

# GRUPOWE BŁĘDY POZNAWCZE W DECYZJACH ZESPOŁÓW PROJEKTOWYCH IT I SPOSOBY ICH OGRANICZANIA

*Marcin Kaczyński*<sup>1</sup>

## Streszczenie

Tematem artykułu są behawioralne uwarunkowania decyzji podejmowanych w trakcie prowadzenia projektu informatycznego przez zespół projektowy IT. Decyzje projektowe są zazwyczaj podejmowane w warunkach ryzyka i wymagają wnikliwej analizy na podstawie dostępnych informacji. Decyzje zespołu projektowego IT są skomplikowane i zagrożone wpływem heurystyk i wystąpieniem błędów poznawczych. Artykuł wskazuje na konieczność większej integracji nauki ekonomii behawioralnej i metodyk zarządzania projektami IT. Pozwoli to na lepsze zrozumienie podstaw podejmowania decyzji projektowych i ograniczenie wpływu błędów poznawczych na te decyzje. Zespoły projektowe IT mogą wykorzystywać konkretne techniki zwinnej metodyki zarządzania projektami – Scrum w celu ograniczenia wpływu błędów poznawczych na decyzje projektowe.

**Słowa kluczowe:** ekonomia behawioralna, teoria decyzji, zarządzanie projektami IT, błędy poznawcze.

## 1. Wstęp

Zespoły projektowe podejmują decyzje na każdym etapie projektu. Decyzje w warunkach niepewności i ryzyka podejmowane są zarówno przez kierowników projektów, jak i cały zespół projektowy. Proces ten zależy od wielu czynników i uwarunkowań wynikających ze specyfiki produktu i projektu, otoczenia projektowego, dojrzałości zespołu projektowego i klienta. Obok czynników związanych ze specyfiką produktu informatycznego, występują czynniki psychologiczne wpływające na zachowania zespołu projektowego indywidualnie i grupowo. Behawioralne zabarwienie zachowań zespołu związane z podejmowaniem decyzji jest

---

<sup>1</sup> Uniwersytet Gdański, Wydział Zarządzania / University of Gdańsk, Faculty of Management, e-mail: mkaczynski@gmail.com.

niezwykle interesujące. Analiza tych zachowań i poznanie psychologii podejmowania decyzji pozwala spojrzeć na procesy zarządzania projektem w innym świetle. Celem artykułu jest przedstawienie psychologicznego ujęcia decyzji i procesu decyzyjnego, analiza racjonalności, sklasyfikowanie błędów poznawczych i wskazanie behawioralnych uwarunkowań decyzji podejmowanych przez zespoły projektowe IT oraz wskazanie technik zwinnej metodyki zarządzania projektami – Scrum, ograniczających wpływ błędów poznawczych. W pracy zastosowano metodę studium literatury, a także obserwacje własne z realizacji projektów IT. Autor od piętnastu lat zarządza zespołami projektowymi IT z użyciem metodyk kaskadowych i zwinnych.

## 2. Proces podejmowania decyzji

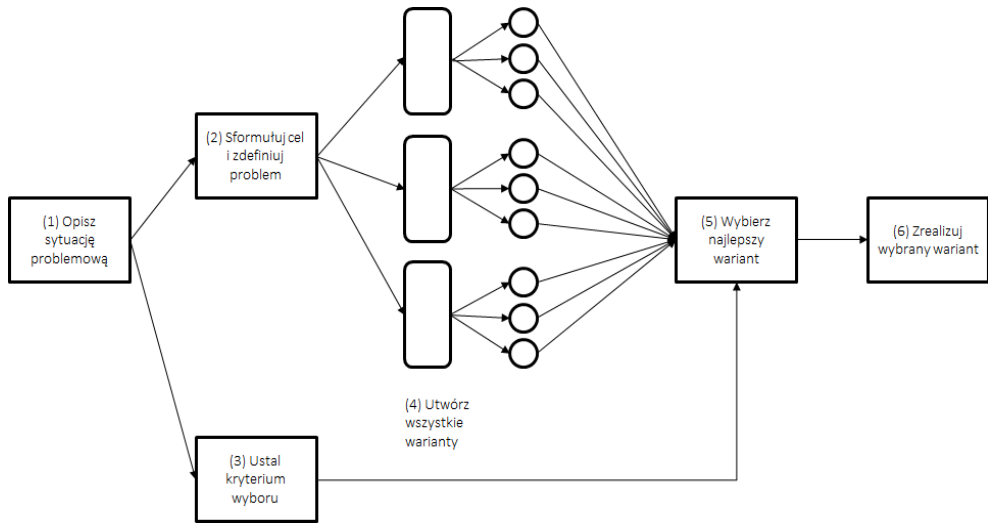
Decyzja to efekt dokonanego przez grupę wyboru w postaci preferowanego wariantu działania. Decyzje projektowe podejmowane są w warunkach niepewności lub ryzyka. Decyzje podejmowane w warunkach niepewności to decyzje podejmowane wówczas, gdy grupa nie jest w stanie przypisać prawdopodobieństwa wystąpienia tych okoliczności, które nie poddają się oddziaływaniom z jej strony, ale z obecnością których należy się liczyć dokonując wyboru. O decyzjach w warunkach ryzyka mówi się, gdy przewidywanym przez grupę okolicznościom można przypisać prawdopodobieństwo ich wystąpienia (Kozłowski, Piotrowski, 2009). Podjęcie decyzji poprzedzone jest procesem myślowym i wiąże się z konkretnym zachowaniem grupy, a także niesie za sobą często poważne i długotrwałe skutki.

