

DOI: 10.15514/ISPRAS-2021-33(1)-12



A Design Guide for Usable Web APIs

Путеводитель по проектированию удобных Web-API

M. Тельо-Родригес, ORCID: 0000-0002-7424-9912 <maritello797@gmail.com>
X.O. Очаран-Эрнандес, ORCID: 0000-0002-2598-1445 <jocharan@uv.mx>
X.K. Перес-Арриага, ORCID: 0000-0003-2354-2462 <juaperez@uv.mx>
К. Лимон, ORCID: 0000-0003-4654-636X <hlimon@uv.mx>
A.X. Санчес-Гарсия, ORCID: 0000-0002-2917-2960 <angesanchez@uv.mx>
 Университет Веракрузана
 Мексика, 91000, Веракрус, Халана

Аннотация. Направления развития облачных вычислений, такие как Software as a Service (SaaS), позволяют поставщикам размещать сложные приложения через Интернет, делая их доступными для внешних потребителей через интерфейсы прикладного программирования (API). Успех SaaS, как и в некотором смысле любой распределенной системы, во многом зависит от ее API. Наличие очень удобных в использовании API повышают эффективность и качество процесса разработки, хотя, конечно для программистов остаются существенными и другие аспекты API. Различные исследования показывают, что в процессе разработки API наиболее подходящим для решения проблем удобства использования является этап проектирования. При проектировании API удобство использования должно являться явным критерием качества. В настоящей статье мы предлагаем путеводитель по проектированию Web-API с акцентом на удобство использования, опираясь на лучшие методы проектирования удобных Web-API. Наш путеводитель по проектированию основано на адаптации методологии *проектного подхода к исследованиям* (Design Science Research Methodology, DSRM) и дополнено систематическим обзором литературы, а также анализом серой литературы по методам, методикам и инструментам, используемым для разработки удобных API.

Ключевые слова: интерфейс прикладного программирования; API; Web-API; удобство использования API; путеводитель по проектированию

Для цитирования: Тельо-Родригес М., Очаран-Эрнандес X.O., Перес-Арриага X.K., Лимон К., Санчес-Гарсия А.Х. Путеводитель по проектированию удобных Web-API. Труды ИСП РАН, том 33, вып. 1, 2021 г., стр. 173-188. DOI: 10.15514/ISPRAS-2021-33(1)-12

Благодарности. Авторы выражают признательность за поддержку и финансирование, предоставленные PRODEP 2018 (Programa para el Desarrollo Profesional Docente en Educación Superior, 48058 511-6/18-9245/PTC-888).

M. Tello-Rodríguez, ORCID: 0000-0002-7424-9912 <maritello797@gmail.com>
J.O. Ocharán-Hernández, ORCID: 0000-0002-2598-1445 <jocharan@uv.mx>
J.C. Pérez-Arriaga, ORCID: 0000-0003-2354-2462 <juaperez@uv.mx>
X. Limón, ORCID: 0000-0003-4654-636X <hlimon@uv.mx>
Á.J. Sánchez-García, ORCID: 0000-0002-2917-2960 <angesanchez@uv.mx>
 Universidad Veracruzana,
 Xalapa, Veracruz, México, 91000

Abstract. Cloud computing trends such as Software as a Service (SaaS) enable providers to host complex applications over the Internet, making them available to external consumers through an Application Programming Interface (API). The success of a SaaS, and in some sense any distributed system, is greatly influenced by its API. Highly usable APIs improve the efficiency of the development process and its quality, ensuring that programmers continue to appreciate other aspects of the API while increasing their productivity. Different studies state that the design phase within the development process of an API is the most appropriate to address usability issues. Therefore, usability should be considered as an explicit criterion in the design of an API. In this paper, we propose a design guide for web APIs with an emphasis on usability, using the best practices of usable web APIs design. Our design guide is based on an adaptation of the design science research methodology (DSRM), and it is complemented with a systematic literature review and gray literature analysis concerning methods, techniques, and tools used to develop usable APIs.

Keywords: Application Programming Interface; API; Web API; API Usability; Design Guide.

For citation: Tello-Rodríguez M., Ocharán-Hernández J.O., Pérez-Arriaga J.C., Limón X., Sánchez-García Á.J. A Design Guide for Usable Web APIs. Trudy ISP RAN/Proc. ISP RAS, vol. 33, issue 1, 2021, pp. 173-188 (in Russian). DOI: 10.15514/ISPRAS-2021-33(1)-12.

Acknowledgments. The authors are grateful for the support and funding granted by PRODEP 2018 (Programa para el Desarrollo Profesional Docente en Educación Superior, 48058 511-6/18-9245/PTC-888).

1. Введение

Web-API обеспечивают эффективный и расширяемый подход для взаимодействия приложений [1] в гетерогенных и распределенных средах, таких как облачные вычисления. В качестве интерфейсов API-интерфейсы позволяют пользователю (разработчику клиентской части приложения) воспринимать программы, контролировать их и взаимодействовать с ними [2]. Точно так же, как в любом пользовательском интерфейсе отражается опыт пользователей по взаимодействию с приложениями, в API отражается опыт разработчиков приложений. Удобство использования является абсолютно необходимым минимумом для того, чтобы разработчик использовал данный API [5].

