

К ВОПРОСУ ОБ ИСПОЛЬЗОВАНИИ АРТЕФАКТОВ, ПРИ РАЗРАБОТКЕ НОВОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Могучев Борис Дмитриевич, кандидат экономических наук, доцент кафедры «Гуманитарных и социальных дисциплин» МГОТУ

Технологический университет, Московская область, г. Королев, Россия,
sadpoet@mail.ru, <https://orcid.org/0000-0003-2757-7523>

***Аннотация.** На сегодня, с учетом сложившейся в мире ситуации из-за коронавируса, многие компании вынуждены искать новые пути своего развития, используя новые виды деятельности. Одним, как раз из таких видов деятельности является компьютерная тематика и все что с ней связано. В частности, это разработка нового компьютерного обеспечения, поиск новых решений в данной сфере. Как раз таким вариантом может быть использование, так называемых «артефактов», при разработке нового программного обеспечения. Данная научная работа посвящена вопросам их использования в современных реалиях. Автором сформулированы определенные задачи, для успешной реализации данной концепции, даны методические рекомендации для последующих работ в данной области, сформулированы основные понятия и предположения.*

Ключевые слова: артефакты, программное обеспечение, гибкий подход, менеджмент.

On the use of artifacts in the development of new software

Moguchev Boris Dmitrievich, Candidate of Economic Sciences, Associate Professor of the Department of Humanities and Social Sciences, MGOTU

Technological University, Moscow region, Korolev, Russia,
sadpoet@mail.ru, <https://orcid.org/0000-0003-2757-7523>

***Annotation.** Today, taking into account the current situation in the world due to the coronavirus, many companies are forced to look for new ways of their development using new types of activities. One of these types of activities is computer topics and everything related to it. In particular, it is the development of new computer software, the search for new solutions in this area. Just such an option may be the use of so-called "artifacts" in the development of new software. This scientific work is devoted to the issues of their use in modern realities. The author formulated certain tasks for the successful implementation of this concept, gave methodological recommendations for subsequent work in this field, formulated the basic concepts and assumptions.*

Keywords: artifacts, software, flexible approach, management.

Крупномасштабные исследования для разработки программного обеспечения в различных компаниях являются сложными, с долгими сроками и высоким рисками ошибки. Несмотря на трудности, связанные с координацией работы нескольких групп разработчиков, применяются гибкие методы. Гибкие процессы адаптированы для поддержки координации групп. Артефакты - это продукты процесса разработки программного обеспечения, предназначенные для обеспечения согласованности подхода групп по одной и той же программе разработки. [1]

Данная научная работа направлена на углубление понимания того, как процессы разработки адаптированы для удовлетворения потребностей крупномасштабных программ разработки программного обеспечения, путем сосредоточения внимания на кадастрах артефактов, используемых в процессе разработки. Автор задается вопросом, что необходимо реализовать при разработке нового программного обеспечения, чтобы артефакты могли помочь в этом процессе. [2]

В качестве основного метода при решении данного вопроса мог бы использоваться анализ данных, основанный на теории концепции открытого кодирования, постоянного сравнения и насыщения.

Основным постулатом данной научной работы является то, что программы развития в этом исследовании создают гибкие и основанные на планах артефакты для улучшения соблюдения стандартов качества предприятия и технологических стратегий, одновременно снижая риск сбоев. Управление этими дополнительными артефактами в настоящее время импровизировано, поскольку в процессах гибкого развития отсутствуют соответствующие церемонии.

