

УДК 004.9

## МЕТОДОЛОГИИ И СТАНДАРТЫ В ТЕХНОЛОГИИ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Шульдова С.Г.<sup>a</sup>

<sup>a</sup> *Минский инновационный университет, кандидат технических наук, доцент кафедры информационных технологий, svetlanash@tut.by*

### **Аннотация**

В статье рассмотрены методологии и стандарты, применяемые в разработке программного обеспечения. Приведены классификации методологий в зависимости от жизненного цикла ПО и условий процесса разработки. Проанализированы составляющие методологий, определены взаимосвязи между ними на этапах разработки ПО. Предложена обобщенная структура артефактов, получаемых в результате выполнения работ на различных этапах ЖЦ ПО. Также выполнен обзор международных стандартов и стандартов Республики Беларусь в сфере разработки ПО.

**Ключевые слова:** программное обеспечение, жизненный цикл, технология, методология, метод, этап, процесс, стандарт.

**Веб:** <http://library.miu.by/journals!/item.science-xxi/issue.5/article.13.html>

**Поступила в редакцию:** 17.10.2016.

## METHODOLOGY AND STANDARDS IN TECHNOLOGY OF SOFTWARE DEVELOPMENT

Shuldava S.<sup>a</sup>

<sup>a</sup> *Minsk Innovation University, PhD in Technical sciences, associate professor of the Department of information technologies, svetlanash@tut.by*

### **Abstract**

Methodologies and standards used in software development are discussed in the article. A classification of methodologies, depending on the life cycle of the software development process and conditions, is presented. The components of methodologies are analyzed, the relationship between them is defined on the stages of software development. The structure of the artifacts resulting from performance of work at different stages of life cycle software is proposed. Also, the overview of international standards and standards of the Republic of Belarus in the sphere of software development is given.

**Keywords:** software, life cycle, technology, methodology, technique, stage, process, standard.

**Web:** <http://library.miu.by/journals!/item.science-xxi/issue.5/article.13.html>

**Received:** 17.10.2016.

## Введение

Сегодня реалии бизнеса требуют от создателей программного обеспечения (ПО) разработки программных продуктов высокого качества в рамках отведенного бюджета и в срок, что достигается применением методичного, упорядоченного, количественно измеримого подхода к проектированию, созданию, эксплуатации и поддержке ПО. Именно такими качествами отличается профессиональный инженерный подход к разработке ПО [1], обусловленный применением современных технологий разработки.

В общем случае, технология разработки программного обеспечения – это система инженерных принципов для создания экономичного ПО с заданными характеристиками качества в установленные сроки [2]. Из определения технологии следует, что создание ПО носит проектный характер и поэтому требует специализированных проектных методологий, призванных обеспечить качество ПО.

## 1. Методологии разработки ПО и их классификация

Представляется важным уточнить понятия «технология» и «методология». Технология разработки ПО представляет собой комплекс организационных мер, приемов и способов осуществления процессов, направленных на разработку программных средств, в то время как методология является формой предписаний и норм, в которых фиксируются содержание и последовательность выполнения этих процессов.

Таким образом, любая технология разработки ПО базируется на некоторой методологии или совокупности методологий, определяемых моделью жизненного цикла (ЖЦ) ПО. К настоящему времени наибольшее распространение получили каскадная, инкрементная и спиральная модели [3]. Выбор модели ЖЦ и реализующей ее методологии зависит от специфики, масштаба и сложности программного проекта, а также специфики условий, в которых ПО создается и функционирует. Предпосылкой успешного выбора методологии является наличие необходимой и достаточной информации о ней самой, практике ее использования и дополнительных рекомендаций по применению.