В своей работе мы руководствовались определением удобства использования, представленным в стандарте ISO / IEC 25010 [6] (который заменил стандарт ISO / IEC 9126). Стандарт определяет удобство использования как подмножество модели качества в использовании (quality-in-use). Эта модель определяется пятью характеристиками: результативность, производительность, удовлетворенность, покрытие контекста и свобода от риска. Удобство использования является подмножеством этих характеристик и включает следующие три характеристики:

- 1) *результативность* отражает возможность для конкретных пользователей использовать систему в определенном контексте и достигать своих целей полностью и точно.
- 2) *производительность* оценивает, как конкретный пользователь использует ресурсы для эффективного достижения своих целей.
- 3) *удовлетворенность* отражает уровень удовлетворенности пользователей использованием системы в определенном контексте; эта вспомогательная характеристика далее делится на четыре атрибута: *полезность* (когнитивное

удовлетворение), *доверие* (уверенность в поведении системы), *удовольствие* (эмоциональное удовлетворение) и *комфорт* (физическое удовлетворение).

В ISO / IEC 25010 определяется вторая модель качества, в которой удобство использования классифицируется как одна из восьми характеристик качества, и определяется как «степень, в которой программный продукт может успешно пониматься, изучаться, применяться пользователем и являться для него наиболее привлекательным при использовании в соответствии с требованиями». Внутренние и внешние свойства удобства использования моделируются с помощью следующих шести подхарактеристик [7]:

- 1) определимость пригодности;
- 2) изучаемость;
- 3) управляемость;
- 4) защищенность от ошибок пользователя;
- 5) эстетика пользовательского интерфейса;
- 6) доступность.

Биль (Matthias Biehl) в [3] отмечает, что удобство использования является желательным свойством любого API, включающим в себя такие характеристики, как ориентация на потребителя, простота, понятность без дополнительных пояснений, интуитивность и предсказуемость. Сторонники удобства использования API, такие как Джошуа Блох (Joshua Bloch) из Google, подчеркивают, что «хорошие API повышают удовольствие от работы и продуктивность разработчиков [...], а также качество программного обеспечения, создаваемое с их использованием» [9].

По мнению Хеннинга (Michi Henning) [10], последствиями применения плохо спроектированных API являются снижение надежности программного обеспечения и продуктивности программиста, использующего API. Проблемы с удобством использования могут вызвать ненужную сложность в коде клиентской части приложений и, следовательно, во всей распределенной системе.

Как следует из литературы, во всем процессе разработки наиболее подходящим для решения проблемы удобства использования является этап проектирования API [12, 13]. Кроме того, удобство использования должно быть добавлено как явный критерий проектирования и оценки, чтобы исключить в ходе разработки возможность случайного создания API, неудобного для использования [11].

Нашей целью является разработка высокоуровневого руководства по проектированию Web-API на основе мирового опыта, артефактов проектирования и методов повышения удобства использования Web-API. Мы предлагаем не строгую пошаговую процедуру, а скорее набор общих рекомендаций, которые разработчики API могут применять независимо от того, какую методологию или процедуру разработки.

В своей работе мы используем адаптацию методологии проектного подхода к исследованиям (Design Science Research Methodology, DSRM) [14], которая включает пять этапов. На сегодняшний день мы сделали следующее: путем систематического обзора литературы (systematic literature review, SLR) и анализа серой литературы определили методы, приемы и инструменты, используемые для разработки удобных для использования API; сравнили и смоделировали процессы разработки API, а также выбрали и интегрировали эти элементы для разработки первой версии путеводителя по проектированию. В будущем мы намерены завершить демонстрацию и оценку руководства, выполнив специальное исследование, в котором с использованием предлагаемого руководства будет спроектирован реальный Web-API, и удобство его использования будет оцениваться с помощью различных методов. С отчетом по проведенным нами SLR и обзору серой литературы можно ознакомиться в [15], настоящая статья является продолжением указанной работы.

Статья организована следующим образом. В разд. 2 приводится обзор некоторых процессов разработки и проектирования, ориентированных на удобство использования и восприятие

пользователями (User Experience, UX). В разд. 3 представлен метод разработки нашего руководства по проектированию API, ориентированного на удобство использования, на основе методологии, предложенной Пейфферсом (Ken Peffers) и др. [14]. Разд. 4 описывает предлагаемое руководство по проектированию и следующую из него модель. Наконец, в разд. приводятся заключение и описываются направления будущей работы.

2. Родственные работы

Несколько авторов представили исследования, которые затрагивают вопросы обеспечения удобства использования API во время разработки, но мало кто пытался это сделать на этапе проектирования. В проанализированной литературе мы нашли методы, процессы, руководства, описание опыта, эвристики, артефакты, методы и инструменты, в которых прямо или косвенно учитывается удобство использования. Ниже в этом разделе некоторые из этих работ детализированы, начиная от общих процессов разработки и заканчивая конкретными артефактами и методами, используемыми в различных проектах.

Авторы работы [5] описывают ориентированный на пользователя процесс, в котором учитываются удобство использования и другие аспекты, такие как безопасность и масштабируемость. Биль [3] описывает полный процесс разработки, но без упоминания практических аспектов удобства использования. С другой стороны, Хантер [16] описывает дизайн-ориентированную методологию, с упором на учет опыта разработчиков и RESTful API – Web-службы на основе архитектуры REST [17].

Кроме того, мы нашли руководства, в которых описываются процессы разработки и проектирования, специально предназначенные для повышения удобства использования API. В 2008 году Стилос (Jeffrey Stylos) и др. в [18] описали свою работу по редизайну, ориентированному на пользователя, для повышения удобства использования существующего API. В 2014 году Ли (Sunghoon Lee) и др. [19] предложили процесс проектирования в расчете на обеспечение удобства использования вместе с рекомендациями по оценке API, ссылаясь на критерии удобства использования, «простые в применении и количественно измеримые». Еще одним примером является проект API Craft на сайте Google Groups, в котором успех разработчика ставится выше любого другого принципа проектирования [4].