Практики, управляющие крупномасштабными программами разработки программного обеспечения, по-видимому, становятся все более привлекательными для сочетания элементов как основанных на планах, так и гибких методов разработки программного обеспечения, в результате чего происходит прагматичная адаптация гибких методов, которые учитывают организационные ограничения, требования к управлению и географическое распределение. [3]

В качестве способа избежать сбоев проекта могут быть предложены гибкие методы. Риск отказа проекта снижается каждый раз, когда предоставляется приращение программного обеспечения, поскольку для разработки в течение каждого приращения выбираются требования наивысшего приоритета, и каждое приращение используется для сбора отзывов клиентов и пользователей. Приращения происходят регулярно и каждый из участников процесса, содержит тщательно обработанный фрагмент общей разработки. Это контрастирует с методами, основанными на планах, в которых риски постепенно возрастают до передачи продукта в конце проекта. Есть доказательства того, что гибкие методы улучшают как производительность разработки программного обеспечения, так и качество продукта. Однако такие подвижные способы обычно ассоциируются с небольшими совместно расположенными командами разработчиков. [4]

Масштабирование гибких методов в крупномасштабные программы разработки программного обеспечения со временем привлекло интерес практиков, и было определено в качестве приоритетной области для исследователей. Например, подход «scrum of scrums» поддерживает несколько параллельных групп «scrum». Команды, работающие параллельно друг с другом, нуждаются в координации, следовательно, мастера «scrum» из каждой команды работают вместе, чтобы координировать действия, управлять зависимостями и избегать совместных ошибок.

Аутсорсинг - это процесс закупки продуктов или услуг у стороннего поставщика или поставщика. При береговом аутсорсинге сторонний поставщик располагается на той же территории, что и организация-клиент. Аутсорсинг включает в себя географически удаленного стороннего поставщика, часто отделенного от клиентской организации временным расстоянием. В отличие от этого, некоторые организации-клиенты создают собственные центры развития. Данные центры

(будь то собственные или аутсорсинговые) могут помочь установить присутствие на развивающихся рынках или воспользоваться их ожидаемой более низкой базой затрат.[5]

Основное внимание в решении указанной проблемы могло бы уделяться крупномасштабным программам развития, включающим разрабатываемую систему или интегрированный портфель соответствующих продуктов. Крупномасштабные системы неизбежно предполагают интеграцию новых функций в существующую кодовую базу, иногда называемую устаревшей системой. [6]

Различные формы артефакта используются для ведения переговоров, регистрации и распространения информации о нарушениях, совершенных в ходе процесса развития. Артефакт — это осязаемый продукт или побочный продукт, производимый при разработке программного обеспечения, обычно включающий в себя модели, конструкции, отчеты и исходный код. Артефакты выступают в качестве объектов границы между техническими специализациями заинтересованных сторон, участвующих в программах развития. [7]

Команды разработчиков должны записывать проектные решения и обмениваться ими, чтобы избежать дублирования и разрешить зависимости. Артефакты фиксируют результаты переговоров между группами заинтересованных сторон, принятые решения и пересмотренные решения. Письменные записи противоречат гибкому манифесту, в котором пропагандируется сосредоточение внимания на работе программного обеспечения над всеобъемлющей документацией. Таким образом, крупномасштабные оперативные программы развития диктуют формы документации для координации деятельности групп и групп, но в то же время оперативные методы пропагандируют сосредоточение внимания на рабочем кодексе. Отсюда следует вывод что, артефакты представляют собой область напряженности между традиционными методами, основанными на планах, и гибкими методами. Как следствие, артефакты могут дать представление об адаптации гибких методов в рамках крупномасштабных программ развития. [8]

Чтобы улучшить понимание процесса разработки программного обеспечения, адаптированного в крупномасштабных программах гибкой разработки программного обеспечения, необходимо исследование взаимодействия практиков с используемыми артефактами. Главный вопрос, возникающий при решении данной проблемы, заключается в следующем: «как практикующие специалисты описывают перечень артефактов, которые они используют в крупномасштабных программах разработки программного обеспечения?»

Этот первичный вопрос исследования должен изучаться дополнительно с использованием двух вспомогательных вопросов исследования:

1) «как артефакты сопоставляются с процессами разработки программного обеспечения, используемыми в крупномасштабных программах разработки программного обеспечения?»

2) «как эти описания практиков способствуют нашему пониманию артефактов в гибком методе настройки в крупномасштабных программах разработки программного обеспечения?».