Наиболее популярными в настоящее время являются методологии [2–8]:

- **Rapid Application Development (RAD)** – быстрая разработка приложений, обеспечивает экстремально короткий цикл разработки, основана на спиральной и инкрементной моделях ЖЦ ПО;

- **Rational Unified Process (RUP)** – рациональный унифицированный процесс разработки, одна из спиральных методологий разработки программного обеспечения. Методология поддерживается компанией Rational Software, которая в настоящее время входит в состав IBM. В качестве языка моделирования в RUP используется язык Unified Modelling Language (UML);

- **eXtreme Programming (XP)** – экстремальное программирование, базируется на спиральной модели жизненного цикла. Разработка представляет собой итеративный процесс, где фазы разбиваются на «экстремально» малые шаги. В основе методологии – командная работа и эффективная коммуникация между заказчиком и исполнителем;

- **Microsoft Solutions Framework (MSF)** – методология разработки программного обеспечения от корпорации Microsoft. MSF опирается на практический опыт корпорации и описывает управление людьми и рабочими процессами в процессе разработки решения. В модели процессов MSF сочетаются каскадная и спиральная модели разработки: проект реализуется поэтапно, с наличием соответствующих контрольных точек, а сама последовательность этапов может повторяться по спирали;

- **Oracle Custom Development Method (CDM)** – методология разработки ПО небольших масштабов на основе Oracle, ориентированная на использование каскадной модели ЖЦ;

- **Scrum** – методология делает акцент на качественном контроле процесса разработки, который состоит из жестко фиксированных и небольших по времени итераций, называемых спринтами (sprints). По окончании спринта конечному пользователю предоставляется работающее ПО с новыми возможностями, для которых определен наибольший приоритет;

- **Crystal Clear** – методология предназначена для разработки бизнес-приложений небольшими командами в 6–8 человек, описывает естественный порядок разработки ПО, который устанавливается в достаточно квалифицированных коллективах, основана на инкрементной модели ЖЦ ПО.

Помимо классификации по модели в зависимости от условий реализации проекта выделяют методологии прогнозируемые («тяжелые») и адаптивные («подвижные») [3].

Прогнозируемые методологии основываются на предпосылке о возможности и целесообразности детального планирования будущего. Для ИТ-проекта формулируются требования к разрабатываемой системе, формируется план проекта и определяется потребность в ресурсах. Изменения в плане проекта и требованиях считаются нежелательными. К «тяжеловесным» относятся классические варианты RAD и RUP-методологий.

Адаптивные методологии нацелены на преодоление предполагаемой неполноты требований, а также их постоянного изменения. Примерами адаптивной методологии являются Extreme Programming, Crystal Clear и Scrum. Адаптивные методологии учитывают психологические особенности процесса разработки ПО. Одним из значимых факторов успеха использования адаптивных методологий является высокая квалификация специалистов, в первую очередь – разработчиков.

Цель методологии разработки ПО – внедрение методов, обеспечивающих достижение соответствующих характеристик качества. Метод в технологии разработки ПО представляет собой совокупность следующих составляющих [2, 3]:

- концепций и теоретических основ. В качестве таких основ выступают структурный, объектно-ориентированный и компонентный подходы к разработке ПО;

- нотаций, используемых для построения моделей статической структуры и динамики поведения проектируемой системы. В качестве таких нотаций обычно используются графические диаграммы: диаграммы потоков данных и диаграммы «сущность-связь» для структурного подхода, диаграммы вариантов использования, диаграммы классов и другие – для объектно-ориентированного и компонентного подходов;

- процедур, определяющих практическое применение метода (последовательность и правила построения моделей, критерии, используемые для оценки результатов).

Современные методологии ориентированы на использование инструментальных средств проектирования, называемых CASE-средствами (Computer Aided Software/System Engineering). В настоящее время к CASE-средствам относят любое программное средство, автоматизирующее ту или иную совокупность процессов ЖЦ ПО. Методы и CASE-средства составляют базис в реализации процессов разработки ПО.

## 2. Стандарты в разработке программного обеспечения

Реальное применение любой технологии разработки ПО предполагает использование стандартов, регламентирующих различные аспекты разработки. Стандарты разрабатываются международными и государственными организациями по стандартизации, отраслевыми комитетами, исследовательскими институтами, каковыми в первую очередь являются [2]:

- ISO (International Organization for Standardization);

- IEEE (Institute of Electrical and Electronics Engineers, Институт Электронной и Электротехнической Инженерии);

- МЭК (Международная Электротехническая комиссия);

- а также крупными ИТ-компаниями: Bell, Hewlett Packard, Sun Microsystems, IBA, Oracle, Microsoft и др.