В отличие от упомянутых процессов разработки и проектирования, некоторые авторы ставят на передний план артефакты дизайна, практические приемы и методы, нацеленные на повышение удобства использования. Прототипирование – это один из наиболее рекомендуемых методов, поскольку прототипы могут использоваться для привлечения пользователей (программистов клиентских частей приложений) и получения обратной связи на ранних этапах, что помогает улучшить пользовательское восприятие [13, 18, 20].

Часто упоминаются методы проверки и оценки удобства использования. В [16] приемочные тесты, основанные на описании требований заказчиков, предлагаются для проверки того, что тестируемый вариант использования является простым, понятным и функциональным. Другим примером является «эвристическая оценка» Нильсена для проверки дизайна API. Хотя рекомендации Нильсона можно использовать как отдельное руководство по проектированию API, разработчикам API следует иметь их в виду, даже если они используют какой-либо другой подход [11].

Робиллард (Martin P. Robillard) и Делайн (Robert DeLine) [21] отмечают, что документирование API не может быть четко отделено от обязательств по проектированию API, даже если для этого требуются другие навыки. Документация часто используется, чтобы помочь заполнить разрыв между дизайном API и пониманием разработчика программного обеспечения того, как успешно использовать API [22]. Мы нашли несколько работ, посвященных документации API и удобству использования. Например, имеется исследование, авторы которого убеждают в том, что, следуя их рекомендациям по поводу

документирования REST API, разработчики API могут уменьшить количество ошибок, повысить уровень успеха и удовлетворенность разработчиков, использующих API [23].

Таким образом, мы смогли найти только исследования, в которых предлагаются руководства по проектированию и соответствующие процессы для повышения удобства использования API-интерфейсов, но не с упором на Web-API. Руководства по проектированию, которые включают Web-API, не предназначены для решения проблем удобства использования, но охватывают другие аспекты проектирования, например, документирование API. В отличие от этого, в настоящей работе представлено руководство по проектированию удобных Web-API, в котором собраны предложения авторов по улучшению удобства использования. Нашей целью является обеспечение полного набора, методов и артефактов проектирования, чтобы каждый разработчик API мог применить то, что лучше всего подходит для его проекта.

3. Метод исследования

Применяемый нами метод исследования был вдохновлен методологией проектного подхода к исследованиям Пеффера и др. [14]. Авторы [14] утверждают, что методология работает как проводник для исследований в области науки проектирования (Design Science, DS), помогая в признании и легитимизации целей исследования, процессов и результатов, помогая исследователям представить свою работу. На рис. 1 показаны пять последовательных фаз используемого метода, фазы нашего исследования подробно описаны в последующих параграфах.

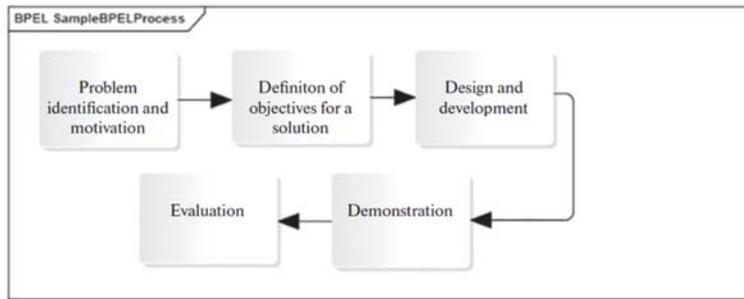


Рис. 1. Этапы методологии, описанной Пеффером и др.
Fig. 1. The phases in the methodology described by Peffers et al.

3.1 Выявление и мотивация проблемы

С целью выявления проблемы был выполнен обзор литературы с ориентацией на определение Web-API, удобство использования как атрибут качества, важность удобства использования API и отрицательное влияние неудобных API. Цель состояла в том, чтобы обосновать полезность рассмотрения удобства использования как значимого атрибута при проектировании Web-API.

Согласно информации, полученной при работе над обзором, проблему можно резюмировать следующим образом: на удобство использования API может влиять несколько факторов, таких как учебные ресурсы, контекст использования, технические ограничения и проектные решения на низком и высоком уровнях абстракции [13, 24, 25]. Пренебрежение удобством API в процессе не позволяет избежать аномалий, которые могут привести к снижению показателей надежности и соответствия требованиям [26]. Человеческий фактор имеет отношение и к разработчикам API, а отсутствие опыта разработки API может негативно влиять на конечный продукт. Кроме того, не хватает методологий, нацеленных на удобство использования, учитывая, что литература по этому вопросу технически неоднородна, и

многие авторы вообще не затрагивают или обсуждают только поверхностно некоторые аспекты удобства использования [20, 26].

3.2 Определение целей решения

Чтобы определить инструменты, методы и приемы, используемые для повышения удобства использования API на любом этапе разработки, мы выполнили систематический обзор литературы (SLR). Отчет об этой работе [15] (на испанском языке) был подготовлен на основе руководства, предложенного Барбарой Китченхэм (Barbara Kitchenham) и др. в [27], в котором описывается процесс систематических обзоров в области инженерии программного обеспечения..

Для достижения цели SRL мы сформулировали четыре исследовательских вопроса (Research Questions, RQ), которые перечислены в табл. 1. Эти RQ легли в основу создания поискового запроса, которая использовалась при поиске статей в семи базах данных: Scopus, Web of Science, Science Direct, Wiley, SpringerLink, IEEE Xplore Digital Library и ACM Digital Library.

Табл.1. Исследовательские вопросы для SRL

Table 1. Research questions for the SRL

№	Вопрос
RQ 1	Какие методы, практики, эвристики, техники, методологии или процессы, найденные в литературе, посвящены разработке удобных в использовании API?
RQ 2	Какие существуют инструменты, поддерживающие практики, техники, методы, методологии и процессы?
RQ 3	Каковы характеристики удобных в использовании API?
RQ 4	Какие существуют доказательства эффективности этих методов, техник, практик, методологий, процессов и инструментов?

