

О СИСТЕМЕ СТРУКТУР ЖИЗНЕННОГО ЦИКЛА РАЗРАБОТКИ ИТ-ПРОДУКТОВ

© 2021 Ю. П. Преображенский, О. Н. Чопоров, Е. Ружицкий

*Воронежский институт высоких технологий (Воронеж, Россия)
Воронежский государственный технический университет (Воронеж, Россия)
Панъевропейский университет (Братислава, Словакия)*

В данной статье рассматривается система структур жизненного цикла разработки ИТ-продуктов.

Ключевые слова: управление, ИТ-продукт, жизненный цикл, процесс.

Модель жизненного цикла ИТ-продукта – это схема, которая объясняет, как будут выполняться действия по разработке ИТ-продукта, его эксплуатации и сопровождению (включая вывод из эксплуатации) посредством описания соответствующих действий.

В настоящее время процесс поиска оптимальных моделей и методологий разработки ИТ-систем продолжается, о чем свидетельствуют публикации в научных изданиях и материалах международных конференций.¹

В жизненном цикле процесса создания автоматизированной системы (АС) можно отметить такие стадии:

- формирование требований относительно АС;
- проведение разработки концепции АС;
- подготовка технического задания [1];
- разработка эскизного проекта;
- реализация технического проекта;
- разработка рабочей документации;
- ввод в действие;
- осуществление сопровождения АС.

Они, в свою очередь, делятся на этапы (действия), результатами которых являются определенные документы, в том числе регламентируемые другими стандартами, например, Единой системой программной документации [3] и Единой системой конструкторской документации [4].

При этом следует понимать, что выше отмечено лишь небольшое количество обя-

зательных этапов создания системы, что позволяет его успешно масштабировать как к более простым проектам, так и к более сложным [5, 6].

К моделям с параллельным планированием фаз следует отнести V-образную модель и ее модификации. Основные идеи, положенные в основу V-образной модели, нашли свое отражение в международном стандарте ISO/IEC 12207. Кроме того, разработанная на ее основе модель V-Modell 97 и далее V-Modell XT стала национальным стандартом ФРГ на разработку ИТ-систем.

Считается, что идею прототипирования при создании ПО предложил Фредерик Брукс в известном труде «Мифический человек-месяц» [3, 4], хотя непосредственно сам Брукс считал, что эту идею предвосхитил Винстон Ройс – создатель каскадной модели.

Тем не менее, в своей книге «Мифический человек-месяц» именно Фредерик Брукс писал: «В большинстве проектов первой построенной системой с трудом можно пользоваться. Она может быть слишком медленной, слишком большой, неудобной в использовании, а то и все вместе. Не остается другой альтернативы, кроме как, поумнев, начать все с начала и построить перепроектированную версию, в которой эти проблемы решены» [3, 4].

Именно эта идея лежит в основе эволюционных, инкрементных, спиральных моделей и модели прототипирования. Исходя из рассмотренного опыта, следует отметить, что модель прототипирования пользуется большой популярностью среди студентов, поскольку она, с их точки зрения, позволяет быстро приступить к реализации системы и не отягощает «бумажной работой».

На сегодняшний день для большинства малых ИТ-компаний характерно использование неформальных методов разработки

Преображенский Юрий Петрович – Воронежский институт высоких технологий, канд. техн. наук, профессор, Petrovich@vvt.ru.

Чопоров Олег Николаевич – Воронежский государственный технический университет, профессор, choporov_oleg@mail.ru.

Ружицкий Евгений – Панъевропейский университет, г. Братислава, Словакия, канд. техн. наук, доцент, rush_ev_g_br53@yandex.ru.

ИТ-продуктов, особенно если коллективы таких компаний формируются за счет молодых людей, зачастую с незаконченным образованием.

Такие методы известны под общим названием «Agile development» или «гибкие методы разработки», благодаря так называемому «Манифесту Agile», опубликованному в феврале 2001 г. группой из 17 ведущих разработчиков ПО, собравшихся в штате Юта (США) с целью обсуждения облегченных методов разработки [3, 4].

