

ИНЖЕНЕРНЫЙ ПОДХОД В РАЗРАБОТКЕ КВАНТОВОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Ваныкина Галина Владиславовна, к.пед.н., доцент,

Сундукова Татьяна Олеговна, к.пед.н., доцент,

Институт повышения квалификации и профессиональной переподготовки работников образования Тульской области, Тула, Россия,

ENGINEERING APPROACH TO DEVELOPING QUANTUM SOFTWARE

Vanykina Galina Vladislavovna, PhD, dist-edu@yandex.ru,

Sundukova Tatyana Olegovna, PhD, sto-ata@yandex.ru,

Institute of Advanced Training and Professional Retraining of Educators of the Tula Region, Tula, Russia

В статье представлена концепция жизненного цикла квантовой инженерии программного обеспечения, которая включает инженерию квантовых требований, проектирование, реализацию, тестирование и поддержку квантового программного обеспечения.

Ключевые слова: квантовые вычисления, кубиты, программное обеспечение, инженерия квантового программного обеспечения, жизненный цикл разработки квантового программного обеспечения.

The article presents the concept of quantum software engineering life cycle, which includes quantum requirements engineering, design, implementation, testing and support of quantum software.

Keywords: quantum computing, qubits, software, quantum software engineering, quantum software development life cycle.

Приоритеты в развитии технологий цифрового общества значительно ускорили применение алгоритмических решений в различных областях, включая искусственный интеллект, криптографию и обработку больших данных. Увеличение сложности структур, объемов обрабатываемых данных, неопределенности при постановке задач создают для классических вычислений существенные ограничения, особенно когда речь идет о решении сложных вычислительных задач (например, моделирование молекулярных взаимодействий, оптимизация крупных сетей или факторизация больших чисел, используемая в криптографии). Выявленные сложности свидетельствуют о том, что в настоящее время выделяется область классов прикладных задач, которые на традиционных технических средствах с классическими алгоритмами демонстрируют значительную сложность, иногда невозможность ввиду длительности по времени.

Квантовые вычисления (QC – quantum computing) в данном контексте можно рассматривать как новое направление, принципиально иную концепцию со значительным потенциалом, в частности, в решении задач оптимизации. Квантовые вычисления от чисто научной концепции переходят в категорию практико-ориентированной коммерчески жизнеспособной технологии, демонстрируя в некоторых вопросах преимущество перед традиционными вычислениями. Крупные технологические компании за последние годы инвестировали значительные средства в создание программных фреймворков (frameworks) и аппаратного обеспечения для разработки приложений, специально предназначенных для квантовых вычислений [1, 2, 3]. Развитие аппаратного обеспечения ускоряется, однако возникает необходимость в программно-интенсивных методологиях, подходах, процедурах, инструментах, а также в определении ролей и обязанностей для создания промышленно-ориентированных квантовых приложений.

Основным отличием методологии квантовых технологий от традиционных является использование кубитов или квантовых битов вместо дискретных статичных битов, на которых построена современная технология обработки данных. Применение кубитов позволяет «обрабатывать» информацию экспоненциально быстрее, чем это возможно на классических компьютерах, благодаря свойствам квантовой физики, таким как квантовая запутанность и квантовая суперпозиция» [2, 4]. Современные исследования в области квантовых вычислений раскрывают потенциальные воз-

возможности использования фотонных процессоров, приближая тем самым момент квантового превосходства, когда квантовый компьютер сможет выполнять с требуемой точностью и стабильностью отдельные классы задач быстрее, чем классический компьютер. Приложения, основанные на квантовых технологиях, требуют пересмотра как инженерии программного обеспечения, аппаратных средств реализации, так и вопросов информационной безопасности.

В связи с проектированием квантовых компьютеров возрастает важность разработки «подходящих квантовых приложений и программного обеспечения, их интеграции в существующие программные архитектуры» [4]. Обеспечение стабильности является на данный момент сложной задачей, требующей «экспертных знаний в различных областях, включая математику, компьютерные науки и физику» [3].

