



Международный журнал информационных технологий и энергоэффективности

Сайт журнала:

<http://www.openaccessscience.ru/index.php/ijcse/>



УДК 004.9

СОВРЕМЕННЫЕ ИНСТРУМЕНТЫ ДЛЯ АВТОМАТИЗАЦИИ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Антонов Д.А.

ФГБОУ ВО «Чувашский государственный университет им. И.Н. Ульянова», г. Чебоксары, Россия (428015, г. Чебоксары, Чувашская Республика, Московский просп., 15), e-mail: bydevil21@gmail.com

В статье рассматривается проблема эффективной разработки программного обеспечения. Описываются наиболее распространенные современные инструменты и практики для автоматизации процесса, позволяющие его упростить, повысить качество создаваемых цифровых продуктов и снизить общую стоимость IT-проектов. Целевая аудитория – разработчики, тестировщики, аналитики.

Ключевые слова: Разработка программного обеспечения, инструменты автоматизации, интегрированная среда разработки, искусственный интеллект, машинное обучение, облачные технологии, low-code, no-code, IDE, BaaS, CaaS, FaaS, IaaS, PaaS, SaaS.

MODERN TOOLS FOR AUTOMATING SOFTWARE DEVELOPMENT

Antonov D.A.

S FSBEI of HE "I.N. Ulianov Chuvash State University", Cheboksary, Russia (428015, Cheboksary, Chuvash Republic, Moskovsky Ave., 15), e-mail: bydevil21@gmail.com

The article deals with the problem of effective software development. The most common modern tools and practices for process automation are described, which make it possible to simplify the process, improve the quality of digital products and reduce the overall cost of IT projects. Target audience – developers, testers, analysts.

Keywords: Software development, automation tools, integrated development environment, artificial intelligence, machine learning, cloud technologies, low-code, no-code, IDE, BaaS, CaaS, FaaS, IaaS, PaaS, SaaS.

В процессе создания программной системы каждый разработчик нередко сталкивается с необходимостью применения типовых синтаксических конструкций, стандартных логических шагов и алгоритмов, а также прочих шаблонных взаимодействий. Несмотря на широкие возможности, предоставляемые современными языками программирования высокого уровня, данная проблема остается актуальной даже при использовании различных парадигм и лучших инженерных практик.

Нацеленность на более простое восприятие кода человеком в совокупности с необъятной вариативностью прикладных задач и путей для их решения, приводят к невозможности формализации всех возможных инструкций в универсальный лаконичный формат. Именно эти факторы в значительной мере приводят к необходимости персонализации инструментов для разработки и командного взаимодействия в случае стремления к максимальной эффективности при реализации программных продуктов.

К современным средствам, позволяющим конструировать программные комплексы различных масштабов, в первую очередь, относятся интегрированные среды разработки. Получившие огромную популярность за последнее десятилетие в силу удобства использования и гибкости настройки, они помогают решать большинство прикладных проблем быстро и эффективно. Однако в последние годы также активно применяются специализированные облачные сервисы и инструменты на основе искусственного интеллекта в дополнение к IDE, а также обособлено – для ряда задач.

Выбор наиболее подходящего инструмента обычно зависит от имеющихся у специалиста знаний, навыков и ресурсов, а также целевых задач и потенциальных внешних требований. В данной работе будут рассмотрены перечисленные выше современные технологии, их преимущества в решении задач по реализации необходимой программной функциональности и дальнейшие перспективы. Это позволит сформировать понимание о том, как оставаться конкурентоспособным игроком в стремительно растущей IT-отрасли.

Основная часть

1. Интегрированные среды разработки

Целевая аудитория: разработчики, тестировщики.