По поисковому запросу было найдено 19112 статей. Мы применили к полученным статьям шесть критериев включения и исключения, в результате чего у нас осталось 65, а после удаления дубликатов мы сократили список до 42 статей. Для ознакомления с этим списком, а также с более подробным описанием процесса SLR мы адресуем читателя к нашей предыдущей публикации [15]. Из найденных работ были извлечены следующие данные:

- название;
- автор;
- год публикации;
- источник;
- ключевые слова;
- ответы на RQ;
- определяемые метод, техника, эвристика, практика и соответствующее описание;
- этапы и шаги метода, техники, эвристики или практики;
- желательные свойства в используемых API;
- характеристики неудобных API;
- рассмотренные аспекты удобства использования;
- конкретные примеры метода, техники, эвристики или практики;
- отношение инструмента к методу, технике или практике;
- проверяется или измеряется ли эффективность метода, техники, эвристики или практики, и каким образом?

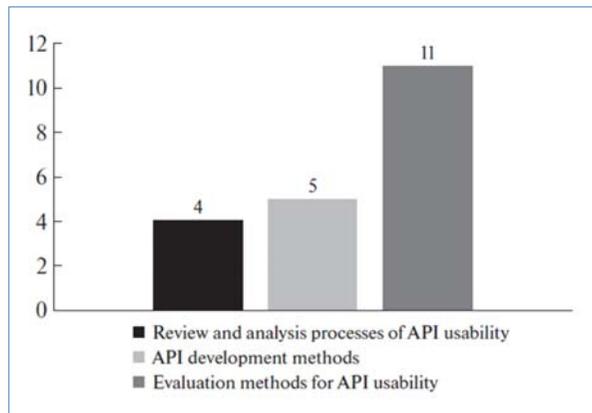


Рис. 2. Статьи, посвященные различным вопросам удобства использования API

Fig. 4. Papers addressing different API usability topics

На рис. 2 мы можем увидеть, что количество найденных методов оценки удобства использования API превышает количество найденных процессов обзора и анализа, а также методов разработки.

После извлечения данных был произведен синтез данных, следуя методу метаагрегации [29], который приводит к извлечению найденных результатов и их классификации по категориям. Всего извлечено 158 результатов. В итоге мы получили следующие четырнадцать категорий:

- определение удобства использования;
- факторы, влияющие на удобства использования;
- аспекты удобства использования;
- оценка и измерение удобства использования;
- проблемы удобства использования;
- удобные в использовании API:
 - важность и преимущества;
 - обычная практика;
 - характеристики и свойства;
 - методы разработки;
 - инструменты поддержки;
 - документация:
 - характеристики;
- неудобные API:
 - характеристики и свойства;
 - последствия.

Чтобы дополнить полученную информацию, был проведен обзор серой литературы с учетом книг и онлайн-ресурсов, ссылки на которые можно найти в [15]. Процесс обзора заключался в быстром чтении тематического указателя, аннотации и содержания каждого ресурса, чтобы определить, будет ли это интересно.

Учитывалось, упоминалось ли явно удобство использования или был ли описанный процесс разработки ориентирован на восприятие пользователей (User Experience, UX). После отбора ресурсов были прочитаны разделы, соответствующие нашему исследованию, и были извлечены существенные данные о самом процессе разработки и его артефактах, этапах, методах.

SLR помог также выявить различные методы оценки удобства использования, которые обычно применяются после некоторого этапа цикла разработки для получения обратной

связи по поводу уровне удобства использования API; например, метод пошагового разбора (API Walkthrough method) [30] или групповой экспертной оценки (Peer review API) [24, 31]. Оба указанных метода используются для обнаружения недостатков удобства использования при проектировании API. Мы также заметили, что в литературе почти не упоминаются инструменты, поддерживающие разработку удобных в использовании API.

3.3 Проектирование и разработка

Для проектирования и разработки нашего путеводителя была принята во внимание информация, полученная в упомянутых выше обзорах литературы. Из таблиц извлеченных данных (одна с данными SRL, другая с данными из серой литературы) наиболее актуальная и полезная информация была собрана в таблицу, в которой мы детализировали выявленные ранее методы, инструменты и артефакты.

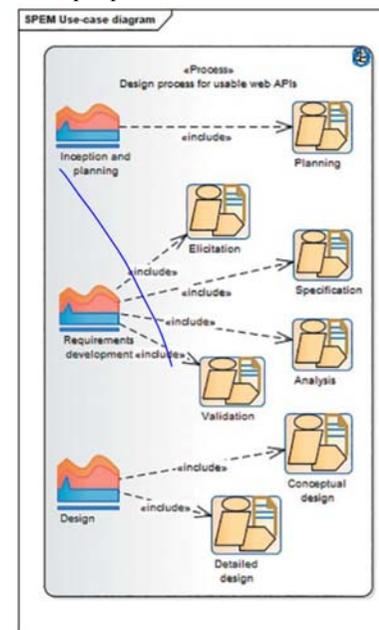


Рис. 3. Этапы создания и планирования, разработки требований и проектирования, а также соответствующие области деятельности

Fig 3. Phases of inception and planning, requirements development and design, and its corresponding areas of activity

3.4 Оценка и демонстрация

Для оценки и демонстрации мы планируем провести специальное исследование, которое будет включать разработку Web-API в соответствии с процессом, основанным на предлагаемом руководстве. Затем удобство использования разработанного API будет оцениваться с помощью различных методов анализа и оценки удобства использования, ранее выявленных в SRL. Можно использовать любой из следующих методов, все они применимы к проектированию API: метод пошагового разбора API [30], Apiness API design review [24], Think Aloud Evaluation [34], Cognitive Dimensions Framework [13, 26, 35, 36], групповые

экспертные оценки [31], метод Concept Maps [9] и исследовательский опросник (Research Questionnaire) [37].