Racjonalność zachowań ekonomicznych może być rozpatrywana w kontekście dwóch podstawowych grup modeli: modelu normatywnego i modelu racjonalności ograniczonej.

Etapy składające się na model racjonalnego podejmowania decyzji to (Harrison, 1995):

1. Zdefiniowanie problemu,
2. Określenie kryteriów decyzji,
3. Rozważenie kryteriów,
4. Wyszukanie innych możliwości,
5. Ocena każdej możliwości według każdego kryterium,
6. Podjęcie optymalnej decyzji.

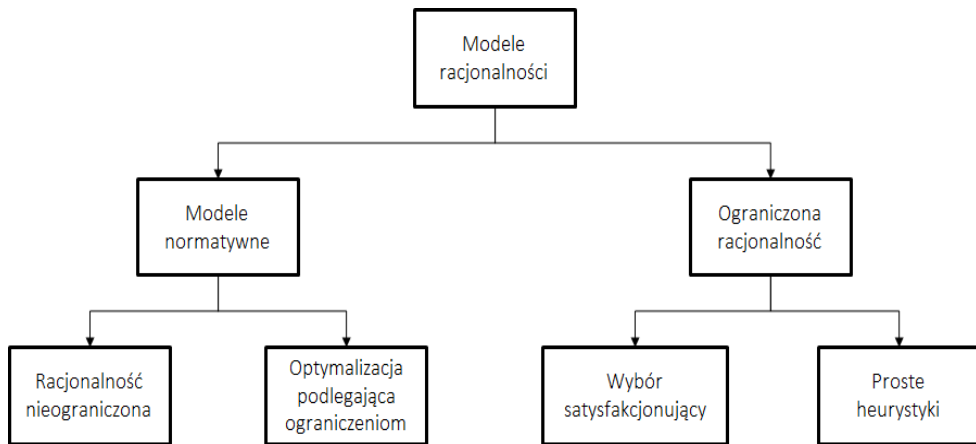
Rysunek 1 przedstawia model podejmowania decyzji racjonalnej, w wyniku której grupa maksymalizuje użyteczność.



**Rys. 1.** Decyzja racjonalna – model normatywny

*Źródło:* Steers, 1981.

Racjonalność ograniczona polega na wykorzystywaniu prostych i szybkich heurystyk oraz dokonywaniu satysfakcjonujących, ale niekoniecznie optymalnych wyborów (Zaleśkiewicz, 2011). Rysunek 2 przedstawia podział modeli racjonalności.



**Rys. 2.** Modele racjonalności

*Źródło:* Zaleśkiewicz, 2011.

Herbert Simon, krytykując założenie o racjonalności podmiotów ekonomicznych, zwrócił uwagę na następujące ograniczenia dotyczące racjonalności procesu podejmowania decyzji (Simon, 1955):

1. Decydenci realizują zwykle nie jeden cel, który chcieliby zmaksymalizować, ale wiele – często niezgodnych – celów.

2. Brak jest pełnej informacji o alternatywach i ich cechach.
3. Decydenci nie będą na ogół w stanie dokonać wszystkich potrzebnych obliczeń i porównań alternatyw wyboru.

