INFORMATYKA EKONOMICZNA BUSINESS INFORMATICS 1(31) • 2014

ISSN 1507-3858

**Mirosława Lasek, Aleksandra Adamus**
Uniwersytet Warszawski

# KIEDY WARTO STOSOWAĆ METODYKI ZWINNE (*AGILE METHODOLOGIES*) W ZARZĄDZANIU PROJEKTAMI WYTWARZANIA OPROGRAMOWANIA?

**Streszczenie:** Artykuł dotyczy metodyk zwinnych (*agile methodologies*) zarządzania projektami wytwarzania oprogramowania, które stają się obecnie coraz bardziej popularne i coraz chętniej stosowane. Celem artykułu jest analiza pojęcia zwinności, metodyk zwinnych oraz próba znalezienia odpowiedzi na pytanie, kiedy warto takie metodyki stosować, poświęcając związany z ich zastosowaniem wysiłek organizacyjny, podejmując trud nabycia odpowiednich umiejętności i ponosząc niezbędne koszty. W artykule opisano metodyki zwinne jako specyficzne metodyki zarządzania projektami wytwarzania oprogramowania, ich rolę w kształtowaniu zwinnej infrastruktury informatycznej i zwinnej kultury organizacyjnej, a także zagadnienia dotyczące możliwości oceny poziomu zwinności metodyk oraz możliwości oceny celowości i korzyści stosowania.

**Słowa kluczowe:** zarządzanie projektami, wytwarzanie oprogramowania, metodyki zwinne, ocena metodyk projektowania, ocena projektów informatycznych.

DOI: 10.15611/ie.2014.1.13

## 1. Wstęp

Zarządzanie projektami wytwarzania oprogramowania należy do skomplikowanych przedsięwzięć. Obejmuje wiele działań i wiąże się z dużym ryzykiem niepowodzenia. Osiągnięcie sukcesu realizacji podjętej inwestycji w obszarze IT wymaga odpowiednio dostosowanej metodyki postępowania. Metodyki zwinne są przykładem metodyk zarządzania projektami wytwarzania oprogramowania, gdzie wykorzystywane są inne zasady i standardy postępowania niż w powszechnie stosowanych metodykach tradycyjnych, zawodnych w sytuacjach, z jakimi mamy do czynienia obecnie: dynamicznych zmian zarówno środowiska ekonomicznego, jak i niesprecyzowanych i zmiennych wymagań początkowych, dotyczących produktu finalnego jakim jest wytwarzane oprogramowanie. Motywem zastosowania podej-

ścia zwinnego w zarządzaniu projektem jest skuteczna reakcja na nowe wyzwania, pojawiające się w trakcie wytwarzania oprogramowania. Dzięki elastycznemu podejściu, jakie zakładają metodyki zwinne, łatwiejsze staje się osiągnięcie możliwie wysokiego poziomu wartości biznesowej uzyskanej poprzez wytworzone i wdrożone oprogramowanie. Użytkownicy wytwarzanego oprogramowania w momencie rozpoczęcia prac projektowych często nie są w stanie określić dokładnych wymagań związanych z oczekiwanym oprogramowaniem. W takiej sytuacji metodyki zwinne pozwalają na prowadzenie projektu i zakończenie go sukcesem, co nie byłoby możliwe, gdy użyte zostały metodyki tradycyjne, gdzie wszystkie wymagania dotyczące docelowego produktu muszą być określone już na początku prac projektowych. Jedną z metodyk zwinnych szerzej opisywanych w literaturze oraz wykorzystywanych w praktyce jest metodyka *Scrum*. Nasze rozważania i wnioski także w znacznym stopniu wynikają z tej metodyki.

Celem artykułu jest analiza pojęcia zwinności, metodyk zwinnych, a przede wszystkim znalezienie odpowiedzi na pytanie, kiedy warto je aplikować w projektach wytwarzania oprogramowania, pomimo że ich stosowanie nie jest proste i wymaga zaangażowania znacznych sił i środków. Metodyki zwinne prezentują idee względnie nowe, ale coraz bardziej popularne w zarządzaniu projektami wytwarzania oprogramowania. Pojęcie zwinności zostało przedstawione już w 2001 r. w tzw. Manifeście Zwinności i jest dość szeroko rozpowszechnione, ale nadal bywa niewłaściwie rozumiane. Zwinność w projektach wytwarzania oprogramowania bywa postrzegana jako brak zasad i wytycznych, traktowanych nawet jako podejmowanie działań nieuporządkowanych, chaotycznych. Prawdą jest natomiast to, że metodyki zwinne zakładają konieczność zachowania dyscypliny w pracach projektowych, a warunkiem ich poprawnego stosowania jest ich dogłębne zrozumienie. Niewłaściwa interpretacja i „swoboda" rozumienia proponowanych przez nie reguł zamiast korzyści może przynieść poważne straty. Opracowane zostały już miary pozwalające ocenić efektywność prowadzenia projektów wytwarzania oprogramowania z zastosowaniem różnych metodyk projektowych, także metodyk zwinnych. Należą do nich m.in. wskaźniki pracochłonności, wydajności, przewidywalności harmonogramu, stopnia realizacji wymagań, gęstości defektów po wytworzeniu oprogramowania. Zostaną one także przedstawione w niniejszym artykule.

