

## О ПОДХОДЕ ИСПОЛЬЗОВАНИЯ HEADLESS CMS ПРИ РАЗРАБОТКЕ КОРПОРАТИВНЫХ ИНТЕРНЕТ-САЙТОВ

**Максим Евгеньевич Коваль**<sup>1</sup>, kovalmax06@gmail.com,  
**Александр Андреевич Шинкарев**<sup>2</sup>, sania.kill@mail.ru,  
**Валерия Олеговна Логиновская**<sup>1,3</sup>, valeria.login@yandex.ru

<sup>1</sup> Южно-Уральский государственный университет, г. Челябинск, Россия

<sup>2</sup> ООО «Софтмаст-Информационные Технологии», г. Челябинск, Россия

<sup>3</sup> Министерство цифрового развития, связи и массовых коммуникаций Российской Федерации, г. Москва, Россия

**Аннотация.** На сегодняшний день очень часто для управления содержимым корпоративного сайта организации применяются системы управления контентом (CMS). Можно выделить две большие группы CMS: традиционные и headless. Логика традиционных CMS объединяет бэкэнд- и фронтенд-части одной системы. Контент в данном случае оказывается связан с конкретными технологиями, архитектурой и шаблонами клиент-серверного приложения. Headless CMS – принципиально иная система управления. Как правило, она отвечает только за универсальное содержимое, которое может использоваться на любых платформах. Обе группы имеют свои плюсы и минусы. В зависимости от сложности сайта следует использовать и разные виды CMS. Традиционные CMS в большинстве поддерживают собственные шаблоны графического оформления сайта и зачастую являются сложными для использования простыми пользователями. Ввиду этого в статье будет предложен подход, который хоть и несколько усложняет процесс разработки сайта, но обеспечивает легкость и гибкость в его дальнейшей поддержке. **Цель исследования.** Рассмотреть разновидности систем управления контентом. Описать преимущества и недостатки двух видов CMS. Рассмотреть случаи, в которых стоит использовать тот или иной вид системы при разработке корпоративных сайтов организации. Предложить компонентный подход для использования в headless CMS с целью облегчения разработки, поддержки и обновления корпоративного сайта. **Материалы и методы.** Рассматриваются два вида CMS-систем. Описываются достоинства и недостатки систем управления контентом разных видов. Проводится сравнительный анализ headless CMS. Формулируются требования к системе управления контентом. Выбирается наиболее подходящая headless CMS с возможностью применения компонентного подхода при разработке корпоративного сайта. **Результаты.** В статье описываются плюсы и минусы различных видов CMS. Дается описание случаев, в которых следует использовать тот или иной вид CMS. Описываются требования к CMS для применения компонентного подхода. **Заключение.** В результате проведенного исследования проведен сравнительный анализ headless CMS и разработан краткий пример использования компонентного подхода для конструирования страниц корпоративного сайта компании.

**Ключевые слова:** корпоративный веб-сайт, разработка веб-сайтов, headless CMS, традиционная CMS, компонент

**Для цитирования:** Коваль М.Е., Шинкарев А.А., Логиновская В.О. О подходе использования headless CMS при разработке корпоративных интернет-сайтов // Вестник ЮУрГУ. Серия «Компьютерные технологии, управление, радиоэлектроника». 2022. Т. 22, № 1. С. 147–153. doi: 10.14529/ctcr220113.

## ABOUT THE APPROACH TO USE HEADLESS CMS IN DEVELOPING CORPORATE INTERNET SITES

**Maksim E. Koval**<sup>1</sup>, [kovalmax06@gmail.com](mailto:kovalmax06@gmail.com),  
**Aleksandr A. Shinkarev**<sup>2</sup>, [sania.kill@mail.ru](mailto:sania.kill@mail.ru),  
**Valeriya O. Loginovskaya**<sup>1, 3</sup>, [valeria.login@yandex.ru](mailto:valeria.login@yandex.ru)

<sup>1</sup> South Ural State University, Chelyabinsk, Russia

<sup>2</sup> LLC "Softmast-Information Technologies", Chelyabinsk, Russia

<sup>3</sup> Ministry of Digital Development, Communications and Mass Media of the Russian Federation, Moscow, Russia