В данную категорию входят различные программные средства, предназначенные для разработчиков и выполняющие вспомогательные функции для более быстрого решения задач, связанных с написанием кода. Среди ключевых функций, позволяющих продуктивно работать с системами различных масштабов, можно выделить следующие:

- автоматическое дополнение (в том числе, благодаря инструментам на основе машинного обучения);
- контекстные подсказки, сообщающие о возможных операциях в текущем окружении, наличии ошибок или иных проблемах;
- генераторы шаблонных частей из предопределенных заготовок («сниппетов»);
- утилиты, порождающие код из декларативных спецификаций или графических схем.

Довольно длительное время каждый из продуктов, появляющийся на рынке текстовых редакторов и интегрированных сред разработки, стремился охватить наибольшую часть аудитории, предоставляя подобные возможности различного качества на основе внутренних разработок. Однако в процессе развития IT-индустрии, в ходе которого все сильнее проявлялась тенденция глобализации, стратегии присутствующих игроков менялись: все чаще стали предприниматься попытки по достижению технических договоренностей, единым подходам и протоколам взаимодействия.

По этим причинам профессиональные инструменты становились более гибкими, позволяя сторонним разработчикам осуществлять интеграции для расширения встроенной функциональности посредством механизмов плагинов. Также были сделаны серьезные шаги в направлении унификации инженерных соглашений по единым форматам структурирования информации и ее шаблонизации, а также по обмену данными и их обработке.

Это привело к тому, что, в соответствии с современными архитектурными подходами, IDE чаще всего представляют собой модульные системы и обеспечивают различные способы настройки под собственные нужды. Работа, проделанная тысячами высококвалифицированных инженеров отрасли, позволила дать доступ каждому программисту, независимо от уровня его компетенций, к гибкой настройке ежедневно

используемых инструментов. В свою очередь, данный фактор послужил дополнительным импульсом для развития индустрии: она стала расширяться с большей скоростью благодаря улучшению межгруппового и внутрикомандного взаимодействия, росту личной эффективности и снижению порога входа в мир корпоративных систем. Поэтому можно с большой уверенностью считать, что среды интегрированной разработки останутся важнейшими помощниками на ближайшее десятилетие.

2. Облачные технологии

Целевая аудитория – разработчики, тестировщики, аналитики.

Данный подход подразумевает наличие в цепи процесса создания ПО промежуточного звена – облачного провайдера. Разработчик (аналитик, компания) при подобной модели не нуждается в самостоятельном приобретении высокопроизводительной техники для функционирования создаваемых программных систем, поскольку все процессы по обработке информации могут выполняться на удаленных ресурсах, за пользование которыми необходимо осуществлять регулярную оплату, соразмерную объему выполненных операций и мощности арендованных вычислительных машин [1].

При использовании рассматриваемой схемы достигается эффективное использование ресурсов в целевых прикладных задачах благодаря отсутствию необходимости в регулярном решении инфраструктурных проблем. Чаще всего облачные решения используют малочисленные команды и молодые компании, для которых покупка дорогостоящего оборудования является серьезным экономическим испытанием. Однако, стоит отметить, что крупные компании также нередко обращаются к рассматриваемой модели – например, для проведения различных экспериментов отдельными группами разработки.

Несмотря на то, что основные концепции облачных вычислений были опубликованы еще в конце прошлого столетия, активное развитие и распространение облачные технологии получили лишь в течение последнего десятилетия. Это обусловлено, в первую очередь, имеющимися ранее техническими ограничениями, которые удалось преодолеть. Так, например, среди ключевых достижений можно выделить следующие:

1. Широкополосное интернет-подключение – стала возможной быстрая передача данных между удаленными серверами и пользовательскими устройствами.

2. Высокоскоростные сети – оптические сети, обеспечивающие эффективную коммуникацию, привели к возможности создания распределенных дата-центров.

3. Виртуализация – возможность создания виртуальных экземпляров операционных систем, хранилищ данных и иных сетевых ресурсов для эффективного использования физических мощностей с обеспечением изолированности и гибкости настройки.

4. Централизованное управление ресурсами – контроль за вычислительными мощностями и предоставление их по требованию пользователя в режиме реального времени.

