

Babayev R.A.♦

DOI: 10.25108/2304-1730-1749.iolr.2025.81.21-27

УДК: 349:681

An Error in Software Code as a Legally Significant Defect of the Result of Civil-Law Works in the Digital Environment

Abstract: The digital transformation of civil circulation has radically reshaped traditional conceptions of the results of civil-law works. Software has ceased to function merely as an auxiliary technical tool and is increasingly becoming an independent object of contractual interest. Unlike tangible results of work, digital products lack stable physical characteristics and cannot be subjected to a comprehensive quality assessment at the moment of delivery to the customer. Their properties emerge gradually in the course of operation, integration into the technological environment, and interaction with user practices.

Under these conditions, traditional notions of defectiveness of the result of work, developed with reference to the material world, lose their universal character. An error in software code represents a phenomenon of a different order: it may coexist with the functional suitability of the software product for a prolonged period and does not always transform into a legally significant defect.

The purpose of this study is to identify the legal significance of errors in software code and to formulate criteria for their qualification as defects in the result of civil-law works. The article examines the legal nature of software as a result of work, distinguishes between permissible manifestations of technical complexity and defects having legal relevance, and determines the scope and limits of the developer's civil liability.

The study employs formal-legal, functional, and systemic approaches, complemented by an analysis of doctrinal sources and judicial practice.

The study substantiates the conclusion that an error in software code, by itself, does not indicate improper quality of the result of work. The legal significance of such an error emerges exclusively in context - through its impact on the achievement of the contractual purpose, the conditions of software operation, and the limits of the developer's permissible professional risk.

Keywords: defect of the result of work; software; civil liability; functional suitability; professional risk; digital products; latent errors; contract for work and services; risk allocation.

Introduction. At the level of formal logic, the assessment of the outcome of civil-law works in software development does not differ from that of any other result: compliance with the contractual terms is either present or absent as a matter of formal logic. However, this apparent certainty quickly loses its persuasiveness when one moves from abstract criteria to an analysis of the actual behavior of a software product.

Software may undergo development and transfer to the customer and then be used for a prolonged period without any visible complaints. It performs its intended functions, yields the expected

♦ Babayev Riyad Arif oglu - the Faculty of Information Technologies and Management, Azerbaijan State Oil and Industry University; a member of the International Organization of Legal Research (Azerbaijan). E-mail: riyadbab@gmail.com

economic effect, and forms part of business operations. At the same time, such a product may contain errors - latent, hidden, and not manifested under typical operating scenarios. These errors become actual only when certain conditions coincide and raise for the law the question of the moment at which a technical deviation acquires the significance of a legally relevant defect.

Unlike tangible results of work, software does not lend itself to a final quality assessment at the moment of transfer to the customer. Its behavior emerges within a dynamic technological environment in which software solutions, hardware configurations, and user actions intertwine. In the absence of special rules directly regulating the legal significance of errors in program code, law enforcement is compelled to resort to contextual assessment, correlating the identified deviation with the contractual purpose and the limits of the contractor's professional risk [7].

Software as a Special Result of Works. To determine the legal significance of errors in program code, it is necessary to clarify the position of software within the system of civil-law categories. Any attempt to view software exclusively through the prism of traditional objects of civil rights inevitably encounters the dual nature of digital products [1, p. 32-43].

On the one hand, program code constitutes the result of intellectual activity and enjoys protection under copyright law. On the other hand, within contractual relations concerning the development of application software, it functions as a result of works whose value derives not from the mere fact of creation but from its capacity to ensure the achievement of a predetermined functional effect [4]. Functional fitness thus becomes the semantic core of the legal assessment of software quality [8, p. 1047-1090].

The behavior of a digital product depends on not only the quality of the program code, but also on hardware parameters, the architecture of the information system, and user scenarios [9, p. 88-94]. This circumstance makes any abstract assessment of software quality outside the conditions of its operation impossible.

Judicial practice confirms this approach. In *Step-Saver Data Systems, Inc. v. Wyse Technology; The Software Link, Inc.*, the court focused not on identifying an "ideal" software product but on allocating contractual risks and verifying compliance of the result with the contractual terms at the moment of delivery [10]. Law therefore evaluates software not autonomously but through the prism of the obligational relationship [2].