**Abstract.** Today, content management systems (CMS) are often used to manage the content of an organization's corporate website. There are two large groups of CMS: traditional and headless. Both groups have their pros and cons. Different types of CMS should be used depending on the complexity of the site. Most traditional CMS support their own templates for graphic design of the site and are often difficult for ordinary users to use. In view of this, the article will propose an approach that, although it somewhat complicates the process of developing a site, but provides ease and flexibility in its further support. **Aim.** Consider the varieties of content management systems. Describe the advantages and disadvantages of the two types of CMS. Consider the cases in which it is worth using this or that type of system when developing corporate sites for an organization. Propose a component-based approach for use in a headless CMS in order to facilitate the development, maintenance and updating of a corporate website. **Materials and methods.** Two types of CMS systems are considered. The advantages and disadvantages of different types of content management systems are described. A comparative analysis of headless CMS is carried out. Requirements for the content management system are formulated. The most suitable headless CMS is selected with the possibility of using the component approach when developing a corporate website. **Results.** This article describes the pros and cons of different types of CMS. A description of the cases in which one or another type of CMS should be used is given. Describes the requirements for a CMS to apply the component approach. **Conclusion.** As a result of the study, a comparative analysis of headless CMS was carried out and a brief example of using the component approach for designing the pages of a company's corporate website was developed.

**Keywords:** corporate website, website development, headless CMS, traditional CMS, component

**For citation:** Koval M.E., Shinkarev A.A., Loginovskaya V.O. About the approach to use headless CMS in developing corporate internet sites. *Bulletin of the South Ural State University. Ser. Computer Technologies, Automatic Control, Radio Electronics*. 2022;22(1):147–153. (In Russ.) doi: 10.14529/ctcr220113.

### Введение

В современном мире многие компании и организации создают свои корпоративные сайты в интернете. На таких сайтах обычно рассказывается о том, чем занимается компания, об услугах и товарах, контакты, а также адрес расположения. Корпоративный сайт – необходимый атрибут бизнеса компании. При просмотре различных корпоративных сайтов можно сделать вывод о том, что они похожи друг на друга по своей структуре. Находясь в постоянном развитии, технологии устаревают, а следовательно, устаревают системы и приложения, которые были созданы при помощи этих технологий.

На сегодняшний день большая часть корпоративных сайтов разработаны с помощью CMS [1]. CMS, или Content Management System – информационная система или компьютерная программа, используемая для обеспечения и организации совместного процесса создания, редактирования и управления содержимым сайта. В целом CMS можно разделить на два вида: традиционные и headless. Headless CMS как правило не предоставляет возможность управления дизайном сайта, а позволяет только редактировать контент, который отображается на сайте [2]. Для управления структурой страниц сайта можно использовать классическую CMS, но такой подход не всегда удобен ввиду того, что такие системы зачастую оказываются довольно сложными для простого пользователя и могут включать множество ненужных модулей. Для решения данной проблемы

далее рассмотрим компонентный подход, который объединяет преимущества классических и headless CMS в одной системе и позволяет добавлять новые компоненты на страницы сайта без участия разработчиков [3].

### 1. Отличия headless от традиционных CMS

Главное преимущество headless CMS заключается в том, что они обеспечивают большую гибкость в дизайне веб-сайта. Фронтенд-разработчики могут извлекать данные из CMS и представлять их любым способом, который они хотят, что открывает гораздо больше возможностей [4]. Вместо того чтобы придерживаться ограниченных функций и стандартных шаблонов традиционной CMS, компании с выделенной командой фронтенд-разработчиков могут использовать headless CMS для разработки крупномасштабных и уникальных веб-сайтов, которые могут выделиться среди конкурентов.

Разделение контента и пользовательского интерфейса также обеспечивает большую гибкость во время разработки. Традиционные инструменты CMS обычно требуют использования определенных фреймворков и языков, тогда как headless CMS позволяют разработчикам использовать все, что угодно. Они также позволяют создателям контента и разработчикам работать параллельно, что может повысить общую продуктивность. Разделение также упрощает изменение конструкции в будущем. В традиционных CMS контент глубоко привязан к макету, что затрудняет изменение аспектов дизайна; для серьезных изменений может даже потребоваться совершенно новая CMS [5].

Контент, управляемый headless CMS, может легко отображаться на нескольких платформах. С другой стороны, контент и макет переплетаются в традиционной CMS, поэтому она не может поддерживать платформы, для которых они не были изначально разработаны. Headless CMS является гибкой и ориентированной на будущее, что позволяет ей использовать другие платформы, даже если они не входят в ваш текущий план развития продукта [6].