5. Автоматизация и оркестрация – позволили создавать гибкую инфраструктуру из физических и виртуальных элементов для автоматизированного масштабирования ресурсов, управления нагрузкой между серверами и реализации эффективного мониторинга.

6. Большие объемы данных (Big Data) – совершенствование подходов по хранению, обработке и анализу данных позволило справляться с непрерывными потоками информации.

Перечисленные достижения научно-технического прогресса в значительной мере способствовали ускоренному совершенствованию существующих подходов в разработке

программного обеспечения и формированию новых. Большинство из них стали фундаментальными для возможности становления облачных технологий и их применения как в профессиональной сфере, так и в повседневной жизни современного человека [2-3].

На сегодняшний день в обширных перечнях облачных услуг, предоставляемых специализированными провайдерами, чаще всего встречаются следующие позиции:

- **ВaaS (Backend-as-a-service – серверная часть, как услуга)** – предоставляется настроенная функциональность и инфраструктура для управления серверной частью приложений;
- **СааС (Containers-as-a-service – контейнеры, как услуга)** – виртуализация на основе контейнеров, позволяющая упростить управление как отдельными микросервисами, так и реализовать оркестрацию над сложными механизмами;
- **ФааС (Function-as-a-service – функция, как услуга)** – возможность «бессерверного» запуска фрагментов кода (н-р, отдельных функций) при наступлении определенных событий в облачной системе;
- **ИааС (Infrastructure-as-a-service – инфраструктура, как услуга)** – гибкая масштабируемая инфраструктура из необходимого количества виртуальных машин и иных вычислительных ресурсов для развертывания приложений с возможностью простого управления;
- **РааС (Platform-as-a-service – платформа, как услуга)** – полноценная платформа для разработки, включающая оборудование, необходимое программное обеспечение с лицензиями, а также настройку и обслуживание облачной инфраструктуры.
- **СааС (Software-as-a-service – программное обеспечение, как услуга)** – возможность аренды лицензионного программного обеспечения и его использования через Интернет без необходимости в самостоятельной настройке и поддержке на собственных мощностях.

Каждая из услуг содержит специализацию под конкретные нужды. Они имеют множество общих преимуществ и недостатков, поскольку базируются на основных принципах облачных технологий. Перечень общих достоинств, не включающий перечисленные ранее достижения НТП, представлен ниже [4-5]:

1. Гибкость и масштабируемость – использование вычислительных ресурсов исключительно по мере необходимости, а также в объеме и производительности в зависимости от потребностей и нагрузки системы.

2. Упрощение управления и снижение затрат – отсутствие необходимости в формировании и полного обслуживания собственной инфраструктуры, автоматизация различных процессов управления, удобство настройки через графический пользовательский интерфейс.

3. Высокий уровень доступности и отказоустойчивости – специализация центров обработки данных в процессах распределения ресурсов, репликации и защиты данных на множестве физических машин.

4. Повышение эффективности – необходимость в меньшей численности команды с фокусом на разработку для качественного улучшения программного продукта.

Ключевые недостатки облачных услуг:

1. Потенциальные проблемы с безопасностью и конфиденциальностью данных, требующие доверия и гарантий со стороны провайдера, особенно в случаях, когда взаимодействие происходит через сторонние серверы или страны.

2. Риск возникновения проблем с производительностью, доступностью и иных технических сложностей в случае сбоев у поставщика, на которые нельзя повлиять самостоятельно.

3. Зависимость от поставщика – ограничения в настройке инфраструктуры и других частей, которые могут затруднить адаптацию некоторых приложений и ресурсов. Возможные проблемы с совместимостью при миграции между различными облачными платформами.

Таким образом, рассматриваемый подход имеет огромные перспективы по улучшению и большему распространению в разных областях. В большей степени, это связано с активным развитием технологий и цифровизацией человеческой деятельности. Исходя из этого, можно ожидать, что в ближайшее десятилетие роль облачных технологий станет еще более значимой [6].

3. Инструменты на основе машинного обучения

Целевая аудитория – программисты, тестировщики.

