

*Парамзина А.А.
магистерская программа «Системное
и прикладное программное обеспечение»
Тищенко Е.Н., д.э.н.
профессор
информационные технологии и защита информации
Ростовский государственный экономический университет (РИНХ)*

КАЧЕСТВО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Аннотация: Программное обеспечение должно выполнять свои функции, соответствовать заданным критериям качества, безопасности, надежности. Оценка продукта, требований к нему, проектной документации – задача инженеров по обеспечению качества, или QA-инженеров.

Обеспечение качества ПО включает в себя мероприятия, которые проводят на каждой стадии его разработки. Цель – предоставить гарантию того, что продукт соответствует функциональным и нефункциональным требованиям.

Ключевые слова: информационные технологии; анализ; характеристики качества программного обеспечения; модели качества программного обеспечения.

*Paramzina A.A.
master program «System and application software»
Tishchenko E.N., D.Sc. economics
professor
department of information technology and information security
Rostov State University of Economic*

SOFTWARE QUALITY

Abstract: The software must perform its functions, meet the specified criteria of quality, safety, reliability. Evaluation of the product, its requirements, and project documentation is the task of quality assurance engineers, or QA engineers.

Software quality assurance includes activities that are carried out at each stage of its development. The goal is to provide a guarantee that the product meets functional and non-functional requirements.

Keywords: information technology; analysis; software quality characteristics; software quality models.

1 КРИТЕРИИ КАЧЕСТВА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

На первый взгляд, «качество ПО» может показаться абстрактным понятием. Однако для менеджеров проекта, программистов, специалистов по тестированию, QA-инженеров и других участников процесса разработки продукта критерии качества прозрачны и измеримы. Сначала рассмотрим общее определение.

Качество ПО – комплекс характеристик программного продукта, определяющих способность выполнять возложенные на него функции.

В настоящий момент этот показатель регулируется международным стандартом ISO/IEC 25010:2011. Данный стандарт устанавливает многоуровневую систему оценки качества ПО, основанную на восьми базовых характеристиках.

ПАРАМЕТРЫ КАЧЕСТВА ПО

Основные характеристики качества программного обеспечения согласно стандарту ISO/IEC 25010:2011:

Функциональность. ПО признается функциональным, если выполняет возложенные на него задачи, отвечает заданным потребностям пользователей. Данный аспект предполагает правильную и точную работу, совместимость всех входящих в состав компонентов.



Рисунок 1 – Блок-схема характеристики качества программного обеспечения

1 Функциональность

Функциональные возможности — совокупность свойств, определяющих способность программного обеспечения выполнять предусмотренные функции в заданной среде в соответствии с заданными требованиями. Под функцией понимается некоторая упорядоченная

последовательность действий для удовлетворения потребительских свойств. Функции бывают целевые (основные) и вспомогательные.

К атрибутам функциональных возможностей относятся:

- защищенность — атрибут, который показывает способность ПО предотвращать несанкционированный доступ (случайный или умышленный) к программам и данным;
- интероперабельность — атрибут, который показывает возможность взаимодействия данного ПО со специальными системами и средами (операционные системы, вычислительные сети);
- функциональная полнота — атрибут, который показывает меру достаточности основных функций для решения задач в соответствии с назначением данного ПО.

2 Надежность

Надежность. Под надежностью ПО понимают бесперебойное выполнение возлагаемых на него задач на заданных условиях в течение установленного времени.

К атрибутам функциональной надежности ПО относятся:

- безошибочность — атрибут, который определяет способность ПО функционировать без ошибок;
- правильность — атрибут, который показывает меру достижения правильных результатов;
- контролируемость — атрибут, который характеризует полноту и эффективность обнаружения ошибок в промежуточных и выходных результатах
- устойчивость к ошибкам — атрибут, который характеризует способность ПО правильно выполнять функции при аномальных условиях (сбой аппаратуры, ошибки в данных и интерфейсах и др.);
- безотказность — атрибут, который характеризует способность ПО не вызывать функциональные отказы информационной системы;
- пригодность к восстановлению — атрибут, который характеризует способность программы к устранению программной ошибки и к перезапуску для повторного выполнения и восстановления данных в случае функционального отказа;
- готовность — атрибут, который показывает способность программы по произвольной заявке безошибочно выполнить предусмотренное преобразование.