Объектами стандартизации в сфере ИТ являются:

- конструкторская документация (состав, структура, требования к оформлению);

- терминология и определения;

- модели процессов;

- модели жизненного цикла;

- требования к безопасности хранения и передачи информации и способы ее обеспечения;

- качество программного обеспечения, характеристики качества, методы получения данных по качеству.

По объекту стандартизации можно выделить следующие группы стандартов:

1. Стандарты, определяющие требования к качеству ПО.

2. Стандарты, определяющие требования к документации.

3. Стандарты, определяющие термины по программному обеспечению.

4. Стандарты на процессы жизненного цикла программного обеспечения.

5. Стандарты, определяющие требования к функциональной безопасности ПО.

6. Обучающие стандарты.

В таблице 1 представлены основные действующие международные стандарты и стандарты Республики Беларусь.

Стандарт СТБ МЭК 9126-2003 «Информационные технологии. Оценка программной продукции. Характеристики качества и руководства по их применению» разработан на основе международного стандарта ISO/IEC 9126:1991, который в настоящее время отменен. Аналогичная ситуация наблюдается с большинством стандартов Республики Беларусь. В связи с этим целесообразно использование международных стандартов.

Общие принципы обеспечения качества процессов производства во всех отраслях экономики регулируются набором стандартов серии ISO 9000. Стандарт ISO/IEC 90003:2014 «Software engineering – Guidelines for the application of ISO 9001:2008 to computer software» конкретизирует положения ISO 9001 для разработки программного обеспечения с упором на обеспечение качества при процессе проектирования.

Необходимо отметить, что показатели качества, закрепленные в стандартах, не исчерпывают полностью понятие качества ПО.

## 3. Взаимосвязь составляющих методологии разработки ПО

В соответствии со стандартом ISO/IEC 12207 все процессы ЖЦ ПО разделены на три группы: пять основных процессов (приобретение, поставка, разработка, эксплуатация, сопровождение); восемь вспомогательных процессов, обеспечивающих выполнение основных процессов (документирование, управление конфигурацией, обеспечение качества, верификация, аттестация, совместная оценка, аудит, разрешение проблем); четыре организационных процесса (управление, создание инфраструктуры, совершенствование, обучение).

Данный стандарт имеет весьма общий («рамочный») характер, поскольку определяет не структуру ЖЦ ПО, а только некоторый набор видов деятельности, на основе которых вводят ту или иную структуру ЖЦ ПО.

Этапы создания ПО выделяются по соображениям рационального планирования и организации работ, заканчивающихся заданными результатами (артефак-

Таблица 1 – Стандарты в разработке ПО

Действующий стандарт Республики Беларусь	Действующие международные стандарты
<b>Стандарты качества ПО</b>	
<b>СТБ ИСО/МЭК 9126-2003</b> Информационные технологии. Оценка программной продукции. Характеристики качества и руководства по их применению	<b>ISO/IEC 25010:2011</b> Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models
<b>СТБ ИСО/МЭК 12119-2003</b> Информационные технологии. Пакеты программ. Требования к качеству и тестирование	<b>ISO/IEC 25051:2014.</b> Software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – Requirements for quality of Ready to Use Software Product (RUSP) and instructions for testing
<b>Стандарты, определяющие требования к документации</b>	
<b>СТБ ИСО/МЭК ТО 9294-2003</b> Информационные технологии. Руководство по управлению документированием программного обеспечения	<b>ISO/IEC TR 9294:2005.</b> Information technology – Guidelines for the management of software documentation <b>IEEE 1063-1987</b> Standard for Software User Documentation;
<b>Стандарты, определяющие термины по программному обеспечению</b>	
	<b>IEEE Std 610.12-1990.</b> – IEEE Standard Glossary of Software Engineering Terminology
<b>Стандарты на процессы жизненного цикла программного обеспечения</b>	
<b>СТБ ИСО/МЭК 14764-2003</b> Информационные технологии. Сопровождение программных средств	<b>ISO/IEC 14764:2006</b> Software Engineering – Software Life Cycle Processes – Maintenance
<b>СТБ ИСО/МЭК 12207-2003</b> Информационные технологии. Процессы жизненного цикла программных средств	<b>ISO/IEC 12207:2008</b> Information Technology – Software Life Cycle Processes
<b>Обучающие стандарты</b>	
<b>СТБ ИСО/МЭК 90003-2007</b> Разработка программных средств. Руководство по применению СТБ ИСО 9001-2001 для программных средств.	<b>ISO/IEC 90003:2014</b> Software engineering – Guidelines for the application of ISO 9001:2008 to computer software
<b>Стандарты безопасности</b>	
<b>СТБ ИЕС 61508-4-2014 (ИЕС 61508-4:2010:IDT).</b> Функциональная безопасность электрических, электронных, программируемых электронных систем, относящихся к безопасности.	<b>IEC 61508-4:2010</b> Functional safety of electrical/electronic/ programmable electronic safety-related systems – Part4: Definitions and abbreviations
<b>СТБ 34.101.3-2014 (ISO/IEC 15408-3:2008) (ISO/IEC 15408-3:2008, MOD).</b> Информационные технологии и безопасность. Критерии оценки безопасности информационных технологий. Часть 3. Гарантийные требования безопасности	<b>ISO/IEC 15408-3:2008</b> Information technology – Security techniques – Evaluation criteria for Information technology security – Part 3. Security assurance components