4. Предлагаемое руководство по проектированию

В руководстве основное внимание уделяется представлению артефактов, методов, задач, инструментов и передовых практик, используемых в дизайне веб-API для повышения удобства использования конечного API, но также рассматриваются этапы начала/создания и планирования, а также разработки требований. Дизайнеры веб-API могут выбрать использование артефактов, методов, задач, инструментов и практик, которые они считают подходящими для своего проекта, поскольку не все из них могут быть применимы. Руководство призвано предоставить предложения, но не является методом разработки, поэтому мы не рекомендуем использовать его как таковой.

Для моделирования процессов, рассматриваемых в руководстве, была использована SPEM 2.0 (Software & Systems Process Engineering Metamodel, метамодель разработки программного обеспечения и систем), модель для определения конкретных программных процессов. На рис. 3 можно увидеть схему рассматриваемых этапов данного руководства и соответствующие области деятельности. К этим этапам мы обратимся отдельно в последующих подразделах.

4.1 Этап создания и планирования

Выявленные нами ориентированные на пользователя процессы с самого начала процесса разработки учитывают интересы клиентского программиста и отдают приоритет его нуждам, что помогает разработчикам API принимать решения, которые в конечном итоге приносят пользу тому же клиентскому программисту [5]. Поэтому в области планирования деятельности мы предлагаем принять на вооружение задачи и артефакты, перечисленные в табл. 2.

Табл. 2. Задачи и артефакты, предлагаемые на этапе создания и планирования
Table 2. Tasks and artifacts proposed within the phase of inception and planning

Название	Тип	Ссылка
Определение проблемы и степени воздействия	Задача	[38]
План	Артефакт	[38]
Функциональная спецификация	Артефакт	[16]
Определение бизнес-ценности	Задача	[16]
Установка метрик	Задача	[16]

Мы заметили, что некоторые из предложенных задач и артефактов тесно связаны друг с другом. Например, задача определения проблемы и степени ее воздействия может быть учтена в плане Web-API, и то же самое может произойти с оценкой бизнес-ценности.

Кроме того, Хантер [16] предлагает использовать метрики, которые могут быть связаны с удобством использования, такие как коэффициент внедрения. Этот же автор рассматривает функциональную спецификацию в рамках управления проектом и планирования, а не разработку требований, как мы обнаружили у большинства авторов.

4.2 Этап разработки требований

Фаза разработки требований включает в себя такие области деятельности, как выявление, анализ, спецификация и валидация. Из рис. 4 мы видим, насколько итеративным является этот процесс [39], и приходим к пониманию того, что предлагаемые задачи, артефакты и методы в этих областях деятельности можно использовать гибким образом.

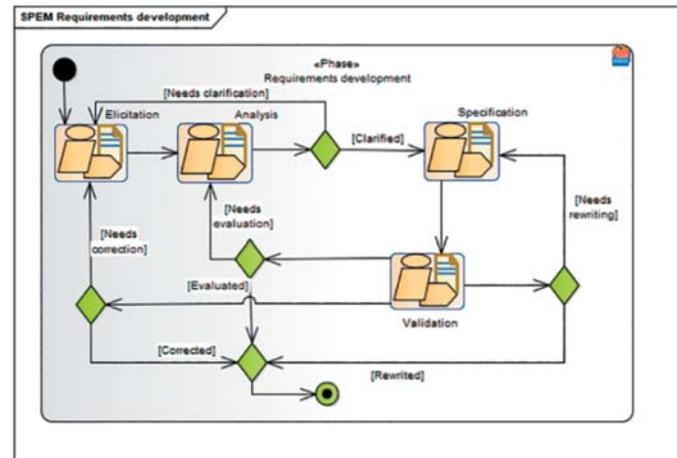


Рис. 4. Этап разработки требований и соответствующие области деятельности
Fig. 4. Phase of requirements development and its corresponding areas of activity

Несколько примеров артефактов и методов приведены в табл. 3, включая область деятельности для их предлагаемого использования. Картина целей – это артефакт, который может быть использован в двух областях деятельности – анализе и валидации; как и определение профилей программистов, картина целей может быть использована для извлечения и анализа информации. Методы создания сценариев использования и описания требований пользователей предложены несколькими авторами, что является логичным, поскольку эти методы помогают выявить потребности клиентских программистов, использующих API, и осуществлять проектирование с учетом их нужд [3, 13]. Предлагаемые методы сбора информации с помощью интервью и опросников помогают определить, что пользователи могут делать, как они это сделают, входные и выходные данные и другую важную информацию. Методы валидации особенно полезны для достижения работоспособности, подхарактеристики удобства использования в соответствии с ISO/IEC 25010, поскольку работоспособность включает в себя удовлетворение требований пользователя, учитывая все возможные сценарии [40]. Таким образом, методы и артефакты помогают лучше понять точку зрения программиста, использующего API, что является решающим для обеспечения удобства использования [41].