3 Юзабилити

Юзабилити (удобство использования). Этот параметр характеризует степень удобства ПО для пользователей, его наглядность, легкость эксплуатации и изучения.

К атрибутам удобства применения относятся:

– понимаемость — атрибут, который определяет усилия, затрачиваемые на распознавание логических концепций и условий применения ПО;

– изучаемость (легкость изучения) — атрибут, который определяет усилия пользователей, направленные на определение применимости ПО путем использования операционного контроля, диагностики, а также процедур, правил и документации;

– оперативность — атрибут, который показывает реакцию системы при выполнении операций и операционного контроля.

4 Эффективность

1.4 Эффективность

Эффективность. Параметру соответствует степень обеспечения продуктом необходимой производительности при заданных условиях.

К атрибутам эффективности ПО относятся:

– реактивность — атрибут, который показывает время отклика, обработки и выполнения функций;

– эффективность ресурсов — атрибут, показывающий количество и продолжительность используемых ресурсов при выполнении функций ПО;

– согласованность — атрибут, который показывает соответствие данной характеристики заданным стандартам, правилам и предписаниям.

5 Сопровождаемость

Удобство сопровождения. Этот показатель характеризует простоту анализа, тестирования, коррекции компонентов ПО, его обслуживания, а также степень адаптации к новым условиям.

Сопровождаемость включает следующие атрибуты:

– анализируемость — атрибут, определяющий необходимые усилия для диагностики отказов или идентификации частей, которые будут модифицироваться;

– стабильность — атрибут, указывающий на постоянство структуры и риск ее модификации;

– тестируемость — атрибут, указывающий на усилия при проведении валидации и верификации с целью обнаружения несоответствий требованиям, а также на необходимость проведения модификации ПО и сертификации;

– изменяемость — атрибут, который определяет возможность удаления ошибок в ПО или внесение изменений для их устранения, а также введение новых возможностей в ПО или в среду функционирования.

6 Мобильность

Мобильность. Степень легкости его переноса на другую платформу. Обеспечение качества ПО предполагает его проверку по каждому из перечисленных параметров, выявление слабых сторон и устранение неисправностей.

Мобильность включает атрибуты:

– адаптивность — атрибут, определяющий усилия, затрачиваемые на адаптацию к различным средам;

– настраиваемость (простота инсталляции) — атрибут, который определяет необходимые усилия для запуска данного ПО в специальной среде;

– сосуществование — атрибут, который определяет возможность использования специального ПО в среде действующей системы;

– заменяемость — атрибут, который характеризует возможность переноса ПО с одной инструментальной среды в другую с необходимой инсталляцией или адаптацией ПО.

2 МОДЕЛИ КАЧЕСТВА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Модели качества ПО имеют следующие четыре уровня представления, рассмотрим мы все 4 уровня в модели ISO 9126

Модель МакКола

Первая модель качества была предложена МакКолом [4-6]. Предложенная модель была в основном предназначена для определения полной характеристики качества программного продукта через его различные характеристики. Модель качества МакКола (см. рис. 2) имеет три главных направления для определения и идентификации качества программного обеспечения:

– использование (корректность, надежность, эффективность, целостность, практичность);

– модификация (тестируемость, гибкость, сопровождаемость – факторы качества важные для разработки новой версии программного обеспечения);

– переносимость (мобильность, возможность многократного использования, функциональная совместимость – факторы качества важные для переносимости программного продукта на другие аппаратные программные платформы).