Strukturze artykułu starano się nadać taki kształt, aby utworzyć konstrukcję logicznych wnioskowań, prowadzących do uzyskania choć częściowej odpowiedzi na pytanie postawione w jego tytule, a mianowicie, kiedy warto stosować metodyki zwinne (licząc się z koniecznością ponoszenia wysiłku i kosztów, aby zdobyć wystarczający poziom umiejętności ich wykorzystywania). Podjęto kolejno rozważania, począwszy od specyfiki metodyk zwinnych jako metodyk zarządzania projektowaniem oprogramowania, poprzez rolę metodyk zwinnych w kształtowaniu zwinnej infrastruktury informatycznej i zwinnej kultury organizacyjnej firmy, aż do oceny poziomu zwinności metodyki wybranej do zastosowania, aby ostatecznie

ocenić celowość i korzyści zastosowania zwinnego podejścia w projektowaniu oprogramowania.

W przypadkach zamieszczonych w artykule analiz i wyników wnioskowań opartych o już dość bogatą literaturę dotyczącą doświadczeń ze stosowania zwinnego podejścia w procesach wytwarzania oprogramowania wskazywano odpowiednie źródła bibliograficzne. Wykorzystano także własne doświadczenia z udziału w projektach realizowanych z zastosowaniem zwinnego podejścia. W takich przypadkach wszelkie wnioski i spostrzeżenia zostały wyraźnie wyodrębnione jako dorobek własny, nie zawierając odwołań literaturowych (poza odwołaniami do opracowań własnych).

## 2. Metodyki zwinne jako specyficzne metodyki zarządzania projektami wytwarzania oprogramowania

W niniejszym artykule przyjęto stosowanie terminu „metodyki zwinne", choć w literaturze można spotkać się z określeniami „metodologie zwinne" czy też „metody zwinne". Metodyka wydaje się określeniem najbardziej trafnym w języku polskim, nie zawężającym w żaden sposób rozważanej problematyki (zgodnie ze słownikiem Wydawnictwa Naukowego PWN, metodykę należy rozumieć jako zbiór zasad dotyczących sposobów wykonywania określonej pracy). Jako odpowiednik angielskiego *agile* przyjęto używać określenia „zwinny", choć i tu można spotkać się z innymi polskimi odpowiednikami, jak np. „adaptacyjne". Z dyskusją dotyczącą słuszności stosowania określonych terminów związanych z zarządzaniem projektami wytwarzania oprogramowania i wytwarzaniem oprogramowania można się zapoznać w pracy [Adamus 2013, s. 7-9].

Oprogramowanie jest specyficznym produktem, niematerialnym wytworem. Jedną z podstawowych różnic pomiędzy wytwarzaniem oprogramowania, a wytwarzaniem innych produktów, jest to, iż koszty wytwarzania tego produktu są największe w procesie projektowym, natomiast produkcja kolejnych „egzemplarzy" nie wymaga ponoszenia większych nakładów [Grabska, Klimczuk-Kochańska 2011, s. 12]. Już sam ten fakt wskazuje na szczególną wagę wyboru odpowiedniej metodyki wytwarzania oprogramowania. Inną specyficzną cechą wytwarzania oprogramowania jest większa niż w przypadku innych produktów trudność ustalenia wymagań, jakie powinien spełnić gotowy produkt. Jest to w znacznym stopniu związane z brakiem wiedzy użytkowników oprogramowania dotyczącej technologii informatycznej, tworzenia kodu programowego w różnych językach programowania i na różne urządzenia (np. mobilne) oraz ograniczeń, jakim podlega wytwarzanie oprogramowania. Zarówno w literaturze, jak i w praktyce wskazuje się, iż wymagania i oczekiwania użytkowników bywają niezwykle trudne lub wręcz niemożliwe do spełnienia (por. np. [Stepanek 2005, s. 10]). A co więcej, wymagania te są często zmieniane już w czasie realizacji projektu. W literaturze (i w praktyce)

można spotkać się nawet z poglądem, iż zanim nie poznają działającego programu, niezbyt dokładnie potrafią określić, jakie funkcje oprogramowania są im rzeczywiście potrzebne. Zarządzanie projektami wytwarzania oprogramowania utrudnia także fakt, że użytkownicy często sądzą, iż wprowadzenie zmian czy poprawek w oprogramowaniu nie powinno sprawiać poważniejszych trudności. W przypadku bardziej złożonych i różnorodnych zmian, a nawet prostszych poprawek pogląd taki jest całkowicie fałszywy. Stąd przydatna może być tylko taka metodyka zarządzania projektami wytwarzania oprogramowania, która pozwoli na wprowadzanie zmian żądanych przez użytkowników i pozwoli sprostać czasem nawet bardzo wysokim wymaganiom użytkowników.

Rozwój metodyk wytwarzania oprogramowania jest ściśle związany z rozwojem technologii informatycznych, w tym sprzętu komputerowego i języków oprogramowania, a także zwiększającymi się coraz intensywniej wymaganiami użytkowników. Projekty informatyczne są prowadzone na coraz większą skalę i coraz wyraźniej uwidaczniają problemy związane z wytwarzaniem oprogramowania. Projektowanie oprogramowania zaczyna być traktowane jak projektowanie innych, choć specyficznych produktów i przekształca się w dziedzinę działalności inżynierskiej. Znajduje to odzwierciedlenie w terminologii, w tym powszechnie używanego od końca lat 60. XX wieku terminu „inżynieria oprogramowania". Termin ten sugeruje, że projektowanie oprogramowania nie różni się od projektowania wytwarzania każdego innego produktu. Nasuwa się wniosek, że takie metodyki, jak stosowane w przypadku projektowania wytwarzania tych produktów, powinny być odpowiednie także w przypadku oprogramowania. Dla zwiększenia przewidywalności i stopnia kontroli realizacji projektów w sferze informatyki zaczęto stosować sformalizowane, restrykcyjne czy „sztywne" metodyki klasyczne, ukierunkowane na kontrolę zarządzania projektami wytwarzania oprogramowania. Stosowane także i obecnie kładą nacisk na kontrolę wszystkich działań projektowych, w tym kontrolę zmian i ryzyka, przy silnie sformalizowanej strukturze zadaniowej oraz organizacyjnej projektów i strukturze zadaniowej złożonej z etapów, zazwyczaj z założeniem ich kaskadowej realizacji [Miłosz i in. 2011, s. 9, 11].