тами). В состав жизненного цикла ПО обычно включаются следующие этапы, адаптируемые к конкретной модели жизненного цикла ПО:

- выявление и анализ требований к ПО;
- проектирование;
- реализация;
- тестирование;
- внедрение;
- эксплуатация и сопровождение.

Этапы и выполняемые на них процессы в общем виде представлены на рисунке 1.

Для каждого этапа определяются состав и последовательность выполняемых работ, получаемые артефакты, методы и средства, необходимые для выполнения работ, роли и ответственность участников в зависимости от конкретной методологии разработки ПО. Взаимосвязь составляющих методологии на определенном этапе ЖЦ ПО представлена на рисунке 2.

Работа, выполняемая участником, подразумевает четко определенную ответственность, четко определенный артефакт и имеет четко определенные границы, устанавливаемые при планировании проекта. В таком контексте работа представляет собой технологическую операцию, упорядоченная и взаимосвязанная совокупность технологических операций объединяется в технологический процесс, совокупность технологических процессов – в технологию создания программного обеспечения в рамках ЖЦ ПО [9].

Под артефактами понимаются информационные или материальные сущности, которые создаются, модифицируются или используются при выполнении работы – объектные модули, исходный код, базы данных, документация, результаты тестов, а также модели, описывающие различные аспекты разрабатываемого ПО [3]. Различные модели, используемые отдельными разработчиками при создании и анализе ПО,

Основной процесс					
Разработка					
Тестирование					
Вспомогательный процесс					
Администрирование					
Обеспечение качества					
Управление конфигурацией					
Организационные процессы					
Управление					
Создание инфраструктуры					
Мониторинг системы					
Объекты					
Обработка и монтаж данных	Проектирование	Разработка	Тестирование	Эксплуатация	Эксплуатация и сопровождение