Errors in Program Code and the Limits of Defectiveness of the Result of Works. An error in program code manifests itself as a discrepancy between the expected and the actual behavior of a software system. However, such discrepancies do not share a uniform legal nature and require assessment within the context of the obligational relationship.

From a civil-law perspective, errors in program code fall into categories based on their impact on the achievement of the contractual purpose. Errors that exclude the possibility of using the software for its intended purpose affect the functional core of the result of works and constitute a substantial defect.

Different legal significance attaches to errors that distort individual functions of the software product without undermining its overall purpose. Their existence alone does not indicate improper quality of the result of works and requires assessment with due regard to the contractual terms and the possibility of remedy without disproportionate costs.

A distinct category consists of errors that manifest themselves exclusively under non-standard or contractually unforeseen operating conditions. In such situations, adverse consequences generally do not fall on the developer unless the parties' agreement expressly provides otherwise.

Latent errors occupy a special position within this system. Their existence at the moment of delivery does not automatically entail a breach of obligation if the software product as a whole ensures achievement of the contractual purpose. A formal approach under which any deviation from ideal functioning is automatically qualified as a defect of the result of works reflects neither the nature of programming nor the logic of civil liability [8, p. 1047-1090].

In order to prevent arbitrary qualification of technical deviations as legally relevant defects, it appears possible to formulate a minimum test for defectiveness of the result of works. An error in program code constitutes a defect in the result of civil-law works only if three conditions cumulatively apply: it objectively impedes the achievement of the contractual purpose, produces adverse effects within reasonably foreseeable operating scenarios, and exceeds the developer's permissible professional risk.

In the civil law of the Republic of Azerbaijan, this conclusion accords with Articles 752, 762, and 765 of the Civil Code, which link the contractor's liability to the presence of a substantial defect in the result of works that prevents its intended use [3].

Developer Liability and the Limits of Professional Risk. The problem of defining the limits of software developer liability occupies a special place in contemporary civil-law scholarship. Attempts to impose liability on the contractor for any adverse consequences arising from the operation of a software product, regardless of the nature of the error and the degree of control over conditions of use, lead to a methodologically flawed result [5, p. 1-28]. The more complex and technologically advanced the system, the greater the potential scope of liability becomes, which conflicts with the fundamental principles of the law of obligations [2].

Judicial practice consistently rejects such an approach. In *M.A. Mortenson Co. v. Timberline Software Corp.*, the Supreme Court of the State of Washington upheld the permissibility of contractual limitation of the developer's liability, emphasizing that a software product by its nature cannot be free from operational risks [6].

Programming as a form of professional activity takes place under conditions of inherent uncertainty. A developer cannot objectively foresee all possible scenarios for the use of complex information systems, especially when such systems integrate into a constantly evolving technological environment. Nevertheless, professional risk cannot serve as a universal basis for exemption from liability.

Conclusion. An error in program code constitutes legally distinct phenomenon that resists assessment through models of defectiveness developed for tangible results of civil-law works. In the digital environment, the mere presence of an error does not indicate, in itself, the improper quality of the result; legal significance arises only from the error's impact on the achievement of the contractual purpose and the functional fitness of the software product. Recognition of an error as a defect of the result of works is permissible only where it exceeds the limits of acceptable technical uncertainty and objectively impedes the intended use of the software.

The practical significance of this approach lies in shifting the focus from the pursuit of error-free software to the proper allocation of risks between the parties to the obligation. For contractual practice, this implies priority of functional requirements and contractual delineation of the boundaries of developer liability. For law enforcement, it entails abandoning formal comparison of software behavior with an abstract ideal and moving toward contextual assessment of the error's impact on the balance of interests of the parties. In the context of the digital transformation of civil circula-

tion, such an approach ensures normative stability of civil law without unjustified expansion of the developer's civil liability.