Искусственный интеллект и его применение в различных прикладных областях становятся все более распространенными в последние годы, и одной из таких сфер является разработка программного обеспечения. Современные методы и технологии машинного обучения позволяют автоматизировать процессы написания текстов программ, повысить эффективность и качество разработки, а также ускорить время выхода на рынок программных продуктов.

Одной из ключевых возможностей, которую предоставляют «умные» инструменты для написания программного кода, является автоматическое дополнение. Благодаря анализу больших объемов кодовых баз, подобные утилиты могут предлагать разработчику подсказки в процессе работы с текстами программ. Это упрощает специалисту работу, снижает вероятность ошибок и позволяет быть более эффективным при решении задач. Данная функция успешно используется в современных интегрированных средах разработки уже в течение длительного периода, что было отмечено ранее.

Следующим важным аспектом применения инструментов на основе машинного обучения является автоматическое обнаружение и исправление ошибок. Специализированные алгоритмы позволяют анализировать код и выявлять потенциальные проблемы: неправильное использование переменных, несоответствие синтаксису или возможные уязвимости. После обнаружения ошибок утилита может предложить перечень необходимых правок или самостоятельно внести изменения для более эффективного решения проблем.

Инструменты на основе искусственного интеллекта также предоставляют возможности по генерации кода на основе заданных требований. С помощью методов глубокого обучения и генеративных моделей утилита может создавать код, соответствующий определенным спецификациям и шаблонам. Это особенно полезно в случаях, когда требуется большой объем повторяющихся частей или при разработке прототипов, когда нужно быстро получить некоторую функциональность [7-8].

Анализ и оптимизация существующего кода также являются одним из возможных применений инструментов рассматриваемого типа. Алгоритмы машинного обучения и

статистические методы позволяют проводить статический анализ кода, идентифицировать узкие места и неэффективные алгоритмы. Помимо этого, подобные утилиты могут предлагать оптимизированные варианты, которые повысят производительность программы, снизят потребление ресурсов или улучшат общее качество кода.

Последняя ключевая прикладная помощь – возможность автоматического создания документации к коду. Анализируя комментарии, описания и структуру программы, инструмент может автоматически генерировать читаемую и информативную документацию для упрощения понимания и использования кода другими разработчиками.

В качестве обобщения следует отметить, что применение инструментов на основе искусственного интеллекта открывает широкий спектр возможностей для улучшения эффективности, качества и скорости. Несмотря на то, что, на сегодняшний день генерируемые фрагменты не всегда являются полностью готовыми к непосредственному использованию, прослеживается тенденция на совершенствование: определенные утилиты периодически генерируют программный код лучше специалистов среднего уровня. Это значительный прогресс, демонстрирующий, как можно использовать результаты машинного обучения в процессе разработки для более быстрого решения проблем [9].

Заключение

Рассмотренные в данной статье подходы по автоматизации разработки программного обеспечения являются передовыми на сегодняшний день и имеют большие перспективы на улучшение и более широкое распространение на ближайшем этапе развития информационных технологий. Особое внимание следует уделить применению различных облачных услуг и инструментов на основе машинного обучения, чтобы повысить собственную эффективность, снизить стоимость создания программных продуктов и не отставать от технического прогресса, набирающего все большую скорость с каждым годом.