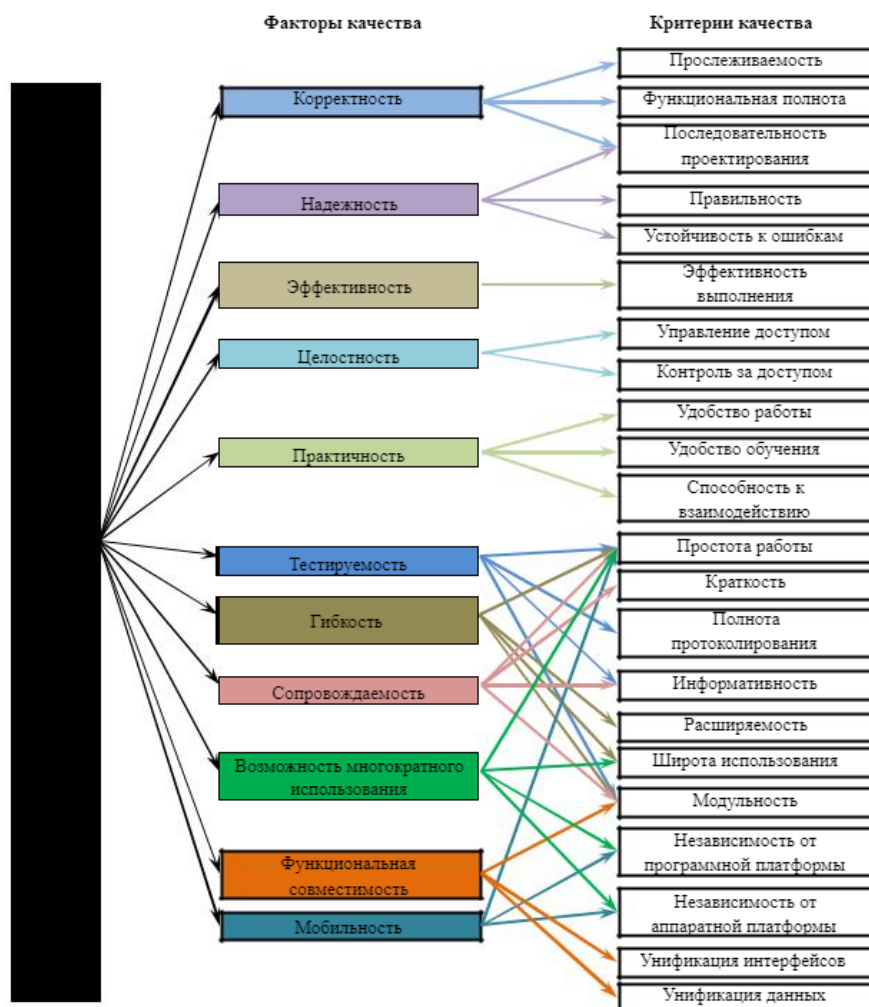


Рисунок 2 – Модель качества программного обеспечения МакКола

Модель Боэма

Второй из основополагающих моделей качества является модель качества Боэма. Модель Боэма имеет недостатки современных моделей, которые автоматически и качественно оценивают качество программного обеспечения. В сущности, модель Боэма пытается качественно определить качество программного обеспечения заданным набором показателей и метрик. Модель качества Боэма представляет характеристики программного обеспечения в более крупном масштабе, чем модель МакКола. Модель Боэма похожа на модель качества МакКола тем, что она также является иерархической моделью качества, структурированную вокруг высокоуровневых, промежуточных и примитивных характеристик, каждая из которых вносит свой вклад в уровень качества программного обеспечения.

В этой модели практичность описывает, как легко, надежно и эффективно программный продукт может быть использован, сопровождаемость характеризует насколько легко изменить и повторно протестировать программный продукт, и мобильность описывает, как

программный продукт может использоваться, даже при изменении программных и аппаратных средств.

