approaches to Software Engineering. FASE 2017. Lecture Notes in Computer Science*. Vol. 10202. Springer, Berlin, Heidelberg, 2017. pp. 99-115. DOI: 10.1007/978-3-662-54494-5_6
- [22] Nikulchev E., Ilin D., Bubnov G., Mateshuk E. Scalable service for predictive learning based on the professional social networking sites. *International Journal of Advanced Computer Science and Applications*. 2017; 8(5):9-15. DOI: 10.14569/IJACSA.2017.080502
- [23] Sinha S. How Request, Response Work in Laravel 5. *Beginning Laravel*. Apress, Berkeley, CA, 2017. pp. 161-165. DOI: 10.1007/978-1-4842-2538-7_18
- [24] Lotfy K., Hale M.L. Assessing pairing and data exchange mechanism security in the wearable Internet of Things. *Proceedings of 2016 IEEE International Conference on Mobile Services (MS)* (27 June-2 July 2016). San Francisco, CA, USA, pp. 25-32, 2016. DOI: 10.1109/MobServ.2016.15
- [25] Zhang Y., Yin G., Yu Y., Wang H. Investigating social media in GitHub's pull-requests: a case study on Ruby on Rails. *Proceedings of the 1st International Workshop on Crowd-based Software Development Methods and Technologies*. ACM, New York, NY, USA, pp. 37-41, 2014. DOI: 10.1145/2666539.2666572
- [26] Chaniotis I.K., Kyriakou K.I. D., Tselikas N.D. Is Node.js a viable option for building modern web applications? A performance evaluation study. *Computing*. 2015; 97(10):1023-1044. DOI: 10.1007/s00607-014-0394-9
- [27] Rankovski G., Chorbev I. Improving Scalability of Web Applications by Utilizing Asynchronous I/O. *Advances in Intelligent Systems and Computing*. 2016; 665:211-218. DOI: 10.1007/978-3-319-68855-8_21
- [28] Hunt J. *Play framework. A Beginner's Guide to Scala, Object Orientation and Functional Programming*. Springer, Cham, pp. 431-446, 2018. DOI: 10.1007/978-3-319-75771-1_38
- [29] Alexopoulos K., Koukas S., Boli N., Mourtzis D. Resource Planning for the Installation of Industrial Product Service Systems. In: Lödding H., Riedel R., Thoben KD., von Cieminski G., Kiritsis D. (eds) *Advances in Production Management Systems. The Path to Intelligent, Collaborative and Sustainable Manufacturing. APMS 2017. IFIP Advances in Information and Communication Technology*. Vol. 514. Springer, Cham, pp. 205-213, 2017. DOI: 10.1007/978-3-319-66926-7_24
- [30] Troelsen A., Japikse P. Introducing ASP.NET MVC. In *Pro C# 7*. Apress, Berkeley, CA, pp. 1179-1221, 2017. DOI: 10.1007/978-1-4842-3018-3_29

Submitted 16.05.2018; revised 10.06.2018;
published online 30.06.2018.

About the authors:

Dmitry Yu. Ilyin, postgraduate student, MIREA – Russian Technological University (78 Vernadsky Prospekt, Moscow 119454, Russia); Chief analyst of the Laboratory for Monitoring the Efficiency and Quality of Scientific Research of the Information and Analytical Center, Russian Academy of Education (8 Pogodinskaya St., Moscow 119121, Russia), <http://orcid.org/0000-0002-0241-2733>, i@dmitryilin.com

Eugene V. Nikulchev, Doctor of Technical Sciences, Professor, Department of Control and Modeling Systems, MIREA – Russian Technological University (78 Vernadsky Prospekt, Moscow 119454, Russia); Chief analyst of the Laboratory for Monitoring the Efficiency and Quality of Scientific Research of the Information and Analytical Center, Russian Academy of Education (8 Pogodinskaya St., Moscow 119121, Russia), <http://orcid.org/0000-0003-1254-9132>, nikulchev@mail.ru

Pavel V. Kolyasnikov, Leading Analyst of the Laboratory of Monitoring the Efficiency and Quality of Scientific Research of the Information and Analytical Center, Russian Academy of Education (8 Pogodinskaya St., Moscow 119121, Russia), <http://orcid.org/0000-0003-3633-5913>, pavelkolyasnikov@gmail.com



This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>), which permits unrestricted reuse, distribution, and reproduction in any medium provided the original work is properly cited.