Квантовая инженерия программного обеспечения (QSE – Quantum software engineering) – это популярная, востребованная и динамично развивающаяся в данный момент область научных исследований, которая изучает симбиотические концепции и лучшие практики для создания, поддержки и развития квантовых приложений [2, 3]. В традиционной программной инженерии жизненные циклы разработки программного обеспечения (SDLC – Software development lifecycles) часто используются для описания различных этапов создания программного продукта, через которые проходит приложение [1] и могут обучать разработчиков, предоставляя общее понимание процесса проектирования и/или формируя опыт сотрудничества с экспертами из различных предметных областей.

Учитывая опыт традиционных вычислений, создание надежного программного обеспечения требует соблюдения SDLC, который обычно включает этапы сбора требований, проектирования и моделирования, разработки, тестирования, внедрения и сопровождения. Поскольку разработка квантового программного обеспечения находится на начальной стадии, для него в настоящий момент не существует конкретного SDLC. Для разработчиков квантового программного обеспечения требуются инновационные методы, процедуры, инструменты и стратегии, которые специально ориентированы на создание приложений, использующих принципы квантовой физики. Фундаментальные свойства кубита в квантовой физике (квантовая запутанность, суперпозиция, дегенерация) создают сложности в реализации на практике квантовых алгоритмов. Поскольку проектирование квантового программного обеспечения является наиболее важным этапом в создании квантовых систем, существует огромная потребность в новых принципах и подходах в вопросах SDLC.

QSE должна предоставлять методы для разработки квантовых практико-ориентированных приложений и опираться на проверенные временем этапы: проектирование квантовых алгоритмов, методы их реализации, тестирование и сопровождение квантового программного обеспечения. По мнению исследователей, QSE должна перейти к «инкрементальному, итеративному и гибкому процессу разработки, следуя традиционному подходу в программировании, который начинался с аппаратно-ориентированных методологий и развился в современные гибкие и итеративные методы разработки» [4, 5]. Учитывая опыт SDLC традиционных разработок, целесообразно предложить новые подходы QSE (с поддержкой инструментов), которые на основе классических методов и принципов традиционной программной инженерии позволят выстроить SDLC для квантовых вычислений.

Создание и проектирование квантовых программных систем представляет собой «революционную технологию, которая требует новой парадигмы в программной инженерии» [5]. По мнению исследователей, в QSE разработка квантового кода не является главной проблемой; гораздо более актуальными и требующими решения являются вопросы проектирования и описание требований [2]. Обсуждение методологий разработки квантового программного обеспечения необходимо проводить «в контексте требований QSE, выходящих за рамки написания кода» [5]. Разработка квантового программного обеспечения «включает в себя не только квантовое программирование» [4], поэтому для решения прикладных задач «необходимо применять совершенно новый подход, включая жизненный цикл» [5]. На основании международного опыта развития технологий квантовых вычислений, предложим систематизацию этапов SDLC [2, 5], приведенную в таблице.

Мы считаем, что инженерия требований для квантового программного обеспечения будет похожа на процесс инженерии требований в классических вычислениях, поскольку основное внимание уделяется аспектам сбора и управления требованиями. Однако для моделирования аспектов QC, таких как логические функции и состояния, потребуются новые методы моделирования и

спецификации. Мы полагаем, что исследовательскому сообществу в области инженерии требований необходимо расширить классические методы использования сценариев, пользовательских историй и моделирования целей для «поддержки процесса инженерии квантовых требований» [4].