Традиционные системы управления контентом также имеют свои преимущества, самым большим из которых является то, что традиционное программное обеспечение CMS намного проще в установке и использовании. Все, от управления контентом до дизайна и пользовательского интерфейса, собрано в одном месте, что упрощает быструю разработку и публикацию контента.

Традиционная CMS может быть лучшим вариантом для небольших организаций, не имеющих необходимых ресурсов для эффективного использования автономной CMS. Это также лучший вариант для простых веб-сайтов, у которых нет функций, которые может не иметь стандартный шаблон CMS. Но если вы не хотите использовать шаблон для своего веб-сайта или хотите добавить функциональность, которую традиционные инструменты CMS не могут предоставить, в таком случае стоит рассмотреть использование headless CMS [7].

### 2. Выбор headless CMS

Одним из главных требований к CMS была возможность редактирования макета сайта, возможность конструирования страниц из набора готовых компонентов. Это было нужно для того, чтобы в дальнейшем пользователи данной CMS могли самостоятельно добавлять новые страницы на свой сайт без привлечения сторонних разработчиков. Планировалось, что система будет поддерживать server-side rendering для обеспечения возможности построения дерева компонентов на основе информации, полученной от CMS, чтобы отобразить их на странице сайта. Также требовалось, чтобы CMS поддерживала возможность веб-хуки (webhooks) для вызова функции перезагрузки сайта и его обновления.

Были рассмотрены такие headless CMS, как Strapi, Orchard. Данные CMS очень просты для понимания и использования. Однако они предоставляют возможность только плоских моделей и не обеспечивают возможность создания вложенных моделей. Такой подход подходит для создания простых сайтов, в которых не требуется сложных программных и дизайнерских решений. Также был проведен анализ Directus CMS. Данная система предоставляет функционал редактирования контента посредством таблицы. Данный функционал оказался не очень удобным ввиду не очень понятного отображения данных [8–10].

Из всех рассмотренных CMS особое внимание обратили на систему с названием Storyblok. Эта система реализует компонентный подход к управлению контентом. Поддерживает функционал для предпросмотра вносимых на сайт изменений. Storyblok дает возможность настраивать контент для различных языков, что позволяет сделать сайт мультиязычным. Имеет встроенный механизм для пагинации данных в таблицах. Все эти функции позволяют сократить время на их реализацию с нуля, а следовательно, экономят средства заказчика и обеспечивают еще больше простоты для дальнейшей работы. Но самое главное то, что данная CMS поддерживает возможность формирования деревьев компонентов, которое отображается на сайте, а значит, удовлетворяет главному требованию, которое описано выше. Единственный минус, который стоит отметить, – данная CMS предоставляется как сервис, поэтому развернуть ее на собственных серверах не получится. Однако существующие тарифы на покупку данной системы довольно невысоки, а также имеется бесплатный тариф с ограниченным функционалом [11].

### 3. Пример использования компонентного подхода

В данной главе будет представлен пример, который демонстрирует возможности компонентного подхода и динамическое встраивание компонентов в разметку страницы сайта при использовании Storyblok. Для того чтобы использовать данный подход, требуется базовое понимание работы ReactJS и Next.js. Также предполагается, что уже имеется проект и подключены все необходимые библиотеки. Инструкции по установке и подключению можно изучить в документации к Storyblok. Далее будет описан предлагаемый подход [12].

При открытии главной страницы CMS выводится список страниц сайта (рис. 1). На этом этапе можно добавить нужные страницы.