Рисунок 1 – Этапы разработки ПО и выполняемые на них процессы

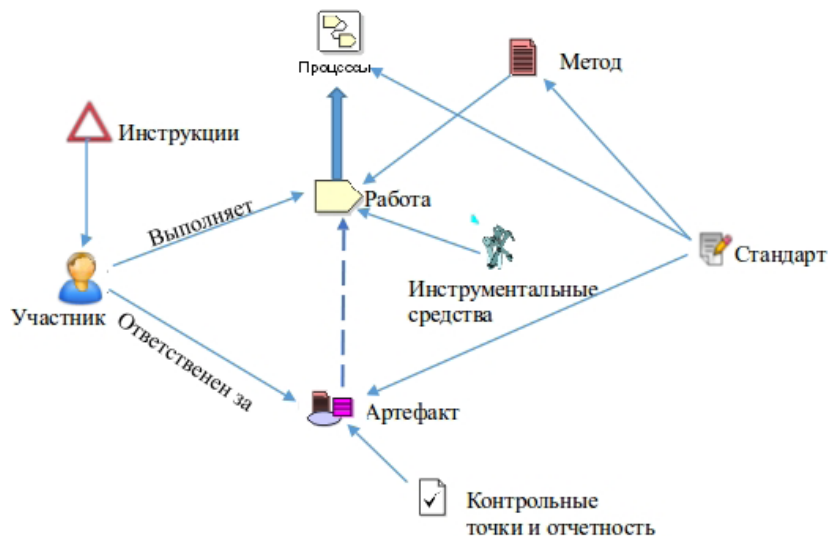


Рисунок 2 – Составляющие методологии разработки ПО и их взаимосвязь

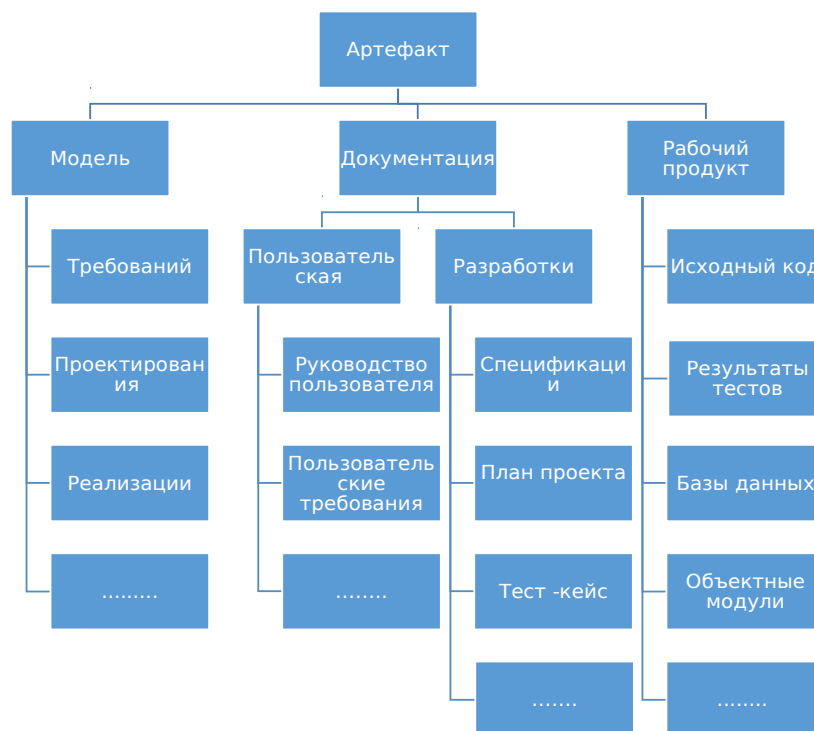


Рисунок 3 – Обобщенная структура артефактов

но не зафиксированные в виде доступных документов, не могут считаться артефактами. Состав артефактов определяется выбранной методологией, обобщенная структура артефактов приведена на рисунке 3.

В создании программных средств принимают участие различные специалисты, которые объединяются в команды. Методологии содержат, как правило, описание команды и ролей, выполняемых ее участниками.

Существует стандартный набор ролей, выполняемых членами проектной команды:

- руководитель проекта;
- бизнес-аналитик (системный аналитик);
- системный архитектор;
- разработчик (программист);
- специалист по тестированию ПО (тестирущик);
- технический писатель;
- инженер по сопровождению ПО.

Для небольших проектов некоторые роли могут быть совмещены и могут выполняться одним человеком. Как правило, в основной состав проектной группы входят аналитик, разработчик и тестирущик.

На основании изложенного выше можно сделать вывод, что процессы, связанные с жизненным циклом ПО, весьма разнообразны и трудны, и помимо специальных знаний, соответствующих роли, тре-

буют наличия у всех участников программного проекта знаний в области технологий разработки, стандартизации и сертификации ПО.

### Заключение

Таким образом, проектные методологии разработки ПО содержат следующий набор элементов: описание модели жизненного цикла разработки ПО, описание ролей проектной команды, описание разрабатываемых артефактов и рекомендации по использованию инструментальных средств, таких как языки графического моделирования, языки программирования и т.д. Основное различие между проектными методологиями связано с применением разных моделей ЖЦ ПО и условий разработки.

Стандарты, регламентирующие требования к процессам разработки и выходным продуктам, дополняют методологии разработки ПО.