В таких случаях сложно говорить о следовании компанией какой-либо определенной модели жизненного цикла (ЖЦ) разработки ПО.

По сути, такая разработка попадает в класс итеративных моделей, характеризующихся разработкой продукта в ходе нескольких итераций, где непрерывно и параллельно идут процессы анализа и корректировки результатов работы [5, 6].

В последнее время наиболее известным и популярным методом такой "быстрой" разработки является методология Scrum.

Создатели методологии отмечают, что ключевым принципом Scrum является принятие разработчиком факта возможности изменений требований заказчика в ходе работы, смены его понимания целей проекта в ходе его выполнения.

Поэтому в данной методологии, вместо погружения в понимание проблемы заказчика на начальных стадиях, выполняющая проект команда фокусируется на возможностях предоставить продукт или, как минимум его часть, как можно быстрее, а затем срочно отреагировать на вновь появившиеся (уточненные) требования [7, 8].

В отличие от основных моделей жизненного цикла разработки ПО, методология Scrum не предписывает определенную последовательность разработки продукта и не определяет состав работ [9, 10].

Вместо этого данная методология жестко фиксирует время итерации («спринта»), за которое должны быть получены конкретные результаты, так называемый «инкремент продукта» (англ. potentially shippable increments, PSIs).

При этом состав требуемых компонентов и функциональности «инкремент продукта» определяется командой непосредственно перед началом «спринта», с помощью инструмента, называемого «беглог проекта» - перечня требований [9, 10], поддерживающий их приоритизацию по важности для стейкхолдеров проекта.

Сами требования к продукту записываются в виде «пользовательских историй» (англ. user stories) – специального шаблона, применяемого и в других методологиях Agile: «Как я хочу, чтобы получить».

Методология Scrum подразумевает повторение «спринтов» до того, пока все требования, описанные в «беглоге проекта», не будут реализованы в поставляемом продукте или в принятой терминологии «сожжены» (англ. burned).

К синтетическим моделям ЖЦ отнесем такие модели, в основу которых положен итеративный принцип разработки, включающий в себя циклическое повторение определенных фаз в жизненном цикле продукта [11, 12], но при этом количество таких повторений и объем работ (необходимые артефакты и их качество) были бы зафиксированы, как в последовательных (водопадных) моделях [13, 14].

Сама компания Microsoft дает следующее определение: «MSF представляет собой набор моделей, принципов и рекомендаций по проектированию и разработке решений масштаба предприятия, который позволит вам успешно управлять такими составляющими проекта, как люди, процессы и инструментальные средства» [3, 4].

Термин MSF можно перевести на русский язык как «каркас решений от Майкрософт». Концептуально, MSF состоит из двух моделей: процессов и команд, и трех дисциплин управления проектами, рисками и готовностью [15, 16].

Модель рационального унифицированного процесса (Rational Unified Process, RUP) – рассматривается в виде модели разработки программного обеспечения. Она создана компанией Rational Software, которая стала подразделением компании IBM в 2003 году. Основателями RUP являются трое американских ученых: G. Booch, J. Rumbaugh и I. Jacobson, которые ранее создали язык UML.

Причиной появления данной модели стали проблемы с применением каскадной модели ЖЦ при создании программных продуктов [17]. И хотя уже существовала спиральная модель, предложенная Барри Боем (Barry Boehm) в 1988 г., многие компании ощущали потребность в поиске решения, наиболее подходящего при разработке проектов, использующих объектно-ориентированную парадигму, и способного сохранить преимущества каскадной модели.