Wprowadzenie sformalizowanego postępowania metodyk klasycznych dało niewątpliwie wiele korzyści, doprowadzając do ograniczenia chaosu w zarządzaniu projektami wytwarzania oprogramowania. Jednocześnie stało się łatwo zauważalne, że nadmierna kontrola oraz wymagania tworzenia drobiazgowej, obszernej dokumentacji zamiast pomagać, stają się hamulcem w realizacji projektów informatycznych. Obiektem krytyki stała się biurokracja. Oczywiście, szczegółowe opracowanie precyzyjnego planu w oparciu o początkowe wymagania dotyczące oprogramowania i stosowanie go bez zmian podczas realizacji projektu nie jest właściwym sposobem postępowania i może przynieść wiele ujemnych skutków, przede wszystkim z uwagi na zmiany wymagań co do oprogramowania zgłaszane przez użytkowników, postęp w dziedzinie informatyki i bardzo szybko rozwijające się

technologie informatyczne. Aby rozwiązać pojawiające się problemy, w realizacji projektów wytwarzania oprogramowania za pomocą metodyk klasycznych coraz więcej uwagi zaczęto poświęcać zastosowaniom iteracji. Iteracje stosowano wprawdzie już od początku wytwarzania oprogramowania, ale do lat 90. XX wieku dominujące było podejście kaskadowe, związane z metodyką kaskadową [Larman, Basili 2003, s. 6]. W literaturze można spotkać się z utożsamianiem iteracyjności ze zwinnością, należy zaznaczyć jednak, że iteracyjność nie jest równoznaczna pojęciu zwinności. Wiele metodyk uwzględnia iteracje, ale nie można ich uważać za metodyki zwinne, ale za tradycyjne. Iteracyjność jest warunkowana wymogami dokumentacyjnymi oraz kontrolnymi. Wiele metodyk, zaliczanych do grupy zwinnych, powstało w latach 90. XX wieku wraz z rosnącą popularnością iteracji w projektowaniu, a także sprzeciwu wobec biurokratyzacji, związanej przede wszystkim ze sporządzaniem olbrzymiej ilości szczegółowej dokumentacji. Wówczas opracowano pierwsze metodyki o charakterze zwinnych. Można do nich zaliczyć Programowanie Ekstremalne oraz Scrum, nadal stosowane i rozwijane obecnie.

Za datę wyznaczającą początek rozwoju „ruchu agile" uznawany jest rok 2001. Został wówczas opracowany Manifest Zwinności (zwany też Manifestem Zwinnego Wytwarzania Oprogramowania) – dokument, w którym zawarto postulaty zwinności, będące kryteriami umożliwiającymi ocenę, czy daną metodykę można uznać za zwinną, i formułującym wytyczne tworzenia metodyk zwinnych. W Manifeście Zwinności przedstawiono cztery postulaty zwinności dla wytwarzania oprogramowania. Stały się one podstawą metodyk będących alternatywą dla tradycyjnych metodyk, pozwalając na szybsze i zakończone sukcesem wytwarzanie oprogramowania [Calo i in. 2010, s. 68]. Celem Manifestu nie było opracowanie metodyki, ale wskazanie cech, jakimi powinny odznaczać się metodyki zwinne [Cockburn 2008, s. 383]. Zacytujmy [Manifest Zwinnego Wytwarzania Oprogramowania 2001]: „Wytwarzając oprogramowanie i pomagając innym w tym zakresie, odkrywamy lepsze sposoby wykonywania tej pracy. W wyniku tych doświadczeń przedkładamy: Ludzi i interakcje ponad procesy i narzędzia. Działające oprogramowanie ponad obszerną dokumentację. Współpracę z klientem ponad formalne ustalenia. Reagowanie na zmiany ponad podążanie za planem. Doceniamy to, co wymieniono po prawej stronie, jednak bardziej cenimy to, co po lewej". Analizując manifest, należy zwrócić uwagę, o czym świadczy ostatnie jego zdanie, na fakt, że nie odrzuca się w nim dotychczasowych „elementów" obecnych w projektowaniu oprogramowania, wymienionych po prawej stronie postulatów, tj. potrzeby procesów i narzędzi, tworzenia dokumentacji, formalnych ustaleń, planowania i realizacji planów. Dążeniem manifestu jest podkreślenie ważności tego dla sukcesu projektu, co umieszczono po lewej stronie postulatów i co może być ważniejsze niż to, co uwidoczniono po prawej stronie i jest dobrze znane z tradycyjnych metodyk. Manifest podkreśla kolejno ważność: roli relacji interpersonalnych w projekcie (por. [Highsmith 2005, s. 32-33]), czego nie zastąpią procesy i narzędzia; powstawania