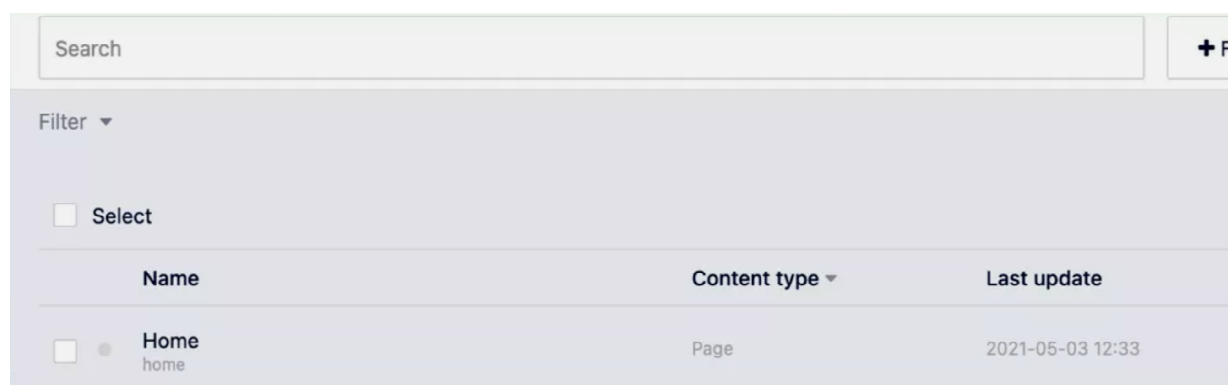


Рис. 1. Главная страница Storyblok  
Fig. 1. Main page of Storyblok

После добавления необходимых страниц следует перейти к редактированию компонентов, которые можно применять для создания структуры страниц сайта. Storyblok по умолчанию предлагает компонент Grid. Наш подход заключается в том, что нужно изменить данный компонент и создать несколько своих, это позволит строить страницы из нескольких колонок и добавлять любое содержимое в эти колонки независимо друг от друга. Для этого нужно выбрать вкладку Components в меню Storyblok. Сначала требуется создать компонент Column. На рис. 2 изображена его структура.

После этого переходим в компонент Grid и изменяем его таким образом, чтобы он мог содержать в себе компонент column (рис. 3).

Данная структура компонентов позволяет в поле body компонента Column вставлять любой другой компонент, дерево которых затем будет отображаться на странице сайта. Далее можно использовать созданные компоненты и добавлять их на нужные страницы [13–15]. Пример использования изображен на рис. 4.