Na skutek tych ograniczeń zasadą wyboru staje się zasada satysfakcji, polegająca na poszukiwaniu takiej opcji, która jest wystarczająco dobra. Decydenci po znalezieniu opcji wystarczająco dobrej nie będą sprawdzać pozostałych alternatyw, ponieważ znaleziona opcja spełnia ich wymagania. Teoria racjonalności ograniczonej skupia się na procesie decyzyjnym w wyszukiwaniu satysfakcjonującego rozwiązania. W tym paradygmacie zakłada się, że istnieją proste reguły (heurystyki), pozwalające szybko i niewielkim kosztem znaleźć optymalne lub prawie optymalne rozwiązanie (Tyszka, 2004).

### **3. Wybrane błędy poznawcze popełniane przez zespoły projektowe IT**

Decyzje podejmowane w trakcie trwania projektu IT podejmowane są w warunkach niepewności i ryzyka. Zespół projektowy obciążony jest niebezpieczeństwem występowania heurystyk, które prowadzą do błędów poznawczych.

Heurystyki to uproszczone reguły wnioskowania, nieformalne i intuicyjne strategie poznawcze. Zostały one po raz pierwszy nazwane i opisane przez Tversky'ego i Kahnemana w stworzonej przez nich w 1979 roku teorii perspektywy (Kahneman, Tversky, 1979). Autorzy nie negują użyteczności tych heurystyk, ale zwracają uwagę, że prowadzą one do systematycznych błędów. Wadą używania heurystyk jest duża zawodność w sytuacjach nowych i nietypowych.

Wyróżnia się trzy główne heurystyki z grupy heurystyk wydawania sądów:

1. Heurystykę zakotwiczenia i dostosowania, polegającą na oparciu się na jakiejś informacji, a następnie zmodyfikowaniu jej i dostosowaniu się do niej (Kahneman i inni, 1982). W wielu sytuacjach zespoły projektowe dokonują estymacji na podstawie początkowych wartości, które są podstawą do sformułowania problemu decyzyjnego, czego skutkiem jest kotwiczenie.
2. Heurystykę dostępności, polegającą na przypisywaniu większego prawdopodobieństwa zdarzeniom, które łatwiej przywołać do świadomości i które są bardziej nacechowane emocjonalnie (Kahneman, 2012). Zgodnie z tą heurystyką zespoły projektowe wnioskujeją bądź określają prawdopodobieństwa zdarzeń, kierując się łatwością, z jaką są w stanie odtworzyć przeszłe zdarzenia. Łatwość z jaką zespół projektowy sięga do obrazów zdarzeń, sprawia, że dostępność jest przydatnym elementem w procesie prognozowania, jednak prowadzić może do licznych błędów.
3. Heurystykę reprezentatywności polegającą na dokonywaniu klasyfikacji na podstawie częściowego podobieństwa do przypadku reprezentatywnego (Kahneman, 2012). Zespoły projektowe, posługujące się tą heurystyką, formułują ogólne wnioski na podstawie małej próby, bądź pojedynczych zdarzeń.

Zespoły projektowe podejmując decyzje przy użyciu heurystyk narażone są na różnorodne błędy poznawcze, których występowania w większości przypadków nie są świadome. Wybrane błędy poznawcze zostaną opisane poniżej.

Efekt potwierdzenia jest tendencją do preferowania informacji, które potwierdzają wcześniejsze oczekiwania, niezależnie od tego, czy te informacje są prawdziwe, jednocześnie ignorując informacje niezgodne z przekonaniem (Amott, 2006). Powoduje to, że zespół projektowy poszukuje informacji i zapamiętuje je w sposób selektywny, interpretując informacje w błędny sposób. Efekt ten jest szczególnie silny w przypadku zagadnień wywołujących silne emocje i dotyczących mocno ugruntowanych opinii. Efekt potwierdzenia jest widoczny w kontekście testów oprogramowania jako „efekt pozytywnych testów”. Przejawia się on w tworzeniu wielokrotnie większej liczby scenariuszy testowych potwierdzających prawidłowe działanie oprogramowania, aniżeli scenariuszy negatywnych, mających wykryć błąd. Testerzy przygotowują dane testowe, które są zgodne ze specyfikacją aplikacji, a ignorują sprawdzenie danymi, na które aplikacja nie jest gotowa (Leventhall i inni, 1993). Efekt potwierdzenia uwidacznia się także podczas przeszukiwania informacji w dokumentacji oprogramowania – poszukuje się informacji, które potwierdzają zdobytą wcześniej wiedzę (Graaf i inni, 2014).