Табл. 3. Методы и артефакты, предлагаемые в рамках разработки требований
Table 3 Techniques and artifacts proposed within the phase of requirements development

Название	Тип	Область деятельности	Ссылки
Требуемые документы	Артефакт	Спецификация	[19]
Высокоуровневые сценарии использования	Метод	Анализ	[5, 20]
Интервью с заинтересованными сторонами	Метод	Сбор информации	[18, 20]
Пользовательские истории	Метод	Анализ	[3, 5, 13]
Картина целей	Артефакт	Анализ, валидация	[41]
Опросник	Метод	Сбор информации	[41]
Функциональная спецификация	Artifact	Спецификация	[16]

4.3 Этап проектирования

На этом этапе мы можем выявить места использования большинства найденных артефактов, методов, инструментов и даже руководств. На рис. 5 можно видеть области деятельности в рамках проектирования API, как это предлагается в литературе: все начинается с концептуального проекта, за которым следует его валидация; затем подготавливается детальный проект, после чего производится еще одна валидация. Несмотря на то, что поток работ между областями деятельности может казаться в некотором роде последовательным, этап проектирования на практике может быть очень гибким, и разработчики API могут вернуться к концептуальному проекту после выполнения деятельности по детальному проектированию.

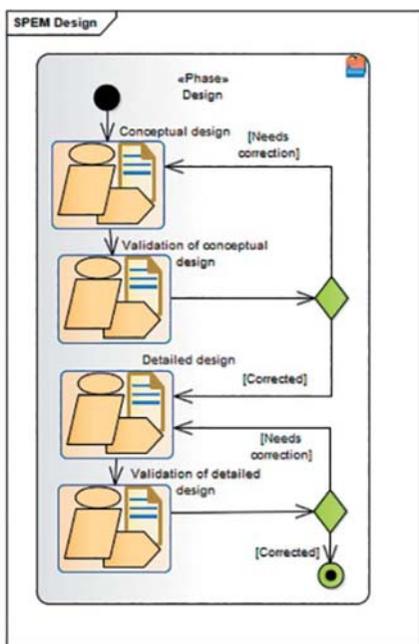


Рис. 5. Этап проектирования и соответствующие области деятельности
Fig 5. The phase of design and its corresponding areas of activity

В табл. 4 мы демонстрируем несколько техник, артефактов, рекомендаций и инструментов, используемых при проектировании Web-API для содействия в улучшении удобства использования результирующего API. Мы обнаружили, что инструмент для создания эскизов при концептуальном проектировании предложен только в одной работе [32], в то время как методы прототипирования предлагаются многими авторами и по-разному. Например, Рамакришнан (Lavanya Ramakrishnan) и др. [20] предлагают использовать нефункциональный прототип для получения обратной связи от пользователей. Джин (Brenda Jin), Сахни (Saurabh Sahni) и Шеват (Amir Shevat) [5] напротив, предлагают использовать реалистичные прототипы как подготовку к реализации, что связано с техникой валидации на основе приемочного тестирования: предполагается, что созданные прототипы будут тестироваться потребителями API. Другой метод валидации – это использование оценок удобства использования, таких как метод Think Aloud Evaluation, цель которого состоит в нахождении областей для улучшения проекта [18], или Cognitive Dimensions Framework, где несколько измерений могут использоваться для оценки различных проблем удобства

использования [13]. В рекомендациях Нильсона по «эвристической оценке» описываются десять свойств, которые разработчик может использовать для проверки своего проекта API, включая предотвращение ошибок, эстетику и эффективность использования. Заметим, что эти свойства являются подхарактеристиками удобства использования в соответствии с ISO/IEC 25010 [11].

Табл. 4. Методы, артефакты, рекомендации и инструменты, предлагаемые на этапе проектирования
Table 4. Techniques, artifacts, guidelines, and tools proposed within the phase of design

Название	Тип	Область деятельности	Ссылки
Эскиз	Метод, инструмент	Концептуальный дизайн	[32]
Прототипирование	Метод	Концептуальный дизайн	[3, 5, 18, 20]
Оценка удобства использования методом «think aloud»	Метод	Валидация	[18]
Руководства на основе критерия удобства использования	Руководства	Концептуальный и детальный дизайн	[19]
Оценка удобства использования с помощью CDF	Метод	Валидация	[13]
Таблица ресурсов	Артефакт	Детальный дизайн	[41]
Блок-схема для проверки данных параметров	Артефакт	Детальный дизайн	[41]
Описание API	Артефакт, руководство	Концептуальный и детальный дизайн	[3, 5, 16, 20, 41, 42]
Тест приемки	Метод	Валидация	[16]
Рейтинги Нильсона по степени удобства использования	Метод	Валидация	[38]
Рекомендации по «эвристической оценке» Нильсона	Метод	Валидация	[11]

Один из наиболее упоминаемых артефактов – это описания API, использование которых полезно для общения с потребителями API и получения от них отзывов. Кроме того, этот артефакт служит для валидации, поскольку описание может использоваться при моделировании, чтобы можно было ответить на некоторые вопросы, связанные с удобством использования, такие как «является ли API по-прежнему простым в использовании?» или «это все еще небольшой, гибкий и удобный API, или же мы создали чудовищный API?» [42]. Мы ожидаем, что предлагаемое руководство может служить набором методов, задач, инструментов и артефактов, чтобы разработчики Web-API могли использовать некоторые из этих элементов в процессе своих собственных разработок для создания удобных в использовании веб-API.

5. Выводы и дальнейшая работа

В настоящей работе представлены результаты проекта, направленного на создание руководства по разработке Web-API с акцентом на удобство использования. Проведен систематический обзор литературы с целью выявления методов, практик, серий руководств,

артефактов, техник и инструментов, с ориентацией на разработку удобных в использовании API, а также обзор серой литературы в дополнение к результатам SLR. Из всех обнаруженных методов и процессов выбраны элементы, которые включены в первый вариант руководства. Документация не была рассмотрена в явной форме, учитывая, что наш подход был сосредоточен на артефактах для разработки самого API, а не сопроводительной документации, но важно признать документацию решающим фактором, влияющим на удобство использования API.