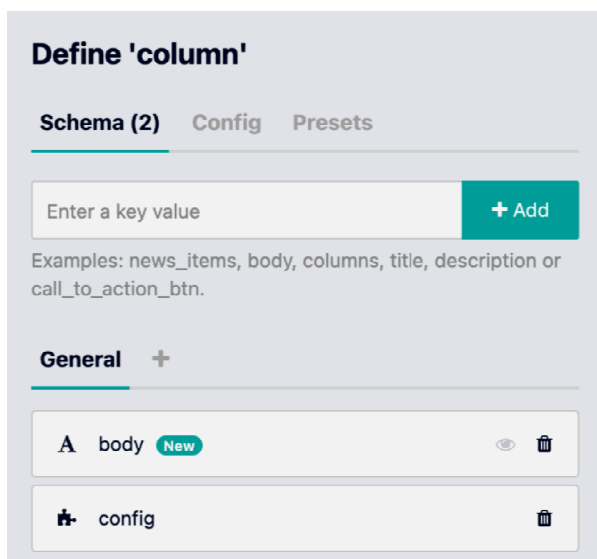


Рис. 2. Структура компонента Column  
Fig. 2. Structure of Column component

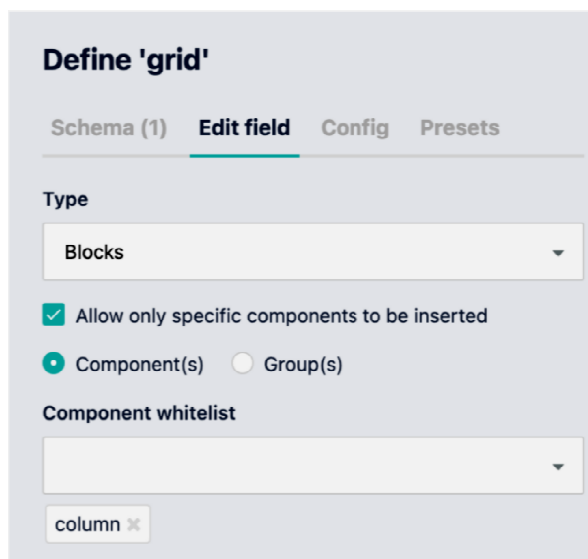


Рис. 3. Структура компонента Grid  
Fig. 3. Structure of Grid component

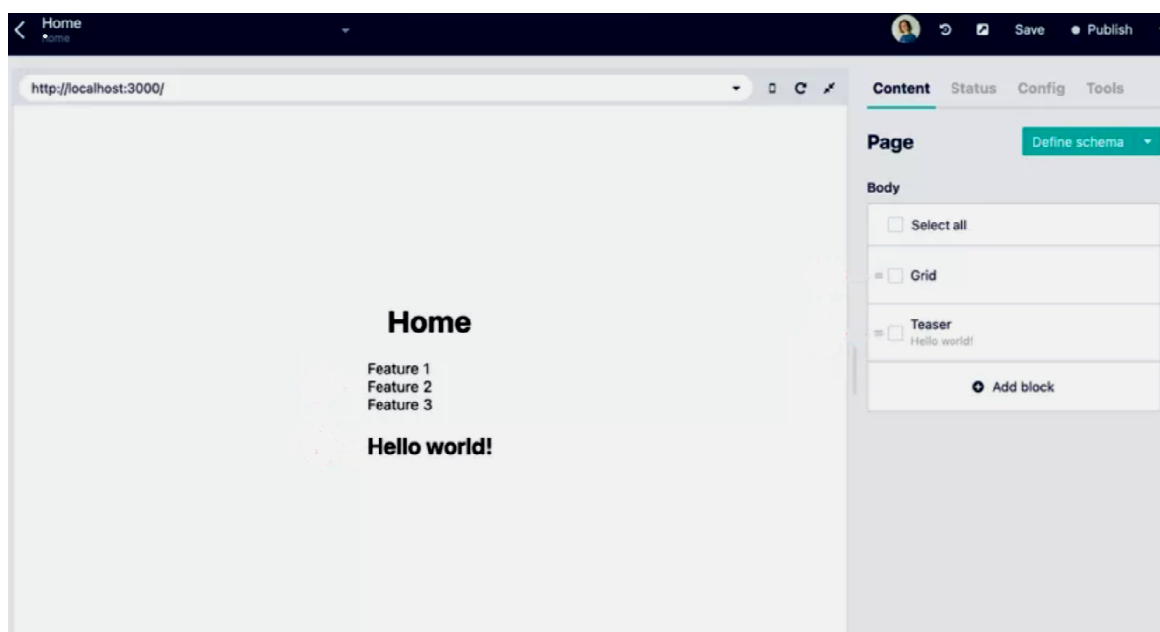


Рис. 4. Пример использования компонентов  
Fig. 4. Example of component usage

### Закключение

В свете изложенного в статье можно увидеть, что традиционные и headless CMS имеют свои достоинства и недостатки. Достоинства традиционных CMS состоят в том, что они довольно легко и быстро могут быть развернуты на серверах, однако они имеют и недостатки, такие как наличие множества зачастую ненужных модулей и сложности при редактировании страниц. Можно сказать, что если корпоративный сайт компании разработан с помощью традиционной CMS, то для его поддержки может понадобиться помощь сторонних разработчиков.

Однако существует такое решение, как headless CMS. Разрабатывая сайт с применением предложенного компонентного подхода, можно достичь того, что страницы сайта сможет редактировать пользователь, который далек от информационных технологий. Следовательно, подход позволяет избежать привлечения сторонних разработчиков для поддержки и обновления сайта, их участие требуется только при создании каркаса сайта и подключении CMS, что в свою очередь позволяет сократить затраты компании на поддержку корпоративного сайта.

### Список литературы

1. Что такое CMS сайта? URL: <https://hostiq.ua/wiki/cms/>.
2. Горнаков С.Г. *Осваиваем популярные системы управления сайтом*. М.: ДМК-пресс, 2009. 327 с.
3. Palas P. *The Ultimate Guide to Headless CMS: Everything you need to know to choose the right CMS*. Traverse City: Independently published, 2017. 68 p.
4. Tissera E. How to choose a headless CMS // *Smashing magazine*. 2021. No. 15.
5. Колисниченко Д.С. *Движок для вашего сайта. CMS Joomla!, Slaed, PHP-Nuke*. СПб.: БХВ-Петербург, 2008. 213 с.
6. Garcia V., Writer S. Should your content management system go headless? // *The American Genius*. 2021. No. 9.
7. Mixon E. Headless CMS powers personalized, omnichannel e-commerce // *TechTarget*. 2020. No. 11.
8. Madalitso D. Creating a static blog with Sapper and Strapi // *Smashing magazine*. 2020. No. 13.
9. Zablocki J. *Orchard CMS: Up and Running: ASP.NET Website Development Made Easy*. Newton: O'Reilly Media, 2012. 132 p.
10. Directus CMS. URL: <https://docs.directus.io/>.
11. Storyblok CMS. URL: <https://www.storyblok.com/>.
12. Yin M. *The Definitive Guide to Next.js and Wagtail*. Saanich: Leanpub, 2021. 162 p.
13. Blokdyk G. *Component content management system Standard Requirements*. London: 5STARCOoks, 2021. 320 p.
14. Farrell B. *Web Components in Action*. New York: Manning, 2019. 432 p.
15. Tomlinson T. *Beginning Backdrop CMS*. New York: Apress, 2016. 288 p.