ЛИТЕРАТУРА

1. Йордон Э. Управление сложными Интернет-проектами / Э. Йордон. – М.: Лори, 2014. – 344 с.
2. Афонин А. М. Управление проектами: Учебное пособие / А. М. Афонин, Ю. Н. Царегородцев, С. А. Петрова. – М.: Форум, 2010. – 184 с.
3. Володин С. В. Стратегическое управление проектами: на примере аэрокосмической отрасли / С. В. Володин. – М.: Ленанд, 2014. – 152 с.
4. Гонтарева И. В. Управление проектами: Учебное пособие / И. В. Гонтарева, Р. М. Нижегородцев, Д. А. Новиков. – М.: КД Либроком, 2013. – 384 с.
5. Зуб А. Т. Управление проектами: Учебник и практикум для академического бакалавриата / А. Т. Зуб. – Люберцы: Юрайт, 2016. – 422 с.
6. Рожкова Т. С. Подходы к постановке задачи оптимизации распределения ресурсов в вычислительной сети / Т. С. Рожкова, В. В. Афанасьев, И. И. Ветров // Моделирование, оптимизация и информационные технологии. – 2020. – Т. 8. – № 4 (31). – С. 9-10.
7. Лащенко Д. П. Математическое моделирование и оптимизация сложноструктурированных объектов / Д. П. Лащенко // Моделирование, оптимизация и информационные технологии. – 2020. – Т. 8. – № 4 (31). – С. 22-23.
8. Коробкин Д. М. Метод формирования критериальных оценок морфологических признаков технических систем / Д. М. Коробкин // Моделирование, оптимизация и информационные технологии. – 2020. – Т. 8. – № 4 (31). – С. 23-24.
9. Евсин В. А. Математическое и имитационное моделирование закрытого распределенного реестра с управляющим узлом / В. А. Евсин, С. Н. Широкова, С. П. Воробьев, В. А. Евсина // Моделирование, оптимизация и информационные технологии. – 2020. – Т. 8. – № 2 (29). – С. 27-28.
10. Шаповалов А. В. Возможности применения методов оптимизации в управлении портфелями проектов / А. В. Шаповалов, А. П. Преображенский, О. Н. Чопоров // Моделирование, оптимизация и информационные технологии. – 2020. – Т. 8. – № 1 (28). – С. 32-33.
11. Воронов А. А. Обеспечение системы управления рисками при возникновении угроз информационной безопасности / А. А. Воронов, И. Я. Львович, Ю. П. Преображенский, В. А. Воронов // Информация и безопасность. – 2006. – Т. 9. – № 2. – С. 8-11.
12. Львович Я. Е. Адаптивное управление марковскими процессами в конфликтной ситуации / Я. Е. Львович, Ю. П. Преображенский, Р. Ю. Паневин // Вестник Воронежского государственного технического университета. – 2008. – Т. 4. – № 11. – С. 170-171.
13. Преображенский Ю. П. Проблемы управления в производственных организациях / Ю. П. Преображенский // В сборнике: Актуальные проблемы развития хозяйствующих субъектов, территорий и систем регионального и муниципального управления. Материалы XIII международной научно-практической конференции. Под редакцией Ю. В. Вертаковой. – 2018. – С. 208-211.
14. Преображенский А. П. Возможности обеспечения развития предприятий / А. П. Преображенский // В мире научных открытий. – 2015. – № 10 (70). – С. 196-201.
15. Землянухина Н. С. О применении информационных технологий в менеджменте / Н. С. Землянухина // Успехи современного естествознания. – 2012. – № 6. – С. 106-107.
16. Свиридов В. И. Лингвистическое обеспечение автоматизированных систем управления и взаимодействие пользователя с компьютером / В. И. Свиридов, Е. И. Чопорова, Е. В. Свиридова // Моделирование, оптимизация и информационные технологии. – 2019. – Т. 7. – № 1 (24). – С. 430-438.
17. Львович И. Я. Использование информационных технологий в менеджменте / И. Я. Львович, А. П. Преображенский, О. Н. Чопоров // В книге: Инновационная экономика и менеджмент в современном мире. Одесса, 2019. – С. 49-60.

ABOUT THE SYSTEM OF LIFE CYCLE STRUCTURES FOR IT PRODUCTS DEVELOPMENT

© 2020 Yu. P. Preobrazhenskiy, O. N. Choporov, E. Ruzhicky

Voronezh Institute of High Technologies (Voronezh, Russia)

Voronezh State Technical University (Voronezh, Russia)

Pan-European University (Bratislava, Slovakia)

This paper/article discusses the system of structures of the life cycle of IT product development.

Keywords: management, IT product, life cycle, process.