Список литературы

1. Арутюнов, Г. А. Большие трансформеры для генерации кода. / Г. А. Арутюнов, С. М. Авдошин. - Текст : электронный // Труды института системного программирования РАН. - 2022. - № 4 (34). - С. 79-88. URL: <https://elibrary.ru/item.asp?id=50005199> (дата обращения: 02.06.2023). - Режим доступа: Научная электронная библиотека eLIBRARY.RU.
2. Ванясин, Н. В. Архитектура интегрированной среды разработки программного обеспечения с поддержкой структурного редактирования / Н. В. Ванясин, И. Г. Сидоркина, В. И. Поляков. - Текст : электронный // Научно-технический вестник информационных технологий, механики и оптики. - 2019. - № 6 (19). - С. 1079-1085. URL: <https://www.elibrary.ru/item.asp?id=41559161> (дата обращения: 02.06.2023). - Режим доступа: Научная электронная библиотека eLIBRARY.RU.
3. Воронцов, А. И. Анализ различных инструментов управления и мониторинга облачной инфраструктурой / А. И. Воронцов, А. А. Бусенков, О. Д. Куприков. - Текст : электронный // Экономика и качество систем связи. - 2022. - № 2 (24). - С. 65-71. URL: <https://www.elibrary.ru/item.asp?id=48591219> (дата обращения: 02.06.2023). - Режим доступа: Научная электронная библиотека eLIBRARY.RU.

4. Гончаренко, В. А. Актуальные проблемы моделирования конфигураций объектов виртуальной среды единого информационного пространства / В. А. Гончаренко, Д. И. Казанцев. - Текст : электронный // Информация и космос. - 2018. - № 3. - С. 89-95. URL: <https://www.elibrary.ru/item.asp?id=35785169> (дата обращения: 02.06.2023). - Режим доступа: Научная электронная библиотека eLIBRARY.RU.
5. Какутин, Д. Ю. Обзор вспомогательных инструментов на основе машинного обучения для написания исходного кода программ / Д. Ю. Какутин, А. С. Дмитриев, И. А. Абрамов. - Текст : электронный // Электронный научный журнал «Инженерный вестник Дона». - 2022. - № 5 (89). - С. 27-33. URL: <https://www.elibrary.ru/item.asp?id=48925589> (дата обращения: 02.06.2023). - Режим доступа: Научная электронная библиотека eLIBRARY.RU.
6. Матвеев, А. О. Разработка программных средств для улучшения работы механизма автодополнения кода с использованием алгоритмов машинного обучения в интегрированной среде разработки для языка Python / А. О. Матвеев, А. В. Быстров, В. И. Бибаев, Н. И. Поваров. - Текст : электронный // Вестник Новосибирского государственного университета. Серия: Информационные технологии. - 2020. - № 2 (18). - С. 62-75. URL: <https://www.elibrary.ru/item.asp?id=43848163> (дата обращения: 02.06.2023). - Режим доступа: Научная электронная библиотека eLIBRARY.RU.
7. Мейкшан, В. И. Выбор оптимального размещения данных при использовании облачной инфраструктуры / В. И. Мейкшан, Н. Б. Тесля. - Текст : электронный // Журнал "Доклады Академии наук высшей школы Российской Федерации" - содержание выпуска № 2(51) за 2021 год. - 2021. - № 2 (51). - С. 34-42. URL: <https://www.elibrary.ru/item.asp?id=46311534> (дата обращения: 02.06.2023). - Режим доступа: Научная электронная библиотека eLIBRARY.RU.
8. Минакова, О. В. Разработка программных инструментов на базе расширяемых платформ с открытым исходным кодом / О. В. Минакова, Н. В. Акамсина, О. В. Курипта. - Текст : электронный // Вестник Воронежского государственного технического университета. - 2022. - № 4 (18). С. 56-63. URL: <https://www.elibrary.ru/item.asp?id=49369730> (дата обращения: 02.06.2023). - Режим доступа: Научная электронная библиотека eLIBRARY.RU.
9. Мурат, Е. П. Внедрение облачных технологий как вектор развития компании / Е. П. Мурат. - Текст : электронный // Вестник Академии знаний. - 2020. - № 2 (37). - С. 205-211. URL: <https://cyberleninka.ru/article/n/vnedrenie-oblachnyh-tehnologiy-kak-vektor-razvitiya-kompanii> (дата обращения: 02.06.2023). - Режим доступа: Научная электронная библиотека «КиберЛенинка».