References

1. Ayusheeva I.Z. Digital Objects of Civil Rights (Tsifrovye ob'ekty grazhdanskikh prav). // Lex Russica, Vol. 176, No. 7, pp. 32-43. DOI: 10.17803/1729-5920.2021.176.7.032-043. Available at: <https://lexrussica.msal.ru/jour/article/viewFile/1987/1078>. (in Russian).
2. Braginsky M.I., Vitryansky V.V. Contract Law. Book 3: Contracts for Works and Services (Dogovornoe pravo. Kniga3: Dogovory o vypolnenii rabot i okazanii uslug 2nd ed. Moscow: Statut, 2011. 1055 p. Available at: URL: https://library.nlu.edu.ua/POLN_TEXT/KNIGI-2010/DOGOVOR/DOGOVOR_KN_3.pdf (in Russian).
3. Civil Code of the Republic of Azerbaijan. Approved by the Law of the Republic of Azerbaijan No. 779-IQ of 28 December 1999 (as amended and in force at the time of reference). Official text. Available at: URL: https://www.icnl.org/wp-content/uploads/Azerbaijan_Civil_Code_RUS.pdf. (in Russian).
4. Lemley M.A. Software Patents and the Return of Functional Claiming. Stanford Law Review, 2012, pp. 1–57. Available at: <https://web.stanford.edu/dept/law/ipsc/Paper%20PDF/Lemley%2C%20Mark%20-%20Paper.pdf>.
5. Levy L. B., Bell S. Y. Software Product Liability: Understanding and Minimizing the Risks. High Technology Law Journal, 1989-1990, Vol. X, pp. 1-28.
6. M.A. Mortenson Co. v. Timberline Software Corp. Supreme Court of Washington, 2000. 140 Wn.2d 568, 998 P.2d 305. Available at: <https://law.justia.com/cases/washington/supreme-court/2000/67796-4-1.html>.
7. Reed C. Making Laws for Cyberspace. Oxford: Oxford University Press, 2012. 274 p. Available at: <https://global.oup.com/academic/product/making-laws-for-cyberspace-9780199657612>.
8. Samuelson P. Allocating Responsibility for Software Errors. Berkeley Technology Law Journal, 2016, Vol. 31, No. 2, pp. 1047-1090. DOI: 10.15779/Z38WW76Z83. Available at: https://btlj.org/data/articles2016/vol31/31_2/31-2_complete.pdf.
9. Slakoper Z., Tot I. Digital Technologies and the Law of Obligations. Routledge, 2024. 254 p. Available at: <https://www.routledge.com/Digital-Technologies-and-the-Law-of-Obligations/Slakoper-Tot/p/book/9781032065038>.
10. Step-Saver Data Systems, Inc. v. Wyse Technology; The Software Link, Inc. United States Court of Appeals for the Third Circuit, 1991. 939 F.2d 91. Available at: <https://law.justia.com/cases/federal/appellate-courts/F2/939/91/31795/>.

The article was submitted: 2025 December 12

Accepted for publication: 2025 December 25

Babayev R.A.*

DOI: 10.25108/2304-1730-1749.iolr.2025.81.21-27

UOT: 349:681

Rəqəmsal mühitdə proqram təminatındakı səhv mülki-hüquqi işlərin nəticəsinin hüquqi baxımdan əhəmiyyətli qüsuru kimi

Xülasə: Mülki dövriyyənin rəqəmsal transformasiyası mülki-hüquqi işlərin nəticələri haqqında formalaşmış təsəvvürləri köklü şəkildə dəyişdirmişdir. Proqram təminatı artıq köməkçi texniki vasitə olmaqdan çıxaraq getdikcə daha çox müstəqil öhdəlik marağının obyektinə kimi çıxış edir. Maddi xarakterli iş nəticələrindən fərqli olaraq, rəqəmsal məhsullar sabit fiziki xüsusiyyətlərə malik deyil və sifarişçiyə təhvil verildiyi anda keyfiyyətinin tam və hərtərəfli qiymətləndirilməsinə imkan vermir. Onların xüsusiyyətləri tədricən — istismar prosesində, texnoloji mühitə inteqrasiya zamanı və istifadəçi praktikaları ilə qarşılıqlı əlaqə çərçivəsində üzə çıxır.