Чтобы ответить на эти вопросы исследования, необходимо провести качественные эмпирические исследования минимум с десятком международными компаниями, участвующими в крупномасштабных программах разработки гибкого программного обеспечения; чтобы в свою очередь привело к различным открытым полу структурированным интервью с практиками, начиная от старших руководителей и заканчивая начинающими тестерами и разработчиками. Кроме того, необходимо рассмотреть документальные источники, описывающие стандарты и руководящие принципы процесса деинсталляции, и проведены наблюдения на рабочем месте.[9]

Основным итогом при проведении всех вышеописанных процедур может стать систематическое описание взаимодействия практикующих исследователей с артефактами, созданными в крупномасштабных программах гибкой разработки мягких программ. Проектные группы обычно используют несколько гибких методов, в частности: ежедневные совещания по резервированию, краткие итерации, приоритетные задержки, планирование итераций, ретроспективы и планирование выпуска.

Из эмпирических данных, полученных бы в ходе исследования, необходимо выделить категории артефактов. Затем установить таксономия, которая связывает эти артефакты с их ролью в процессе разработки программного обеспечения. Участник процесса разработки определяется как основной владелец каждого артефакта. Кроме того, источники информации и потребители информации для каждой артефакта получены из данных, полученных в ходе этого исследования. Предполагается, что подвижные процессы являются отсутствующими элементами для управления определенными артефактами и что подвижные процессы должны быть улучшены с помощью дополнительных усилий для управления этими артефактами в крупномасштабных программах разработки программного обеспечения. [10]

Библиография

- 1) Abrahamsson, P., Oza, N., & Siponen, M. (2010). Agile Software Development Methods: A Comparative Review. In T. Dingsøyr, T. Dybå, & N. B. Moe (Eds.), *Agile Software Development: Current Research and Future Directions* (. Springer.
- 2) Chau, T., Maurer, F., & Melnik, G. (2003). Knowledge sharing: agile methods vs. Tayloristic methods. In *Proceedings of the Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, Linz, Austria, 9-11 June, 2003 . IEEE.
- 3) Herbsleb, J. D., & Moitra, D. (2001). Global software development. *IEEE Software*, 18(2).
- 4) Méndez Fernández, D., Penzenstadler, B., Kuhmann, M., & Broy, M. (2010). A Meta Model for Artefact-Oriented: Fundamentals and Lessons Learned in Requirements Engineering. In D. C. Petriu, N. Rouquette, & H. Oystein (Eds.), *Model Driven Engineering Languages and Systems - Proceedings of the 13th International Conference on Model Driven Engineering Languages and Systems*, Oslo, Norway, 3-8 October, 2010. Springer.
- 5) Sharp, H., Robinson, H., & Petre, M. (2009). The role of physical artefacts in agile software development: Two complementary perspectives. *Interacting with Computers*, 21(1-2).
- 6) Stapleton, J. (1997). *DSDM, Dynamic Systems Development Method: The Method in Practice*. Addison Wesley Publishing Company.
- 7) Weerd, I. van de, & Brinkkemper, S. (2008). Meta-Modeling for Situational Analysis and Design Methods. In M.R. Syed & S.N. Syed (eds.), *Handbook of Research on Modern Systems Analysis and Design Technologies and Applications* . IGI Global.
- 8) Xu, L., & Brinkkemper, S. (2007). Concepts of product software. *European Journal of Information Systems*.

- 9) Yanzer Cabral, A., Blois Ribeiro, M., Lemke, A. P., Silva, M. T., Cristal, M., & Franco, C. (2009). A case study of Knowledge Management usage in agile software projects. In J. Fil-ipe & J. Cordeiro (Eds.), *Enterprise Information Systems - Proceedings of the 11th International Conference on Enterprise Information Systems*, Milan, Italy, 6-10 May, 2009. Springer.
 - 10) Yin, R. K. (2009). *Case Study Research: Design and Methods* (4th ed.). Sage Publications, Inc.
-