Tendencja do przywiązywania większej wagi do informacji, która potwierdza przekonania, może dotyczyć wiele obszarów wytwarzania oprogramowania. Przez efekt potwierdzenia członkowie zespołów projektowych są często nadmiernie optymistyczni w kontekście szacowania czasu potrzebnego do wykonania zadań technicznych. Programiści niechętnie też sprawdzają możliwości dużych zmian w istniejących funkcjonalnościach lub badają wykluczające się wymagania (Mohan i inni, 2009). Podczas debugowania, efekt potwierdzenia czasami myli programistów w kwestii przyczyn błędnego działania systemu (Stacy, MacMillan, 1995).

Efekt wiedzy po fakcie, nazywany też efektem „wiedziałem-że-tak-będzie” lub pełzającym determinizmem, to tendencja do oceniania przeszłych wydarzeń jako bardziej przewidywalnych, niż rzeczywiście były (Leary, 1982). Prawdopodobnie wynika z tego, że wiedza na ich temat jest bardziej dostępna niż wiedza na temat możliwości, które się nie zdarzyły. Zespoły projektowe mają tendencję również do pamiętania swoich własnych przewidywań jako dokładniejszych i celniejszych, niż rzeczywiście były. Brak zarchiwizowanych danych z poprzednich projektów może nasilać efekt wiedzy po fakcie (Cunha, Moura, 2015).

Efekt aureoli, inaczej zwany efektem halo, to tendencja do automatycznego, pozytywnego lub negatywnego, przypisywania cech osobowościowych na podstawie pierwszego wrażenia (Long-Crowell, 2015). Przesłuchując kandydatów na członków zespołu projektowego osoba odpowiedzialna za rekrutację może błędnie ocenić charyzmatycznego kandydata jako bardziej doświadczonego technicznie.

Efekt kosztów utopionych to tendencja do irracjonalnego ponoszenia dodatkowych kosztów w sytuacji, gdy poprzednia inwestycja została dokonana, w porównaniu do analogicznej sytuacji, gdyby tej inwestycji nie było (Arkes, 1999). Tendencje związane z tym efektem mogą być zakorzenione w cechach osobowości

kierowników projektów, w tym w uporze i krótkowzroczności.

Efekt framingu, inaczej zwany efektem ram interpretacyjnych to nieuzasadniony wpływ sformułowania problemu na ludzkie przekonania i preferencje (Kahneman, 2012). Badania dowodzą, że wątpliwe wymagania biznesowe do implementacji w oprogramowaniu prowadzi do jego utrwalenia i jednocześnie ogranicza kreatywność w fazie projektowania (Mohanani i inni, 2014).

Hiperboliczne obniżenie wartości to preferowanie mniejszej nagrody w bliższej przyszłości od większej nagrody w dalszej przyszłości (Wirfs-Brock, 2007). Efekt ten prowadzi programistów do preferowania szybkich poprawek i prostego refaktoringu, czyli przebudowy istniejącego kodu programu od bardziej ambitnych jego zmian.

Efekt autorytetu jest to zachowanie polegające na postępowaniu tak, jak robią to inni (Vandenbos, 2007). Badania wskazują, że niewystarczające przetwarzanie informacji prowadzi członków zespołów projektowych do bezkrytycznej akceptacji wymagań klienta. Efekt autorytetu powoduje też akceptację decyzji kierownika projektu bez analizy innych alternatyw (Ralph, 2013).

#### **4. Techniki ograniczające wpływ błędów poznawczych na grupowe decyzje w projektach informatycznych**

Błędy poznawcze występujące w codziennej pracy zespołów projektowych IT mają charakter systemowy i nie można ich całkowicie wyeliminować. Natomiast badania prowadzone nad zespołami IT dowodzą, że można ograniczyć ich wpływ na decyzje projektowe (Cunha, Moura, 2015). Doświadczenie autora w zarządzaniu projektami informatycznymi potwierdza te wnioski.

Pierwszy krok w celu ograniczenia działania błędów poznawczych to uświadomienie sobie przez zespół projektowy IT ich istnienia i sytuacji, w których błędy wpływają na efektywność podejmowania decyzji. Kolejnym krokiem będzie ograniczenie ich wpływu poprzez użycie konkretnych technik proponowanych przez jedną z metodyk zwinnych – Scrum, której autorami są Ken Schwaber i Jeff Sutherland.