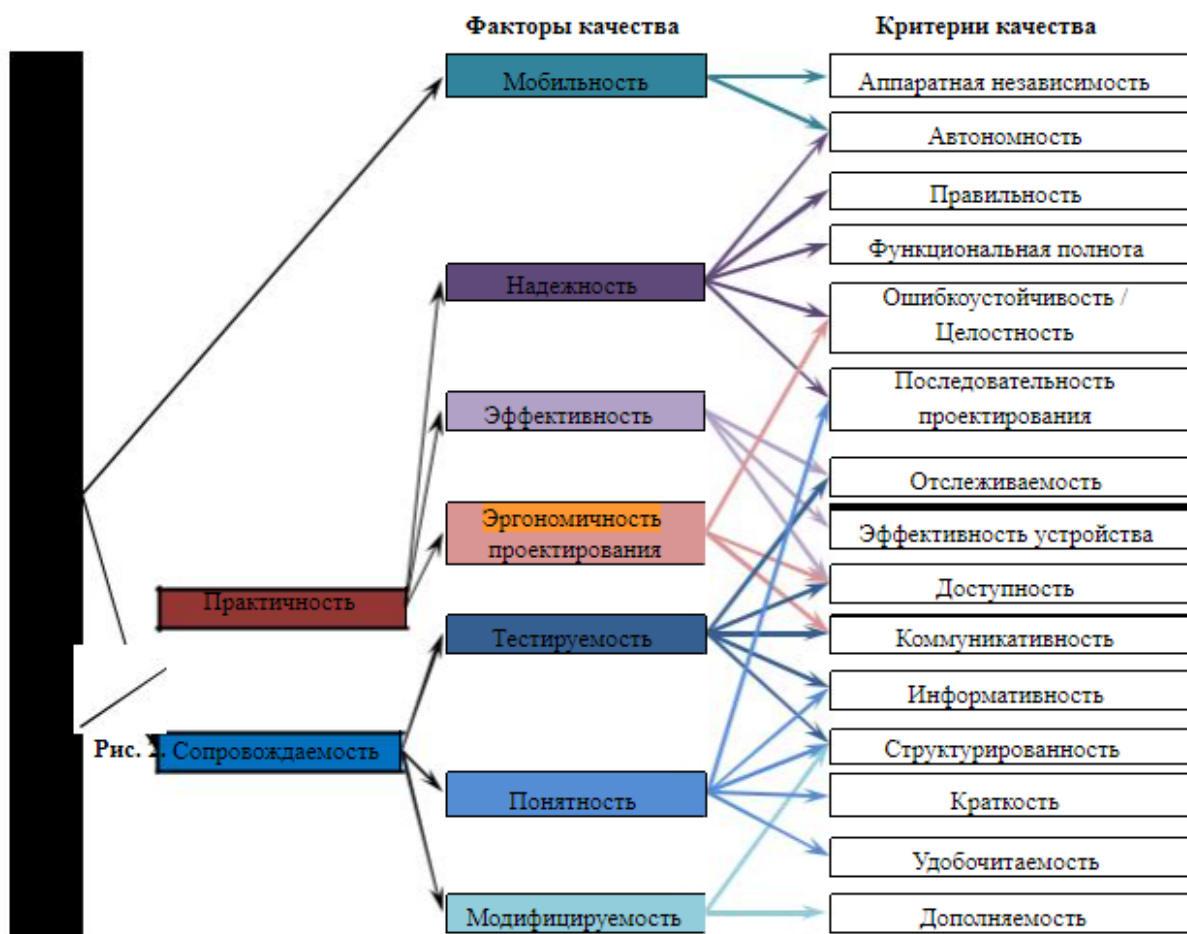


Рисунок 3 – Модель Боэма

Модель FURPS/FURPS+

Акроним FURPS, используемый в обозначении модели, обозначает следующие категории требований к качеству ПО:

Functionality (Функциональность) /особенности, возможности, безопасность/;

Usability (Практичность) /человеческий фактор, эргономичность, пользовательская документация/;

Reliability (Надежность) /частота отказов, восстановление информации, прогнозируемость/;

Performance (Производительность) /время отклика, пропускная способность, точность, доступность, использование ресурсов/;

Supportability (Эксплуатационная пригодность) /тестируемость, расширяемость, адаптируемость, сопровождаемость, совместимость, конфигурируемость, обслуживаемость, требования к установке, локализуемость/.

Символ «+» расширяет FURPS модель, добавляя к ней:

- ограничения проекта (ограничения по ресурсам, требования к языкам и средствам разработки, требования к аппаратному обеспечению);
- интерфейс (ограничения накладываемые на взаимодействие с внешними системами);
- требования к выполнению,
- физические требования,
- требования к лицензированию.

FURPS модель качества, предложенная Грейди и Hewlett Packard, построена схожим образом с моделями МакКола и Боэма, но в отличие от них состоит из двух слоев, первый определяет характеристики, а второй связанные с ними атрибуты. Основной концепцией, лежащей в основе FURPS модели качества, является декомпозиция характеристик программного обеспечения на две категории требований, а именно, функциональные (F) и нефункциональные (URPS) требования. Эти выделенные категории могут быть использованы как в качестве требований к программному продукту, так и в оценке качества ПП. В настоящее время модель FURPS+ широко используется в разработке программного обеспечения и при идентификации требований к разрабатываемой системе целесообразно использование FURPS+ модели в качестве универсального контрольного перечня характеристик ПО.