References

1. Arutyunov, G. A. Large transformers for code generation. / G. A. Arutyunov, S. M. Avdoshin. - Text: electronic // Proceedings of the Institute for System Programming of the Russian Academy of Sciences. - 2022. - No. 4 (34). - S. 79-88. URL: <https://elibrary.ru/item.asp?id=50005199> (date of access: 06/02/2023). - Access mode: Scientific electronic library eLIBRARY.RU.
2. Vanyasin, N.V., Sidorkina, I.G., Polyakov, V.I. Architecture of an integrated software development environment with structural editing support. - Text: electronic // Scientific and

- technical bulletin of information technologies, mechanics and optics. - 2019. - No. 6 (19). - S. 1079-1085. URL: <https://www.elibrary.ru/item.asp?id=41559161> (date of access: 06/02/2023). - Access mode: Scientific electronic library eLIBRARY.RU.
3. Vorontsov, A. I., Busenkov, A. A., Kuprikov, O. D. Analysis of various tools for managing and monitoring cloud infrastructure. - Text: electronic // Economics and quality of communication systems. - 2022. - No. 2 (24). - S. 65-71. URL: <https://www.elibrary.ru/item.asp?id=48591219> (date of access: 06/02/2023). - Access mode: Scientific electronic library eLIBRARY.RU.
 4. Goncharenko, V. A. Actual problems of modeling the configurations of objects of the virtual environment of a single information space / V. A. Goncharenko, D. I. Kazantsev. - Text: electronic // Information and space. - 2018. - No. 3. - S. 89-95. URL: <https://www.elibrary.ru/item.asp?id=35785169> (date of access: 06/02/2023). - Access mode: Scientific electronic library eLIBRARY.RU.
 5. Kakutin, D. Yu., A. S. Dmitriev, I. A. Abramov, A review of auxiliary tools based on machine learning for writing program source code. - Text: electronic // Electronic scientific journal "Engineering Bulletin of the Don". - 2022. - No. 5 (89). - S. 27-33. URL: <https://www.elibrary.ru/item.asp?id=48925589> (date of access: 06/02/2023). - Access mode: Scientific electronic library eLIBRARY.RU.
 6. Matveev, A. O. Development of software tools for improving the code completion mechanism using machine learning algorithms in an integrated development environment for the Python language / A. O. Matveev, A. V. Bystrov, V. I. Bibaev, N. I. Povarov. - Text: electronic // Bulletin of the Novosibirsk State University. Series: Information technologies. - 2020. - No. 2 (18). - S. 62-75. URL: <https://www.elibrary.ru/item.asp?id=43848163> (date of access: 06/02/2023). - Access mode: Scientific electronic library eLIBRARY.RU.
 7. Meikshan, V. I. The choice of optimal data placement when using cloud infrastructure / V. I. Meikshan, N. B. Teslya. - Text: electronic // Journal "Reports of the Academy of Sciences of the Higher School of the Russian Federation" - content of issue No. 2 (51) for 2021. - 2021. - No. 2 (51). - S. 34-42. URL: <https://www.elibrary.ru/item.asp?id=46311534> (date of access: 06/02/2023). - Access mode: Scientific electronic library eLIBRARY.RU.
 8. Minakova, O. V. Development of software tools based on extensible open source platforms / O. V. Minakova, N. V. Akamsina, O. V. Kuripta. - Text: electronic // Bulletin of the Voronezh State Technical University. - 2022. - No. 4 (18). pp. 56-63. URL: <https://www.elibrary.ru/item.asp?id=49369730> (date of access: 06/02/2023). - Access mode: Scientific electronic library eLIBRARY.RU.
 9. Murat, E. P. Implementation of cloud technologies as a vector of company development / E. P. Murat. - Text: electronic // Bulletin of the Academy of Knowledge. - 2020. - No. 2 (37). - S. 205-211. URL: <https://cyberleninka.ru/article/n/vnedrenie-oblachnyh-tehnologiy-kak-vektor-razvitiya-kompanii> (date of access: 06/02/2023). - Access mode: Scientific electronic library "KiberLeninka".
-