Таблица – Этапы жизненного цикла разработки квантового ПО

Этап	Категория	Описание
Инженерия требований	Функциональные	• Определение квантовых функций • Фокус на пользовательских требованиях с учетом квантовых аспектов • Описание квантовых механизмов в сценариях использования • Уровни аутентификации и авторизации
	Нефункциональные	• Определение квантовых свойств • Фокус на ожиданиях пользователей с учетом квантовых технологий • Квантовые характеристики как атрибуты качества • Удобство использования, надежность, масштабируемость, производительность
Проектирование	Высокоуровневое	• Абстрактные спецификации квантовых систем • Разработка квантовых модулей • Описание функциональности каждого модуля • Понимание квантовых вентилей
	Низкоуровневое	• Проектирование структур данных для квантовых систем • Проектирование квантовых схем и алгоритмов • Проектирование квантовых интерфейсов • Взаимодействие модулей
Реализация		• Перевод квантовых схем в программный код • Выбор и использование квантового языка программирования • Тестирование с использованием квантовых симуляторов • Запуск программы на сервере
Тестирование		• Разработка планов тестирования • Описание тестовых сценариев • Отчеты и логи тестирования • Измерение кубитов • Модульное и приемочное тестирование
Сопровождение		• Определение необходимости сопровождения • Определение объема сопровождения • Сопровождение архитектурного дизайна квантовых систем • Сопровождение квантовых модулей • Модульное и приемочное тестирование

Рассмотрим два основных этапа проектирования квантового ПО. *Архитектурное проектирование* представляет собой уровень абстракции проектирования, на котором предлагается модульное построение решения и «описывается взаимодействие основных компонентов» [5]. *Детальное проектирование* «явно описывает структуру данных, алгоритмы и интерфейсы для взаимодействия модулей» [5]. В настоящий момент сложности в разработке алгоритмов для проектирования квантового ПО возникают из-за фундаментальных особенностей квантовых вычислений, таких как суперпозиция, запутанность и декогерентность.

В области разработки *квантовых языков программирования* в настоящее время предложены решения [2, 4]. Разработаны несколько квантовых языков программирования, таких как C (QCL), C++ (Scaffold), C# (Q#), Python (ProjectQ, Qiskit, Forest), F# (LIQUi)), Scala (Chisel-Q) и Haskell (Quipper) [2, 4, 5]. Однако исследовательскому сообществу необходимо сосредоточиться на созда-

нии коммерческих интегрированных сред разработки.

Целью этапа *тестирования* является поиск дефектов и проверка поведения системы. При тестировании квантового программного обеспечения в современных реализациях возможны ошибки из-за особенностей квантовых компьютеров, таких как запутанность, суперпозиция и отсутствие клонирования, что делает прогноз поведения квантового программного обеспечения сложной задачей [5]. Существует необходимость в различных инструментах и методах тестирования, которые учитывают характеристики квантовых вычислений, такие как чтение промежуточных состояний [2], обработка вероятностных тестовых прогностических моделей и решение проблем декогеренции.

Этап *сопровождения* включает обновление, изменение и модификацию квантовых приложений для соблюдения требований заказчиков. На наш взгляд, сопровождение можно рассматривать как новую область исследований в QSE с фокусом на реинжиниринге существующих классических решений и их взаимодействии с квантовыми алгоритмами. Существующие на данный момент концепции призваны разграничивать области применения квантовых и традиционных технологий, оптимизируя выбор. В краткосрочной перспективе квантовые компьютеры могут не предоставлять полноценных функций из-за высокой стоимости элементной базы, отсутствия процессов, инструментов и методов достижения устойчивого состояния кубитов, неполноты стохастических моделей для построения прогнозов и учета ошибок [4, 5]. Коллаборация квантовых алгоритмов и существующих классических программных решений является перспективным направлением в науке и технологиях. В данном контексте важно пересмотреть концепции реинжиниринга, областей применения алгоритмических решений, кадрового обеспечения.

Список использованных источников

1. Altman, E. Quantum simulators: Architectures and opportunities / E. Altman et al. // PRX Quantum. – 2021. – Vol. 2, No. 1. – Art. No. 017003.
2. De Stefano, M. Software engineering for quantum programming: How far are we? / M. De Stefano et al. // Journal of Systems and Software. – 2022. – Vol. 190. – Art. No. 111326.
3. Khan, A. Software architecture for quantum computing systems / A. Khan et al. // Journal of Systems and Software. – 2023. – Vol. 201. – Art. No. 111682.
4. Nita, L. The challenge and opportunities of quantum literacy for future education and transdisciplinary problem-solving / L Nita et al. // Research in Science & Technological Education. – 2023. – Vol. 41, No. 2. – P. 564-580.
5. Zhao, J. Quantum software engineering: Landscapes and horizons / J. Zhao. – 2020. – arXiv:2007.07047