i systematycznego prezentowania sprawnego, działającego oprogramowania (aktywnych funkcji, prototypów, sprawnego kodu), czego nie zastąpi nawet najobszerniejsza dokumentacja; utrzymywania dobrych relacji z klientem, aktywnej współpracy użytkownika z zespołem projektowym, współuczestnictwa użytkownika w projekcie, czego nie zastąpią formalne ustalenia: klient – zespół projektowy; szybkich i systematycznych reakcji na zmiany, czego nie zastąpi kontrolowanie wykonywania planu i uparte, prowadzone z bezwzględną determinacją dostosowywanie działań do planu. Postulaty przedstawione w Manifeście Zwinności wyraźnie wskazują na potrzebę metodyk, które dawałyby więcej swobody w działaniach projektowych, stosowanie mniej rygorystycznego i sformalizowanego podejścia, a większej elastyczności w dostosowywaniu działań do uwarunkowań, w jakich realizowany jest projekt, i podlegających zmianom sytuacji podczas prowadzenia projektów.

Opracowano także zbiór bardziej szczegółowych reguł i zasad, wynikających z postulatów zawartych w Manifeście Zwinności. Ich przytoczenie i nawet krótkie omówienie wymagałoby przekroczenia dopuszczalnej objętości tego artykułu. Zostały one dość dokładnie przedstawione w pracy [Adamus 2013, s. 17-21].

## 3. Rola metodyk zwinnych w kształtowaniu zwinnej infrastruktury informatycznej i zwinnej kultury organizacyjnej firmy

W przypadku projektów zarządzanych za pomocą tradycyjnych metodyk, wymagania klienta – przyszłego użytkownika systemu zbierane są od niego podczas rozpoczęcia projektu i przyjmowane jako niezmienne, wyznaczające zakres realizowanych prac. W projektach wykorzystujących metodyki zwinne zakłada się, iż zebrane wymagania będą poddawane ciągłej analizie i weryfikacji w celu ukształtowania elastycznej i zwinnej infrastruktury informatycznej ukierunkowanej na osiąganie coraz wyższego wskaźnika zwrotu z inwestycji w oprogramowanie, przekładającego się na usprawnianie i doskonalenie z uwagi na względy ekonomiczne, biznesowe i osiąganie wyższej wartości biznesowej [Cobb 2012, s. 13].

Stosowanie zwinnych metodyk zarządzania projektami wytwarzania oprogramowania kształtuje nie tylko zwinną infrastrukturę informatyczną, ale ma także swój udział w tworzeniu zwinnej kultury organizacyjnej firmy, co jest związane przede wszystkim z faktem znacznego oddziaływania technologii informatycznych na firmy i sposoby ich funkcjonowania.

Skuteczne wykorzystywanie metodyk zwinnych w zarządzaniu projektami wytwarzania oprogramowania wymaga tworzenia zwinnej infrastruktury informatycznej oraz kultury organizacyjnej. [Adamus 2013, s. 41-47; Sahota 2012, s. 4; Arell i in. 2012, s. 1]. Stają się dostosowane do zwinności niejako „pod naciskiem" metodyk zwinnych stosowanych w zarządzaniu projektami wytwarzania oprogramo-

wania. Możemy określić wówczas, że są zwinne i tworzy się środowisko sprzyjające zwinnemu podejściu w zarządzaniu.

M. Sahota [2012, s. 7-11] podjął próbę przypisania zasad zawartych w Manifeście Zwinności do typów kultur organizacyjnych firmy, takich jak kultura współpracy (*collaboration culture*), gdzie ludzie i ich interakcje wynikają z zaufania i stanowią kluczową wartość, kultura kontroli (*control culture*), z priorytetem utrzymywania kontroli realizowanych procesów, kultura oparta na umiejętnościach (*competence culture*) uznająca za najważniejszy czynnik sukcesu kunszt i profesjonalizm pracowników oraz kultura rozwoju (*Cultivation culture*) z założeniem, że wspieranie rozwoju jest podstawowym warunkiem powodzenia. W odniesieniu do kultury współpracy można wskazać takie zasady Manifestu Zwinności, jak: „współpraca między ludźmi biznesu i programistami musi odbywać się codziennie w trakcie trwania projektu", „najlepsze architektury, wymagania i projekty powstają w samoorganizujących się zespołach", „najwydajniejszym i najskuteczniejszym sposobem przekazywania informacji do (zespołu) i w ramach zespołu jest rozmowa twarzą w twarz". Do kultury kontroli trudno odnieść jakikolwiek z postulatów, zasadę, czy regułę zwinności. Kulturę opartą na umiejętnościach można powiązać z zasadą „poprzez ciągłe skupienie na technicznej doskonałości i dobremu zaprojektowaniu oprogramowania zwiększa zwinność". Kulturze rozwoju można dopasować zasady: „bądź otwarty na zmieniające się wymagania nawet na zaawansowanym etapie projektu", „zwinne procesy wykorzystują zmiany dla uzyskania przewagi konkurencyjnej klienta", „twórz projekty wokół zmotywowanych osób; daj im środowisko i wsparcie, którego potrzebują i ufaj im, że wykonają swoją pracę", „w regularnych odstępach czasu zespół zastanawia się jak poprawić swoją efektywność, dostosowuje lub zmienia swoje zachowanie". Wyniki analizy wskazują, że kultura organizacyjna najbardziej sprzyjająca zwinności to kultura współpracy oraz kultura rozwoju.