В будущем предполагается продолжить работу над этапами демонстрации и оценки метода, описанного в разд. 4. На демонстрационном этапе будет проведено тематическое исследование, которое заключается в применении предлагаемого руководства при создании, планировании, разработке требований и проектировании проекта Web-API. На заключительном этапе с использованием методов оценки удобства использования мы оценим, как использование руководства в процессе проектирования Web-API влияет на удобство использования разработанного API. При этом предполагается приблизиться к ответам на такие вопросы, как «Улучшит ли сочетание каких-либо артефактов удобство использования разрабатываемого API», и, применяя руководство по проектированию к тематическому исследованию реальных примеров, проверить, может ли это общее руководство быть полезным для конкретных ситуаций. Наконец, цель оценки предлагаемого руководства заключается в том, чтобы узнать о его эффективности и определить возможности по совершенствованию для последующих проектов.

Список литературы / References

- [1] Espinha T., Zaidman A., Gross H.G. Web API growing pains: Loosely coupled yet strongly tied. *Journal of Systems and Software*, vol. 100, 2015, pp. 27-43.
- [2] Зосимов В.В., Христордов А.В., Булгакова А.С. Программные решения для динамического изменения пользовательского интерфейса на основе автоматически собранной информации о пользователе. Труды ИСП РАН, том 30, вып. 3, 2018 г., стр. 207-220. DOI:10.15514/ISPRAS-2018-30(3)-1 / Zosimov V.V., Khrystodorov O.V., Bulgakova O.S. Dynamically Changing User Interfaces: Software Solutions Based on Automatically Collected User Information. *Programming and Computer Software*, vol. 44, no. 6, 2018, pp. 492-498 (2018).
- [3] Biehl M. API Architecture: The Big Picture for Building APIs. CreateSpace, 2015, 107 p.
- [4] Mulloy B. Web API Design – Crafting Interfaces that Developers Love. Apigee, 2012, 36 p.
- [5] Jin B., Sahni S., Shevat A. Designing Web APIs: Building APIs That Developers Love. O'Reilly Media, 2018, 232 p.
- [6] ISO/IEC 25010:2011(en). Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models. 2011.
- [7] Speicher M. What is Usability? A Characterization based on ISO 9241-11 and ISO/IEC 25010. arXiv:1502.06792, 2015, 10 p.
- [8] Piccioni M., Furia C.A., Meyer B. An empirical study of API usability. In Proc. of the ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, 2013, pp. 5-14.
- [9] Gerken J., Jetter H.C., Zöllner M., Mader M., Reiterer H. The concept maps method as a tool to evaluate the usability of APIs. In Proc. of the SIGCHI Conference on Human Factors in Computing Systems, 2011, pp. 3373-3382.
- [10] Henning M. API design matters. *Communications of the ACM*, vol. 52, no. 5, 2009, pp. 46–56.
- [11] Myers B.A., Stylos J. Improving API usability. *Communications of the ACM*, vol. 59, no. 6, 2016, pp. 62-69.
- [12] Zibran M.F., Eishita F.Z., Roy C.K. Useful, but usable? Factors affecting the usability of APIs. In Proc. of the 18th Working Conference on Reverse Engineering, 2011, pp. 151–155.
- [13] Bhaskar R.K., Anslow C., Brosz J., Maurer F. Developing usable APIs with XP and cognitive dimensions. In Proc. of the IEEE Symposium on Visual Languages and Human-Centric Computing, 2016, pp. 101–105.
- [14] Peffers K., Tuunanen T., Rothenberger M.A., Chatterjee S. A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems*, vol. 24, issue 3, 2007, pp. 45-77.