Scrum to ramy postępowania, dzięki którym zespoły IT mogą z powodzeniem rozwiązywać złożone problemy adaptacyjne, aby w sposób produktywny i kreatywny wytwarzać produkty o najwyższej wartości. Scrum został osadzony w teorii empirycznej kontroli procesu. Scrum wykorzystuje podejście iteracyjne i przyrostowe w celu zwiększenia przewidywalności i lepszej kontroli ryzyka. Realizacja empirycznej kontroli procesu opiera się na trzech filarach: przejrzystości, inspekcji i adaptacji (Schwaber, Sutherland, 2017).

Ramy Scrum opierają się na rolach, artefaktach i wydarzeniach (Kaczor, 2016), których charakter odzwierciedla filary Scrum. Badania oraz doświadczenia autora wskazują, że użycie konkretnych technik Scrum w trakcie trwania projektu może ochronić zespół projektowy przed wpływem wielu błędów poznawczych (Cunha, Moura, 2015). Przykładowe techniki zweryfikowane przez badaczy to:

1. Wykres spalania obrazujący postęp prac nad iteracją (Cunha, Moura, 2015).

2. Elastyczne oddolne planowanie iteracji (Cunha, Moura, 2015).
3. Organizowanie sesji demo produktu na zakończenie iteracji (Cunha, Moura, 2015).
4. Codzienne spotkania zespołu (Cunha, Moura, 2015).
5. Zbieranie informacji zwrotnej od klientów po pokazaniu demo produktu (Cunha, Moura, 2015).
6. Nacisk na zespół testerów, aby konstruowali scenariusze testowe w taki sposób, żeby wykrywały błędy w systemie, a nie dowodziły jego prawidłowego działania (Stacy, MacMillan, 1995).
7. Stworzenie i utrzymywanie macierzy identyfikowalności (Jain i inni, 2006).
8. Aktywny udział wszystkich interesariuszy projektu i ekspertów domeny w spotkaniach retrospekcji iteracji (Cunha, Moura, 2015).
9. Użycie podwójnej pętli nauki, pociągające za sobą modyfikację celów lub reguł decyzyjnych w świetle doświadczenia (Argyris, 1991).
10. Sformułowanie pytania dotyczącego estymacji w taki sposób, aby ograniczyć wpływ nadmiernego optymizmu (Jorgensen, Halkjelsvik, 2010).

Doświadczenia autora wskazują, że oparcie procesu zarządzania projektem na filarach Scrum – przejrzystości, inspekcji i adaptacji oraz użycie ról projektowych, artefaktów oraz wydarzeń proponowanych w ramach tej metodyki, a w szczególności wyżej wymienionych technik, pozwalają na ograniczenie wpływu błędów poznawczych na decyzje projektowe zespołu. Szczególne znaczenie ma to w projektach informatycznych, które są unikatowe ze względu na charakter produktu – systemu informatycznego i całego procesu.

## 5. Zakończenie

Interdyscyplinarne podejście do procesu podejmowania decyzji projektowych przez zespoły projektowe, uwzględniające psychologiczne aspekty podejmowanych decyzji pozwala zidentyfikować błędy poznawcze, na które narażony jest zespół używający heurystyk w trakcie podejmowania decyzji. Zespół projektowy uświadomiony narażeniem na popełnienie błędów poznawczych, jest w stanie ograniczyć ich wpływ poprzez użycie konkretnych technik oferowanych przez zwinną metodykę zarządzania projektami – Scrum. Oparcie projektu na filarach Scrum i użycie technik ograniczających wpływ błędów poznawczych przyczyni się do zwiększenia prawdopodobieństwa osiągnięcia założonych celów projektu.

## Bibliografia

1. Amott D. (2006), *Cognitive biases and decision support systems development: a design science approach*, „Information Systems journal”, vol. 16, no. 1, s. 55–78.
2. Argyris C. (1991), *Teaching smart people how to learn*, „Harvard Business Review”, 69 (3), s. 99–109.
3. Arkes H.R., Ayton P. (1999), *The sunk cost and concordance effects: Are humans less rational than lower animals?*, „Psychological bulletin”, vol. 125, no. 5, s. 591.