Przedstawiciele Agile Alliance przeprowadzili analizę cech kultury organizacyjnej, które powinny być rozwijane, aby sprzyjało to skuteczności stosowania metodyk zwinnych. Wśród tych cech znalazły się takie jak: rozumienie systemu jako całości, zaadaptowanie „katalitycznego" stylu zarządzania, uczenie się na podstawie doświadczeń, wspieranie otwartej komunikacji, zarządzanie poprzez kreowanie długoterminowej wartości biznesowej i adaptacji, szukanie przez pracowników biegłości w odpowiednich dla nich dziedzinach [Arell i in. 2012, s. 2-9]; por. szczegółowy opis w: [Adamus 2013, s. 45-47].

Aby skuteczne było wykorzystywanie metodyk zwinnych w zarządzaniu projektami wytwarzania oprogramowania, niezbędne są starania, możliwe w ramach działań obejmowanych metodykami zwinnymi, kształtowania kultury organizacyjnej sprzyjającej użyciu tych metodyk, czyli kultury charakteryzującej się zwinnością.

## 4. Metody oceny poziomu zwinności stosowanej metodyki zarządzania projektem wytwarzania oprogramowania

A. Cockburn [2008, s. 286] stwierdza, że żadna z metodyk „sama z siebie nie jest zwinna", natomiast uzyskanie odpowiedniego stopnia zwinności wynika ze sposobu jej implementacji, który powinien zostać odpowiednio dobrany w zależności od takich parametrów, jak kultura organizacyjna, stopień informatyzacji czy zakładany stopień zwinności procesów implementacyjnych. Pomimo istniejących trudności implementacyjnych spotkać się można z wieloma propozycjami parametrów służących do oceny zwinności stosowanej metodyki. Jedną z często wymienianych jest zaproponowana w artykule „A Quantitative Framework for the Evaluation of Agile Methodologies" [Calo i in. 2010, s. 70]. Zwinność oceniana jest jako zgodność z postulatami Manifestu Zwinności i jest mierzona w sposób ilościowy. Aby ocenić stopień zwinności stosowania metodyki, obliczane są punkty dotyczące każdego z czterech postulatów Manifestu Zwinności, na podstawie dwóch miar. Pierwsza – miara m(Pi.1) jest oceną – liczbą punktów odnoszących się do zwinnej zawartości postulatu, druga – miara m(Pi.2) jest oceną – liczbą punktów odnoszących się do tradycyjnej (nie odpowiadającej zwinnej) zawartości postulatu. Ostateczna ocena zwinności stosowania metodyki obliczana jest jako suma wielkości M(Pi): M(Pi) = m(Pi.1) + m(Pi.2), gdzie: M(Pi) – ocena zgodności z *i*-tym postulatem Manifestu Zwinności, m(pi.1) – ocena wartości zwinnych, m(pi.2) – ocena wartości tradycyjnych. Pomocne może być wyznaczenie wartości granicznych, stanowiących punkty odniesienia, wskazujące, czy możemy uznać, że metodyka zapewnia nam wystarczający poziom zwinności.

Zwinne zarządzanie projektami wytwarzania oprogramowania nie jest jednoznaczne związane z koniecznością zastosowania jednej z metodyk traktowanych jako metodyki zwinne, takie jak: Scrum, Programowanie Ekstremalne, Crystal, Feature Driven Development, Agile Modeling (por. metodyki zwinne w [Adamus 2013, s. 27-35]). Wśród tych metodyk niektóre ukierunkowane są przede wszystkim na wytwarzanie oprogramowania (jak Programowanie Ekstremalne oraz Agile Modeling), inne głównie na wspomaganie zarządzania wytwarzaniem oprogramowania (jak Scrum, Crystal). Istnieją też takie, które poświęcają uwagę zarówno zarządzaniu projektami, jak i wytwarzaniu oprogramowania (jak Feature Driven Development) [Abrahamsson i in. 2002, s. 95]. Firmy mogą tworzyć i aplikować własne metodyki lub wykorzystywać już istniejące do realizowanych projektów i prowadzić je w sposób zwinny. Jednakże niezbędne jest, aby użyta metodyka posiadała cechy właściwe dla metodyk zwinnych i zwinnego zarządzania projektami. Do cech tych zaliczane są [Tomasini, Kearns 2012, s. 8-12]: iteracyjny cykl wytwarzania oprogramowania, empiryczna kontrola procesu, zarządzanie przez „wyciąganie", a nie „wypychanie", zastosowanie „odchudzonego" zarządzania. Podczas iteracyjnego cyklu życia projektu wytwarzanie oprogramowania następuje w

ramach kolejnych iteracji. W każdej z nich wytwarzany jest kolejny przyrost produktu, czyli jego funkcjonalna część zgodnie z kaskadowym cyklem życia wytwarzania oprogramowania, z fazami na tyle krótkimi, aby mieściły się w czasie trwania każdej iteracji [*Iterative and incremental development* 2013]. W metodykach zwinnych iteracja jest podstawową jednostką planowania, wykonania i kontroli postępów projektu [Sacha 2010, s. 340]. Zalecany czas jej trwania może różnić się zależnie od metodyki oraz potrzeb konkretnego projektu. W przypadku podejścia zwinnego do wytwarzania oprogramowania znaczną uwagę zwraca się na dużą częstotliwość przekazywania nowych funkcjonalności klientowi, będącą kluczem do sukcesu.