Bu şəraitdə maddi aləm üçün formalaşdırılmış iş nəticəsinin qüsurluluğu ilə bağlı ənənəvi yanaşmalar universallığını itirir. Proqram kodundakı səhv fərqli mahiyyətə malik fenomen kimi çıxış edir: o, uzun müddət proqram məhsulunun funksional yararlılığı ilə yanaşı mövcud ola və hər zaman hüquqi baxımdan əhəmiyyətli qüsura çevrilməyə bilər.

Bu tədqiqatın məqsədi proqram kodundakı səhvlərin hüquqi əhəmiyyətini müəyyən etmək və onların mülki-hüquqi işlərin nəticəsinin qüsuru kimi kvalifikasiyası üçün meyarlar formalaşdırmaqdan ibarətdir. İşdə proqram təminatının iş nəticəsi kimi hüquqi təbiəti təhlil edilir, texniki mürəkkəbliyin yol verilən təzahürləri ilə hüquqi əhəmiyyət kəsb edən qüsurlar arasında fərqləndirmə aparılır, həmçinin proqram tərtibatçısının mülki-hüquqi məsuliyyətinin həcmi və hədləri müəyyən edilir.

Tədqiqatın metodoloji əsasını formal-hüquqi, funksional və sistemli yanaşmalar təşkil edir; bu yanaşmalar doktrinal mənbələrin və məhkəmə təcrübəsinin təhlili ilə tamamlanır.

Əsaslandırılır ki, proqram kodunda səhvin mövcudluğu öz-özlüyündə iş nəticəsinin keyfiyyətsiz olmasına dəlalət etmir. Səhvin hüquqi əhəmiyyəti yalnız kontekst daxilində — onun müqavilə məqsədinə nail olunmasına təsiri, proqram məhsulunun istismar şərtləri və tərtibatçının yol verilən peşəkar riskinin hədləri nəzərə alınmaqla açıqlanır.

Açar sözlər: iş nəticəsinin qüsuru; proqram təminatı; mülki-hüquqi məsuliyyət; funksional yararlılıq; peşəkar risk; rəqəmsal məhsullar; latent səhvlər; podrat müqaviləsi; risklərin bölgüsü.

Məqalə daxil olmuşdur: 12 dekabr 2025-ci il

Çapa qəbul edilmişdir: 25 dekabr 2025-ci il

* **Babayev Riyad Arif oğlu** – Azərbaycan Dövlət Neft və Sənaye Universitetinin “İnformasiya texnologiyaları və idarəetmə” fakültəsi, Beynəlxalq Hüquq Tədqiqatları Təşkilatının üzvü (Azərbaycan). E-mail: riyadbab@gmail.com

Бабаев Р.А.♦

DOI: 10.25108/2304-1730-1749.iolr.2025.81.21-27

УДК: 349:681

Ошибка в программном коде как юридически значимый дефект результата гражданско-правовых работ в цифровой среде

Аннотация: Цифровая трансформация гражданского оборота радикально изменила представления о результатах гражданско-правовых работ. Программное обеспечение перестало быть вспомогательным техническим инструментом и всё чаще становится самостоятельным объектом обязательственного интереса. В отличие от материальных результатов работ цифровые продукты лишены устойчивых физических характеристик и не поддаются исчерпывающей оценке качества в момент передачи заказчику. Их свойства проявляются постепенно - в процессе эксплуатации, интеграции в технологическую среду и взаимодействия с пользовательскими практиками.

В этих условиях традиционные представления о дефектности результата работ, сформированные применительно к материальному миру, утрачивают универсальность. Ошибка в программном коде выступает феноменом иного порядка: она может длительное время сосуществовать с функциональной пригодностью программного продукта и не всегда трансформироваться в юридически значимый дефект.

Цель настоящего исследования состоит в выявлении юридического значения ошибок в программном коде и формировании критериев их квалификации в качестве дефекта результата гражданско-правовых работ. В работе анализируется правовая природа программного обеспечения как результата работ, проводится разграничение между допустимыми проявлениями технической сложности и дефектами, имеющими правовое значение, а также определяются объём и пределы гражданско-правовой ответственности разработчика.