### References

1. *Что такое CMS сайта?* [What is a website CMS?]. Available at: <https://hostiq.ua/wiki/cms/>. (In Russ.)
2. Gornakov S.G. *Osvaivaem populyarnye sistemy upravleniya saitom* [Mastering popular content management systems]. Moscow: DMK-press; 2009. 327 p. (In Russ.)
3. Palas P. *The Ultimate Guide to Headless CMS: Everything you need to know to choose the right CMS*. Traverse City: Independently published; 2017. 68 p.
4. Tissera E. How to choose a headless CMS. *Smashing magazine*. 2021;(15).
5. Kolisnichenko D.S. *Dvizhok dlya vashego saita. CMS Joomla!, Slaed, PHP-Nuke* [The engine for your website. CMS Joomla, Slaed, PHP-Nuke]. St. Petersburg: BHV-Petersburg; 2008. 213 p. (In Russ.)
6. Garcia V., Writer S. Should your content management system go headless? *The American Genius*. 2021;(9).
7. Mixon E. Headless CMS powers personalized, omnichannel e-commerce. *TechTarget*. 2020;(11).
8. Madalitso D. Creating a static blog with Sapper and Strapi. *Smashing magazine*. 2020;(13).
9. Zablocki J. *Orchard CMS: Up and Running: ASP.NET Website Development Made Easy*. Newton: O'Reilly Media; 2012. 132 p.
10. Directus CMS. Available at: <https://docs.directus.io/>.
11. Storyblok CMS. Available at: <https://www.storyblok.com/>.
12. Yin M. *The Definitive Guide to Next.js and Wagtail*. Saanich: Leanpub; 2021. 162 p.
13. Blokdyk G. *Component content management system Standard Requirements*. London: 5STARCOoks; 2021. 320 p.
14. Farrell B. *Web Components in Action*. New York: Manning; 2019. 432 p.
15. Tomlinson T. *Beginning Backdrop CMS*. New York: Apress; 2016. 288 p.

**Информация об авторах**

**Коваль Максим Евгеньевич**, аспирант кафедры информационно-аналитического обеспечения управления в социальных и экономических системах; Южно-Уральский государственный университет, г. Челябинск, Россия; kovalmax06@gmail.com.

**Шинкарев Александр Андреевич**, канд. техн. наук, инженер-программист, ООО «Софтмаст-Информационные Технологии», г. Челябинск, Россия; sania.kill@mail.ru.

**Логиновская Валерия Олеговна**, аспирант кафедры информационно-аналитического обеспечения управления в социальных и экономических системах, Южно-Уральский государственный университет, г. Челябинск; начальник отдела администрирования и координации Нацпрограммы Департамента координации программ и проектов, Министерство цифрового развития, связи и массовых коммуникаций Российской Федерации, г. Москва, Россия; valeria.login@yandex.ru.

**Information about the authors**

**Maksim E. Koval**, postgraduate student of the Department of Information and Analytical Support for Management in Social and Economic Systems, South Ural State University, Chelyabinsk, Russia; kovalmax06@gmail.com.

**Aleksandr A. Shinkarev**, Cand. Sci. (Eng.), software engineer at LLC “Softmast-Information Technologies”, Chelyabinsk, Russia; sania.kill@mail.ru.

Shinkarev Aleksandr Andreevich

**Valeriya O. Loginovskaya**, postgraduate student of the Department of Information and Analytical Support of Management in Social and Economic Systems, South Ural State University, Chelyabinsk, Russia; Head of the Department for Administration and Coordination of the National Program of the Department for the Coordination of Programs and Projects, Ministry of Digital Development, Telecommunications and Mass Media of the Russian Federation, Moscow, Russia; valeria.login@yandex.ru.

**Вклад авторов:** все авторы сделали эквивалентный вклад в подготовку публикации.

Авторы заявляют об отсутствии конфликта интересов.

**Contribution of the authors:** the authors contributed equally to this article.

The authors declare no conflicts of interests.

**Статья поступила в редакцию 29.11.2021; одобрена после рецензирования 13.12.2021; принята к публикации 20.12.2021.**

**The article was submitted 29.11.2021; approved after reviewing 13.12.2021; accepted for publication 20.12.2021.**