Następną z wymienionych cech jest empiryczna kontrola procesu, zakładająca, że realizacja projektów odbywa się z zachowaniem przejrzystości, inspekcji postępu prac i możliwości adaptacyjnych [Schwaber, Sutherland 2011, s. 4]. Przejrzystość oznacza możliwość monitorowania oraz czytelność poszczególnych aspektów oraz etapów procesu dla osób odpowiedzialnych za osiągane rezultaty. Aby uzyskać satysfakcjonujący poziom tej cechy, konieczne jest przygotowanie odpowiedniego, zgodnego z powszechnie znanymi standardami, opisu wszystkich elementów związanych z procesem. Zastosowanie zasady przejrzystości umożliwia ujednolicone podejście wszystkich zainteresowanych osób do różnorodnych aspektów procesu. Inspekcja jest rozumiana jako przegląd postępów realizacji projektu. Jej celem jest wykrywanie niepożądanych odstępstw od oczekiwanego przebiegu realizacji prac w trakcie projektu wytwarzania oprogramowania. Nie oznacza to ścisłej kontroli wykonywania planu, który został przyjęty na początku projektu, lecz zakłada możliwość występowania zakłóceń w procesie tworzenia produktu i takie ukierunkowanie działań, które powodowałoby generowanie jak największej wartości biznesowej dla odbiorcy oprogramowania. Inspekcje powinny być regularne, jednak nie na tyle częste, aby zakłócać prace. Adaptacja określa zdolność do wprowadzania korekt do procesu w przypadku, gdy w ramach inspekcji zostają wykryte nieprawidłowości. Empiryczna kontrola procesu przewiduje ewentualność występowania zmian i ich wykrywanie, a jednocześnie zakłada możliwość adaptacji procesu wytwarzania oprogramowania do tych zmian [Tomasini, Kearns 2012, s. 8]. Podejścia „wyciągające" i „wypychające" odnoszą się do procesu zbierania wymagań od klienta. Cechą metodyk zwinnych jest tzw. podejście „wyciągające" w pozyskiwaniu informacji o wymaganiach klienta [Cobb 2012, s. 27-29]. W metodykach tradycyjnych stosowane jest podejście „wypychające", co oznacza, że na początku projektu zbierane są wszystkie wymagania klienta dotyczące finalnego produktu, a następnie, w trakcie trwania projektu, są one sekwencyjnie „wypychane". Takie podejście jest uzasadnione, gdy istnieje pewność, że wymagania nie zmienią się w trakcie realizacji projektu. Jednakże zazwyczaj warunek ten nie jest spełniony. Dlatego w metodykach zwinnych proponowane jest podejście „wyciągające", oznacza to że w danym przedziale czasu pozyskiwana jest jedynie niezbędna ilość wymagań, pozwalająca na kontynuację prac. Wymogiem zwinnej me-

todyki jest tak zwane „odchudzone" zarządzanie (*lean management*). Podejście *lean management*, wywodzące się z koncepcji doskonalenia organizacji procesów produkcyjnych, którego celem jest eliminacja marnotrawstwa, rozumianego jako zużywanie zasobów na inny cel niż tworzenie wartości dla klienta, ma swoje również w projektach wytwarzania oprogramowania. W przypadku projektów wytwarzania oprogramowania takim marnotrawstwem są czynności nieprzynoszące efektu wartości dodanej, czyli nieprzyczyniające się do tworzenia funkcjonalności oprogramowania, jakiej oczekuje użytkownik. Nie dotyczy to procedur, które są pośrednio niezbędne do spełnienia wymagań użytkownika, ale takich, które wymagają zużycia określonych zasobów, a nie tworzą z punktu widzenia użytkownika oprogramowania żadnej wartości. C. Cobb [2012, s. 22-23] podał warunki w postaci „ogólnych zasad", służące eliminacji niepotrzebnych działań jako: „tyle ile potrzeba" (*just enough*), „dokładnie na czas" (*just-in-time*), „odpowiedniej wielkości" (*right-sized*). Oznacza to konieczność przygotowywania wystarczającej ilości dokumentacji, ale ani za dużo, ani za mało, lub wykonania odpowiednich czynności w odpowiednim czasie, tj. dopiero wówczas, gdy trzeba je wykonać, lecz ani nie wcześniej, ani nie później.

## 5. Ocena celowości i korzyści stosowania metodyk zwinnych

Jeżeli stosowane są tradycyjne metodyki zarządzania projektami wytwarzania oprogramowania, to osiągnięcie jak największego sukcesu jest ograniczone poruszaniem się w ramach obszaru, który można wyobrazić sobie jako trójkąt o wierzchołkach „zakres", „koszt" i „czas", ograniczających możliwości działania. Zmiana jednego „z wierzchołków" zmusza do korekty pozostałych. Aby osiągnąć sukces – produkt o akceptowalnym poziomie jakości, trzeba odpowiednio sterować „zakresem, „kosztem" i „czasem". Uwaga musi być skupiona na zarządzaniu zakresem, kosztami i czasem (harmonogramem), by spełnić ograniczenia, przy zatwierdzonych wymaganiach ustalonych przy podjęciu projektu [Cobb 2012, s. 11]. W tym ujęciu zostaje pominięty problem zapewniania wartości biznesowej. Wprowadza to stosowanie metodyk zwinnych, gdzie nacisk na zapewnianie wartości biznesowej staje się kluczowym czynnikiem projektów zarządzanych zwinnie. Jim Highsmith, jeden z twórców Manifestu Zwinności, przedstawił trójkąt zarządzania projektami zwinnymi, w którym ograniczenia zakresu, kosztu i czasu stanowią jedynie jeden z wierzchołków. Pozostałe dwa wierzchołki obrazują wartość (produkt do wydania) i jakość (produkt niezawodny, adaptowalny) [Cobb 2012, s.13]. W tym ujęciu wyraźnie sygnalizowane jest dążenie do bardziej elastycznego zarządzania projektami. Zakłada się aktywne zaangażowanie użytkownika, wiążące się z możliwością wprowadzania zmian w celu maksymalizowania wartości biznesowej wynikającej z realizowania projektu.