Модель Гецци

Карло Гецци и соавторы [9] различают качество продукта и процесса. Согласно модели Гецци к качеству программного обеспечения относят следующие характеристики программного обеспечения: целостность, надежность и устойчивость, производительность, практичность, верифицируемость, сопровождаемость, возможность многократного использования, мобильность, понятность, возможность взаимодействия, эффективность, своевременность реагирования, видимость процесса разработки.

Модель качества Дроми

Модель качества Дроми [10] основана на критериях оценки. Модель Дроми стремится оценить качество системы, в то время как каждый программный продукт, имеет качество отличное от других. Модель Дроми помогает в предсказании дефектов ПО и указывает на те свойства ПО, пренебрежение которыми может привести к появлению дефектов. Эта модель основывается на отношениях между характеристиками качества и подхарактеристиками, между свойствами программного обеспечения и характеристиками качества ПО. 6 Модель качества SATC

В Центре обеспечения качества программного обеспечения NASA (Software Assurance Technology Center, SATC) была разработана программа метрик [11], обеспечивающая оценку рисков проекта, качества продукции и эффективности процессов. Программа SATC рекомендует отдельно отслеживать качество требований, качество программного обеспечения и

других продуктов (документации), качество тестирования и качество выполнения процессов. Модель качества SATC определяет набор целей, связанных с программным продуктом и атрибуты процессов в соответствии со структурой модели качества программного обеспечения ISO 9126-1.

7 Модель качества ISO-9126

Качество ПО — это совокупность свойств, определяющих полезность изделия (программы) для пользователей в соответствии с функциональным назначением и предъявленными требованиями. При этом требования могут трактоваться довольно широко, что порождает целый ряд независимых определений понятия «качество». Чаще всего используется определение ISO 9001, согласно которому качество есть «степень соответствия присущих характеристик требованиям». Качество ПО — это относительное понятие, которое имеет смысл только при учете реальных условий применения ПО. Поэтому требования, предъявляемые к качеству ПО, ставятся в соответствии с условиями и конкретной областью применения ПО.

Для оценки качества программного обеспечения на основе теории нечетких множеств и метода анализа иерархий, Чанг и др. были определены руководящие принципы, и этот подход ими был применен к модели качества ISO 9126-1. Оценки качества программного обеспечения основаны на характеристиках и подхарактеристиках модели

Шармой А. и др. была предложена компонентно-ориентированная модель качества разработки программного обеспечения, которая включает все характеристики и подхарактеристики модели качества ISO 9126-1, а также предлагает новые подхарактеристики, такие как пригодность к повторному использованию, гибкость, сложность, прослеживаемость, масштабируемость. Метод анализа иерархий в этой модели используется для оценки качества проекта.

Таблица 1

Метрика качества CASE-средств

№ характеристики уровня				Характеристики (показатели) качества программного обеспечения	Коэф. 1 уровня	Коэф. 2 уровня	Коэф. 3 уровня	Коэф. 4 уровня
1	2	3	4	5	6	7	8	9
1				Функциональные возможности	0,5			
	1			Поддерживаемые методы (модели, нотации)		0,3		
		1		Структурные модели функций (процессов)			0,5	
			1	Функциональные (SADT, IDEF0)				0,3
			2	Бизнес-процессов (IDEF3, EPC)				0,3
			3	Потоков данных (DFD)				0,3
			4	Событийные (STD)				0,1
		2		Объектно-ориентированные модели (UML)			0,3	
		3		Иерархические модели			0,1	
			1	Organizational chart				0,5
			2	Objective Diagram				0,5
		4		Информационные модели (ERD)			0,1	
	2			Пригодность		0,2		
		1		Построение моделей			0,6	
			1	Возможности декомпозиции моделей				0,7
			2	Свойства объектов, определяемые пользователем (UDP)				0,2
			3	Наличие встроеной документации				0,1
		2		Экспорт отчетов (формат)			0,4	
			1	Формат простого текстового документа				0,2
			2	Форматы MS Office				0,2
			3	Формат RTF				0,2
			4	Формат HTML				0,2
			5	Формат XML				0,2
	3			Способность к взаимодействию		0,1		
		1		SAP/R3			0,1	
		2		MS Visio			0,1	
		3		ERwin			0,1	
		4		Requisite Pro			0,1	
		5		Performance Studio			0,05	
		6		ClearCase			0,05	
		7		Oracle SQL Developer			0,05	
		8		Lotus			0,05	
		9		Arena			0,1	
		10		Paradigm Plus			0,05	
		11		Rational Data Architect			0,05	
		12		Oracle Designer			0,1	
		13		PVCS			0,05	
		14		SoDA			0,05	
	4			Поддерживаемые процессы ЖЦ ПО		0,3		