- [15] Tello-Rodríguez M., Ocharán-Hernández J.O., Pérez-Arriaga J.C., Estadística F. De, Veracruzana U. Hacia una guía de diseño para APIs Web con un enfoque en la usabilidad. *Abstraction & Application*, vol. 25, 2019, pp. 36-60 (in Spanish).
- [16] Hunter K.L. Irresistible APIs: Designing web APIs that developers will love. Manning Publications, 2016, 232 p.
- [17] Eassa A.M., Elhoseny M., El-Bakry H.M., Salama A.S. NoSQL Injection Attack Detection in Web Applications Using RESTful Service. *Programming and Computer Software*, vol. 44, vol. 6, 2018, pp. 435-444.
- [18] Stylos J., Graf B. et al. A case study of API redesign for improved usability In Proc. of the IEEE Symposium on Visual Languages and Human-Centric Computing, 2008, pp. 189-192.
- [19] Lee S., Lee S., Lim S. et al. An API design process in terms of usability: A case study on building more usable apis for smart TV platform. In Proc. of the IEEE 38th International Computer Software and Applications Conference Workshops, 2014, pp.567-571.
- [20] Ramakrishnan L., Poon S., Hendrix V. et al. Experiences with user-centered design for the tigers workflow API. In Proc. of the IEEE 10th International Conference on e-Science, 2014, pp. 290-297.
- [21] Robillard M.P., Deline R. A field study of API learning obstacles. *Empirical Software Engineering*, vol. 16, 2011, pp.703-732.
- [22] Watson R. Applying the Cognitive Dimensions of API Usability to Improve API Documentation Planning. In Proc. of the 32nd ACM International Conference on The Design of Communication, 2014, pp. 1-2.
- [23] Sohan S.M., Maurer F., Anslow C., Robillard M.P. A study of the effectiveness of usage examples in REST API documentation. In Proc. of the IEEE Symposium on Visual Languages and Human-Centric Computing, 2017, pp. 53-61.
- [24] Macvean A., Maly M., Daughtry J. API Design Reviews at Scale. In Proc. of the Conference Extended Abstracts on Human Factors in Computing Systems, 2016, pp. 849-858.
- [25] Robillard M.P. What Makes APIs Hard to Learn? Answers from Developers. *IEEE Software*, vol. 26, no. 6, 2009, pp. 27-34,
- [26] Munir M.B., Mushtaq A. A framework for extending usability engineering: API usability essentials: Extending usability via component-based platform. In Proc. of the IEEE Conference on Open Systems, 2012, pp. 1-6.
- [27] Kitchenham B., Pretorius R., Budgen D. et al. Systematic literature reviews in software engineering-A tertiary study *Information and Software Technology*, volume 52, issue 8, 2010, pp. 792-805.
- [28] Stylos J., Myers B.A. The implications of method placement on API learnability. In Proc. of the 16th ACM SIGSOFT International Symposium on Foundations of software engineering, 2008, pp. 105-112.
- [29] Pearson A. Balancing the evidence: incorporating the synthesis of qualitative data into systematic reviews. *JBI Reports*, vol. 2, issue 2, 2004, pp. 45–64.
- [30] O'Callaghan P., O'Callaghan P. The API Walkthrough Method: A Lightweight Method for Getting Early Feedback about an API. In Proc. of the Workshop on Evaluation and Usability of Programming, Languages and Tools, 2010, pp. 1-6.
- [31] Farooq U., Welicki L., Zirkler D. API Peer Reviews: A Method for Evaluating Usability of Application Programming Interfaces. In Proc. of the ACM Conference on Computer Supported Cooperative Work, 2010, pp. 207–210.
- [32] Mitra R. Rapido : A Sketching Tool for Web API Designers. In Proc. of the 24th International Conference on World Wide Web, 2015, pp. 1509–1514.
- [33] Murphy-Hill E., Sadowski C., Head A. et al. Discovering API Usability Problems at Scale. In Proc. of the 2nd International Workshop on API Usage and Evolution, 2018, pp. 14-17.
- [34] Beaton J.K.K., Myers B.A.A., Stylos J. et al. Usability Evaluation for Enterprise SOA APIs. In Proc. of the 2nd International Workshop on Systems Development in SOA Environments, 2008, pp. 29–34.
- [35] Zghidi A., Hammouda I., Hnich B. Towards a formal API assessment. In Proc. of the 40th International Conference on Software Engineering: Companion Proceedings, 2018, pp. 398–399.
- [36] Scheller T., Kühn E. Automated measurement of API usability: The API Concepts Framework. *Information and Software Technology*, vol. 61, 2015, pp. 145-162.
- [37] López-Fernández L., García B., Gallego M., Gortázar F.: Designing and evaluating the usability of an API for real-time multimedia services in the Internet. *Multimedia Tools and Applications*, vol. 76, 2017, pp. 14247–14304.
- [38] Arquitectura y estrategia de API Un enfoque coordinado. White paper. CA technologies, 2015, 23 p. (in Spanish).

- [39] Wiegers K., Beatty J. *Software Requirements (Developer Best Practices)*, 3rd Edition. Microsoft Press, 2013, 672 p.
- [40] Mosqueira-Rey E., Alonso-Ríos D. et al. A systematic approach to API usability: Taxonomy-derived criteria and a case study. *Information and Software Technology*, vol. 97, 2018, pp. 46-63.
- [41] Laurent A. *The Design of Web APIs*. Manning Publications, 2019, 392 p.
- [42] Webber J., Parastatidis S., Robinson I. *REST in Practice: Hypermedia and Systems Architecture*. O'Reilly Media, 2010, 448 p.

Информация об авторах / Information about authors

Марибель ТЕЛЛО-РОДРИГЕС – бакалавр в области программной инженерии. Научные интересы: программная инженерия, разработка API.

Maribel TELLO-RODRÍGUEZ, Bachelor in Software Engineering. Research interests include software engineering, API design.

Хорхе Октавио ОЧАРАН-ЭРНАНДЕС, кандидат компьютерных наук, профессор факультета статистики и информатики. Область научных интересов: разработка программного обеспечения, архитектура программного обеспечения, разработка требований, разработка API.

Jorge Octavio OCHARÁN-HERNÁNDEZ, Doctor in Computing Sciences, Professor at the School of Statistics and Informatics. Research interests: software engineering, software architecture, requirements engineering, API design.

Хуан Карлос ПЕРЕС-АРРИАГА, магистр компьютерных наук, разработчик программного обеспечения. Область научных интересов: архитектура программного обеспечения, инженерия программного обеспечения, показатели программного обеспечения, программные инструменты, качество программного обеспечения.

Juan Carlos PÉREZ-ARRIAGA, Master in Computer Science, Software Developer. Research interests include software architecture, software engineering, software metrics, software tools, software quality.

Ксавье ЛИМОН, кандидат наук в области искусственного интеллекта, доцент факультета статистики и информатики. Область научных интересов: распределенные системы, архитектуры программного обеспечения, мультиагентные системы, машинное обучение.

Xavier LIMÓN, Doctor of Artificial Intelligence, Associate Professor of the Statistics and Informatics Faculty. Research interests: Distributed Systems, Software Architectures, Multi-agent systems, Machine Learning.

Анхель Х. САНЧЕС-ГАРСИЯ, кандидат наук в области искусственного интеллекта, штатный профессор факультета статистики и информатики Университета Веракруса. Область научных интересов: искусственный интеллект, машинное обучение, эволюционные вычисления, инженерия программного обеспечения.

Ángel J. SÁNCHEZ-GARCÍA, Doctor of Artificial Intelligence, Full-time Professor of the Faculty of Statistics and Informatics, University of Veracruz. Research interests: Artificial Intelligence, Machine Learning, Evolutionary Computing, Software Engineering.