Realizacja projektów wytwarzania oprogramowania wiąże się ze znacznym ryzykiem niepowodzenia, którego przykładowymi źródłami mogą być: niekompletne wymagania oraz zmiana zakresu projektu, niezgodny z oczekiwanymi sposób działania wykorzystywanych narzędzi oraz przewidywanych do zastosowania zewnętrznych komponentów, niewystarczające umiejętności członków zespołu projektowego, niespodziewanie ujawniające się wady wytworzonego oprogramowania, wymagające poniesienia nieprzewidywanych środków, utrata kluczowych członków zespołu projektowego [Stepanek 2005, s. 46]. W przypadku stosowania metodyk tradycyjnych prowadzi się planowanie zarządzanie ryzykiem jako jeden z elementów planowania projektu niezbędnych jeszcze przed jego rozpoczęciem. Tworzona jest odpowiednio obszerna dokumentacja [Sacha 2010, s. 61]. Wszystkie działania mające na celu zmniejszenie możliwego ryzyka podejmowane już przed rozpoczęciem projektu wymagają ponoszenia odpowiednio wysokich kosztów, a niekoniecznie muszą przynieść korzyści [Cobb 2012, s. 159]. Dodatkowo, w przypadku metodyk tradycyjnych, często już przed rozpoczęciem projektu dąży się do sparametryzowania wszystkich ryzyk. Jest to trudne do zdiagnozowania, podobnie jak sporządzenie drobiazgowego i wyczerpującego opisu finalnego produktu czy planu realizacji wszystkich zadań, gdy projekt nie został jeszcze w pełni zdefiniowany. W tego typu projektach inwestycyjnych wysoce prawdopodobne jest, że definicja oczekiwań realizowana będzie krokowo, immanentne dla projektów wytwarzania oprogramowania.

Metodyki zwinne w odróżnieniu od tradycyjnych traktują ryzyko jako element niejako naturalny. Takie podejście zakłada, że kolejne ryzyka będą identyfikowane i definiowane w toku realizacji projektu. Przyjęcie takiej optyki działania zakłada wprowadzanie systematycznych zmian w zakresie prowadzonego projektu. Prace prowadzone w sposób iteracyjny pozwalają dostrzec i zmniejszyć każde pojawiające się ryzyko zagrażające projektowi. Na początku każdej iteracji przeprowadzana jest weryfikacja istniejących wymagań dotyczących oprogramowania i wybór tych które są najistotniejsze oraz będą realizowane w ramach tej iteracji. Wprowadzanie zmian zakresu projektu nie zawsze determinuje ponoszenie dodatkowych kosztów, a nieuwzględnione dotychczas wymagania użytkowników, które uznane zostaną za istotne, mogą zostać zamieszczone w toku „rutynowego" postępowania. Stosowanie iteracji pozwala także na zmniejszenie ryzyka związanego z niezgodnym z oczekiwaniami sposobem działania wykorzystywanych narzędzi oraz przewidywanych do zastosowania zewnętrznych komponentów oprogramowania. Narzędzia i komponenty mogą bowiem zostać przetestowane w ramach początkowych iteracji, a to przyczyni się do wyeliminowania związanych z nimi problemów. W ten sposób, dzięki działaniu w ramach iteracji, większość niepożądanych ryzyk może zostać odpowiednio wcześniej wykrytych i anihilowanych. Przeprowadzanie inspekcji oraz nacisk na ciągłe testowanie realizowanych prac, w tym powstającego kodu, pozwalają skutecznie unikać niepożądanych sytuacji. Jeżeli zostanie zastosowane

programowanie ekstremalne, zakładające programowanie w parach, refaktoryzację kodu i współwłasność kodu, znika ryzyko wytworzenia niewłaściwego, obarczonego błędami kodu programowego (por. [Adamus 2013, s. 27-28, 38]). Metodyki zwinne nie zawierają opracowanego oddzielnego etapu czy też zestawu działań związanego z zarządzaniem ryzykiem i przeciwdziałania zagrożeniom, jakie może nieść, a tym bardziej ponoszenia związanych z tym dodatkowych kosztów, ale są przygotowane do funkcjonowania w ramach ryzyka, które jest traktowane jako naturalne dla projektów wytwarzania oprogramowania [Highsmith 2005, s. 170].