Связанные с жизненным циклом процессы разнообразны по своей сути, достаточно сложны, включают в себя много отдельных требующих решения задач и предполагают наличие у участников проекта помимо специальных знаний, определяющих роль в команде проекта, знаний технологии разработки, стандартизации и сертификации ПО.

## ЛИТЕРАТУРА / REFERENCES

1. IEEE Standard Glossary of Software Engineering Terminology. IEEE Standard 610.12-1990. New-York: Institute of Electrical and Electronics Engineers, 1990 [Electronic resource] // Jyväskylä yliopisto. University of Jyväskylä. – Mode of access: [http://www.mit.jyu.fi/ope/kurssit/TIES462/Materiaalit/IEEE\\_SoftwareEngGlossary.pdf](http://www.mit.jyu.fi/ope/kurssit/TIES462/Materiaalit/IEEE_SoftwareEngGlossary.pdf). – Date of access: 12.08.2016.
2. Бахтизин, В.В. Технология разработки программного обеспечения : учеб. пособие / В.В. Бахтизин, Л.А. Глухова. – Минск : БГУИР, 2010. – 267 с.  
Bakhtizin, V.V. Tekhnologiya razrabotki programmnoho obespecheniya: ucheb. posobiye / V.V. Bakhtizin, L.A. Glukhova. – Minsk: BGUIR, 2010. – 267 p.
3. Орлов, С.А. Технологии разработки программного обеспечения / С.А. Орлов, Б.Я. Циклер. – СПб.: Питер, 2012. – 608 с.  
Orlov, S.A. Tekhnologii razrabotki programmnoho obespecheniya / S.A. Orlov, B.Ya. Tsikler. – SPb.: Piter, 2012. – 608 p.
4. Коберн, А. Быстрая разработка программного обеспечения / А. Коберн. – М. : Лори, 2013. – 314 с.  
Kobern, A. Bystraya razrabotka programmnoho obespecheniya / A. Kobern. – M.: Lori, 2013. – 314 p.
5. Кон, М. Scrum: гибкая разработка ПО. Описание процесса успешной гибкой разработки программного обеспечения с использованием Scrum / М. Кон. – М. : Вильямс, 2011. – 566 с.  
Kon, M. Scrum: gibkaya razrabotka PO. Opisaniye protsesssa uspeшноy gibkoy razrabotki programmnoho obespecheniya s ispol'zovaniyem Scrum / M. Kon. – M.: Vil'yams, 2011. – 566 p.
6. Microsoft Corporation. Microsoft Solutions Framework: гибкая методология разработки программного обеспечения. Справочное руководство. – М. : Русская редакция, 2010. – 127 с.  
Microsoft Corporation. Microsoft Solutions Framework: gibkaya metodologiya razrabotki programmnoho obespecheniya. Spravochnoye rukovodstvo. – M.: Russkaya redaktsiya, 2010. – 127 p.
7. Бек, К. Экстремальное программирование: разработка через тестирование / К. Бек. – СПб.: Питер, 2003. – 224 с.  
Bek, K. Ekstremal'noye programmirovaniye: razrabotka cherez testirovaniye / K. Bek. – SPb.: Piter, 2003. – 224 p.
8. Кролл, П. Rational Unified Process – это легко. Руководство по RUP для практиков: пер. с англ. / П. Кролл, Ф. Крачтен. – М.: КУДИЦ-ОБРАЗ, 2004. – 432 с.  
Kroll, P. Rational Unified Process – eto legko. Rukovodstvo po RUP dlya praktikov: per. s angl. / P. Kroll, F. Krachten. – M.: KUDITs-OBRAZ, 2004. – 432 p.
9. Вендров, А.М. Современные технологии создания программного обеспечения. Обзор [Электронный ресурс] / А.М. Вендров // Jet Info Online. – 2004. – № 4. – Режим доступа: <http://www.ietinfo.rU/2004/4/1/article1.4.2004.html>. – Дата доступа: 15.09.2016.  
Vendrov, A.M. Sovremennyye tekhnologii sozdaniya programmnoho obespecheniya. Obzor [Electronic resource] / A.M. Vendrov // Jet Info Online. – 2004. – No. 4. – Mode of access: <http://www.ietinfo.rU/2004/4/1/article1.4.2004.html>. – Date of access: 15.09.2016.