Таблица 1 (окончание)

1	2	3	4	5	6	7	8	9
		1		Анализ и проектирование			0,5	
		2		Проектирование БД и файлов			0,1	
		3		Программирование			0,2	
		4		Сопровождение и реинжиниринг			0,2	
	5			Уровень интегрированности		0,1		
		1		Вспомогательные программы			0,4	
		2		Пакеты разработчика			0,8	
		3		Инструментальные средства			1	
2				Эффективность	0,2			
	1			Стоимость ПО		0,6		
		1		Низкая			1	
		2		Средняя			0,6	
		3		Высокая			0,4	
	2			Требования к операционной системе		0,3		
		1		Windows 7			0,3	
		2		Windows XP			0,1	
		3		Windows 2003 Server			0,1	
		4		Windows Vista			0,3	
		5		Linux/Unix			0,1	
		6		Windows NT			0,1	
3				Сопровождаемость	0,1			
	1			Изменяемость		0,6		
		1		Способ модификации отчетов			0,6	
			1	Настройка отчетов пользователем				0,5
			2	Визуальная настройка отчетов				1
		2		Сложность разработки нестандартных отчетов			0,4	
			1	Просто				1
			2	Сложно				0,6
	2			Совместимость версий		0,4		
4				Практичность	0,2			
	1			Простота использования		0,5		
		1		Простота работы			0,5	
			1	Низкая				0,5
			2	Средняя				0,7
			3	Высокая				1
		2		Возможность отмены/повтора изменений модели			0,2	
		3		Наличие русского интерфейса			0,3	
	2			Обучаемость		0,3		
		1		Учебный центр			0,3	
		2		Наличие документации			0,25	
		3		Документация на русском языке			0,45	
	3			Поддержка групповой работы		0,2		

ЗАКЛЮЧЕНИЕ

В заключении, качество программного продукта, нужно, чтобы ориентироваться в многообразии моделей и инструментальных средствах разработки, оценки, метриках и показателях удовлетворительности программного обеспечения. В данной статье мы рассмотрели, что такое качество ПО, какие критерии бывают, что такое модели качества по, на какие группы они делятся, так же на основе модели ISO-9126, увидели все 4 уровня определения качества программных продуктов, аналитики и последующего проектирования системы. Каждая модель несет в себе разные критерии, на котором основываются преимущества и недостатки выбранной модели, так же каждая модель отвечает сертификации и ГОСТу, какую модель мы бы не выбрали, главное проверять программный продукт на качество, тогда мы можем понять подходит ли он для эксплуатации и внедрения на пользовательский рынок или нет. Во всех моделях есть 3 главные составляющие информационной безопасности, а именно целостность, достоверность, конфиденциальность, все программные продукты должны иметь эти критерии.

Использованные источники:

1. Орлов С. А. Программная инженерия. Учебник для вузов. 5-е издание обновленное и дополненное. Стандарты третьего поколения. – СПб.: Питер, 2018. – 640 с.
2. Сергеев С. Ф., Падерно П. И., Назаренко Н. А. Введение в проектирование интеллектуальных интерфейсов. – СПб.: СПбГУ ИТМО, 2011 – 108 с.
3. Савенко И.И. Технология разработки программного обеспечения: конспект лекции. – Томск: Изд-во Томского политехнического университета, 2014 – 67 с.
4. Жизненный цикл программного обеспечения [Электронный ресурс]. URL: <https://qaevolution.ru/zhiznennyj-cikl-programmnogo-obespecheniya/> (дата обращения: 08.12.2021).
5. МОДЕЛИ РАЗРАБОТКИ И ТЕСТИРОВАНИЯ ПО: ИНКРЕМЕНТНАЯ МОДЕЛЬ [Электронный ресурс]. URL: <https://bytextest.ru/2017/11/23/incremental-model/> (дата обращения: 14.01.2021).