Методологическую основу исследования образуют формально-юридический, функциональный и системный подходы, дополненные анализом доктринальных источников и судебной практики.

Обосновывается вывод о том, что наличие ошибки в программном коде само по себе не свидетельствует о ненадлежащем качестве результата работ. Юридическое значение ошибки раскрывается исключительно в контексте - через её влияние на достижение договорной цели, условия эксплуатации программного продукта и пределы допустимого профессионального риска разработчика.

Ключевые слова: дефект результата работ; программное обеспечение; гражданско-правовая ответственность; функциональная пригодность; профессиональный риск; цифровые продукты; латентные ошибки; договор подряда; распределение рисков.

♦ Бабаев Рияд Ариф оглы – факультет “Информационных технологий и управления” Азербайджанского государственного университета нефти и промышленности, член Международной организации международных исследований (Азербайджан). E-mail: riyadbab@gmail.com

Библиография

1. Аюшеева И. З. Цифровые объекты гражданских прав // Lex Russica. - Т. 176, № 7. - С. 32–43. - DOI: 10.17803/1729-5920.2021.176.7.032-043. [Электронный ресурс]. URL: <https://lexrussica.msal.ru/jour/article/viewFile/1987/1078>.
2. Брагинский М. И., Витрянский В. В. Договорное право. Книга 3: Договоры о выполнении работ и оказании услуг. - 2-е изд. - Москва: Статут, 2011. - 1055 с. [Электронный ресурс]. URL: https://library.nlu.edu.ua/POLN_TEXT/KNIGI-2010/DOGOVOR/DOGOVOR_KN_3.pdf
3. Гражданский кодекс Азербайджанской Республики: утв. Законом Азербайджанской Республики от 28 декабря 1999 г. № 779-IQ (в ред. действующей на момент обращения). - Официальный текст. [Электронный ресурс]. URL: https://www.icnl.org/wp-content/uploads/Azerbaijan_Civil_Code_RUS.pdf
4. Lemley M. A. Software Patents and the Return of Functional Claiming // Stanford Law Review. - 2012. - P. 1-57. [Электронный ресурс]. URL: <https://web.stanford.edu/dept/law/ipsc/Paper%20PDF/Lemley%2C%20Mark%20-%20Paper.pdf>.
5. Levy L. B., Bell S. Y. Software Product Liability: Understanding and Minimizing the Risks // High Technology Law Journal. – 1989-1990. - Vol. X. - P. 1-28.
6. M.A. Mortenson Co. v. Timberline Software Corp. // Supreme Court of Washington. 2000. - 140 Wn.2d 568, 998 P.2d 305. [Электронный ресурс]. URL: <https://law.justia.com/cases/washington/supreme-court/2000/67796-4-1.html>.
7. Reed C. Making Laws for Cyberspace. - Oxford : Oxford University Press, 2012. —274 p. [Электронный ресурс]. URL: <https://global.oup.com/academic/product/making-laws-for-cyberspace-9780199657612>.
8. Samuelson P. Allocating Responsibility for Software Errors // Berkeley Technology Law Journal. - 2016. - Vol. 31, No. 2. - P. 1047-1090. - DOI: 10.15779/Z38WW76Z83. [Электронный ресурс]. URL: https://btlj.org/data/articles2016/vol31/31_2/31-2_complete.pdf.
9. Slakoper Z., Tot I. Digital Technologies and the Law of Obligations. Routledge, 2024. - 254 p. [Электронный ресурс]. URL: <https://www.routledge.com/Digital-Technologies-and-the-Law-of-Obligations/Slakoper-Tot/p/book/9781032065038>.
10. Step-Saver Data Systems, Inc. v. Wyse Technology; The Software Link, Inc.// United States Court of Appeals for the Third Circuit. - 1991. - 939 F.2d 91. [Электронный ресурс]. URL: <https://law.justia.com/cases/federal/appellate-courts/F2/939/91/31795/>.

Дата поступления: 12 декабря 2025 г.

Дата принятия в печать: 25 декабря 2025 г.