4. Cunha J.A.O.G. da, Moura H.P. de (2015), *Towards a substantive theory of project decisions in software development project based organizations: A cross-case analysis of it organizations from Brazil and Portugal*, Information Systems and Technologies (CISTI), 10<sup>th</sup> Iberian Conference on IEEE, s. 1–6.
5. Graaf K.A. De, Liang P., Tang A., Van Vliet H. (2014), *The impact of prior knowledge on searching in software documentation*, Proceedings of the 2014 ACM symposium on Document engineering, ACM, s. 189–198.
6. Harrison E.F. (1995), *The Managerial Decision-Making Process*, Houghton Mifflin, Boston, s. 75–113.
7. Jain R., Muro J., Mohan K. (2006), *A cognitive perspective on pair programming*, AMCIS 2006 Proceedings, s. 444.
8. Jorgensen M., Halkjelsvik T. (2010), *The effects of request formats on judgment—based effort estimation*, „Journal of Systems and Software”, vol. 83, no. 1, s. 29–36.
9. Kaczor K., *Scrum i nie tylko. Teoria i praktyka w metodach Agile*, Wydawnictwo Naukowe PWN, Warszawa, 2016, s. 108.
10. Kahneman D. (2012), *Pułapki myślenia. O myśleniu szybkim i wolnym*, Media Rodzina, Poznań, s. 175, 201, 482.
11. Kahneman D., Slovic P., Tversky A. (1982) *Judgment under uncertainty: heuristics and biases*, Cambridge University Press, New York.
12. Kahneman D., Tversky A. (1979), *Prospect Theory: An Analysis of Decision under Risk*, *Econometrica*, XLVII, s. 263–291.
13. Koźmiński A.K., Piotrowski W. (2009), *Zarządzanie. Teoria i praktyka*, Wydawnictwo Naukowe PWN, Warszawa, s. 119.
14. Leary M.R. (1982), *Hindsight Distortion and the 1980 Presidential Election*, „Personality and Social Psychology Bulletin”, vol. 8, no. 2, s. 257–263.
15. Leventhal L.M., Teasley B.M., Rohlman D.S., Instone K. (1993), *Positive test bias in software testing among professionals: A review*, International Conference on Human—Computer Interaction. Springer, s. 210–218.
16. Long-Crowell E. (2015), *The halo effect: definition, advantages & disadvantages*, „Psychology”, vol. 104.
17. Mohan K., Kumar N., Benbunan-Fich R. (2009), *Examining communication media selection and information processing in software development traceability: An empirical investigation*, „IEEE Transactions on Professional Communication”, vol. 52, no. 1, s. 17–39.
18. Mohanani R., Ralph P., Shreeve B. (2014), *Requirements fixation*, Proceedings of the 36<sup>th</sup> International Conference on Software Engineering. ACM, s. 895–906.
19. Ralph P. (2013), *Possible core theories for software engineering*, Software Engineering (GTSE), 2013 2<sup>nd</sup> SEMAT Workshop on a General Theory of, IEEE, s. 35–38.
20. Schwaber K., Sutherland J., *Scrum Guide The Definitive Guide to Scrum: The Rules of the Game*, November 2017, <https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf#zoom=100>, dostęp: styczeń 2019.
21. Simon H. (1955), *A Behavioral Model of Rational Choice*, „Quarterly Journal of Economics”, No. 69.
22. Stacy W., MacMillan J. (1995), *Cognitive bias in software engineering*, Communications of the ACM, vol. 38, no. 6, s. 57–63.
23. Steers R.M. (1981), „Introduction to Organizational Behaviour”, Goodyear Publishing, Santa Monica, California, s. 236.

24. Tyszka T. (2004), *Psychologia ekonomiczna*, Gdańskie Wydawnictwo Psychologiczne, Gdańsk, s. 56.
25. VandenBos G.R. (Ed) (2007), *APA Dictionary of Psychology*, American Psychological Association.
26. Wirfs-Brock R.J. (2007), *Giving Design Advice*, „IEEE Software”, vol. 24, no. 4, s. 13–15.
27. Zaleśkiewicz T. (2011), *Psychologia ekonomiczna*, PWN, Warszawa, s. 68.

## **COGNITIVE BIASES IN GROUP DECISIONS OF IT PROJECT TEAMS AND WAYS TO LIMIT THEM**

### **Abstract**

The article covers behavioral determinants of decisions made by IT project teams during the whole project cycle. Project decisions are usually made in risk conditions and require careful analysis based on available information. The project decisions are usually complicated and threatened by the influence of heuristics and the occurrence of cognitive biases. The article indicates the need for greater integration of behavioral economics and project management methodologies. This will allow a better understanding of the basics of making design decisions and reducing the impact of cognitive biases on these decisions. Project team members can use specific techniques provided by agile project management methodologies, in particular Scrum, to reduce the impact of cognitive biases on project decisions.

**Key words:** behavioral economy, decision theory, IT project management, cognitive biases.