Metodyki zwinne ułatwiają realizację projektów w sytuacji, gdy oprogramowanie ma być wykonane przez firmę zewnętrzną i zawierany jest kontrakt między klientem zlecającym wykonanie oprogramowania a firmą. W procesie produkcji oprogramowania restrykcyjne podejście do kontraktu może doprowadzić do porażki projektu. W kontrakcie zakłada się bowiem, że wymagania klienta ustalone przed rozpoczęciem wytwarzania są wystarczającą podstawą do określenia finalnego produktu oraz dokładnego planu jego wykonania. W przypadku projektów wytwarzania oprogramowania jest to zwykle niezgodne ze stanem faktycznym. Jeżeli klient nie uczestniczy w tworzeniu oprogramowania, a nawet niekiedy dopiero „ogląda" gotowy produkt, nie jest zwykle z niego zadowolony i źle go ocenia [Arbogast i in. 2013, s. 26]. Przypomnijmy, inaczej wygląda sytuacja jeśli zastosowane zostaną metodyki zwinne. Jeden z postulatów Manifestu Zwinności wyraźnie określa, że współpracę z klientem należy przedkładać ponad formalne ustalenia. Silnie akcentowany jest podgląd, że współpraca jest bardziej wartościowa niż formalne umowy. Realizacja projektów zarządzanych za pomocą metodyk zwinnych, wymusza stosowanie postulatów (reguły i zasady) wyrażonych w Manifeście Zwinności. Istotnym czynnikiem jest zaufanie między klientem a wytwórcą oprogramowania. Zarządzając projektem za pomocą metodyk zwinnych, nie sporządza się drobiazgowej specyfikacji wymagań przed rozpoczęciem projektu wytwarzania oprogramowania. Takie działanie traktuje się jako czynnik utrudniający wycenę projektu i nadmierne uściślenie warunków kontraktu, np. dotyczących sposobów płatności. W sytuacji stosowania metodyk zwinnych preferowane jest podejście polegające na tym, że klient płaci firmie wytwarzającej oprogramowanie za rzeczywisty czas spędzony nad produkcją oprogramowania oraz pokrywa koszty pojawiające się podczas produkcji.

Metodyki tradycyjne zakładają, że przed rozpoczęciem projektu znany będzie ostateczny plan realizacji finalnego produktu oraz podejmowanych prac wytwórczych. Jeżeli przyjmiemy takie założenie, to możliwe jest stworzenie budżetu projektu. Jednakże opracowane budżety mogą okazać się niepoprawne, ponieważ projekty wytwarzania oprogramowania cechuje duży poziom niepewności i prawidłowe oszacowanie kosztów nie jest w wielu przypadkach możliwe. W projektach zwinnych, w których przyjmuje się niemal jako pewnik zmienne wymagania dotyczące produktu, prowadzona jest estymacja progresywna. Proces ten następuje stopniowo i odzwierciedla wzrost wiedzy o produkowanym oprogramowaniu

[Highsmith 2005, s. 166]. Estymacja projektów zwinnych wskazuje na jeszcze jedną przewagę nad postępowaniem w przypadku projektów prowadzonych tradycyjnie. Mianowicie w projektach zwinnych szacowane są elementy funkcjonalności, natomiast w projektach tradycyjnych estymacji podlegają zadania. Oszacowanie elementów funkcjonalności jest bardziej zrozumiała dla odbiorcy oprogramowania niż technicznych zadań, które muszą być wykonane przez informatyków, którzy są odpowiedzialni za procesy wytwarzania oprogramowania [Highsmith 2005, s. 45].

Po zakończeniu projektu realizowanego za pomocą metodyki zwinnej może być przeprowadzona ocena ilościowa wykonania projektu informująca o tym, czy celowe i korzystne było zastosowanie metodyki. Jedną z możliwości oceny jest zastosowanie miar proponowanych przez Software Engineering Institute. Należą do nich [Kasunic 2008, s. 7 i n.]:

- pracochłonność (*project effort*) – określająca czas (liczbę godzin), jaki pracownicy poświęcili na realizację projektu;
- wydajność (*productivity*) – liczona jako iloraz rozmiaru projektu (mierzony liczbą linii kodu, który powstał w ramach projektu) i wcześnie zdefiniowanej pracochłonności;
- czas trwania projektu (*project duration*) – mierzony w dniach od rozpoczęcia do ukończenia, wyłączając te dni, w których nie realizowano zadań związanych z projektem,
- przewidywalność harmonogramu (*schedule predictability*) – określająca stopień realizacji zaplanowanych etapów projektu, określająca jak duża jest rozbieżność między rzeczywistym czasem trwania projektu a oszacowanym czasem trwania, liczonym jako iloraz różnicy między rzeczywistym czasem trwania projektu (liczbą dni, w których trwały prace nad projektem) a oszacowanym czasem trwania projektu, i oszacowanym czasem trwania projektu;
- wskaźnik realizacji wymagań (*requirements completion ration*) – wskazujący stopień realizacji wymagań użytkownika oprogramowania, obliczany jak iloraz liczby wymagań, które zostały zrealizowane w ramach projektu, i liczby wymagań zdefiniowanych na początku projektu łącznie z tymi, które pojawiły się w trakcie, w efekcie negocjacji z przyszłym użytkownikiem;
- gęstość defektów po wydaniu oprogramowania (*post release defect density*) – oznaczająca liczbę błędów wykrytych w oprogramowaniu po wdrożeniu go do użytkowania z uwzględnieniem rozmiaru projektu, liczona jako iloraz sumy defektów (jeżeli te same powtarzają się wielokrotnie, to liczone są tylko raz) znalezionych przez użytkowników w ciągu pierwszych sześciu miesięcy używania produktu i rozmiaru projektu (liczby linii kodu, który powstał w ramach projektu). Aby dokonać ostatecznej oceny wykonania projektu, proponuje się wyznaczenie wartości granicznych (minimalnych lub/i maksymalnych), które nie powinny zostać przekroczone dla poszczególnych miar, zgodnie z wymaganiami osób zaangażowanych w projekt.

## 6. Zakończenie

Rozważania przedstawione w artykule wskazują, że można dokonać oceny warunków, w jakich warto wykorzystywać metodyki zwinne w zarządzaniu projektami wytwarzania oprogramowania. Z przeprowadzonej analizy wynika jednoznacznie, że trafne będzie użycie tych metodyk w przypadku projektu, dla którego dokładne wymagania nie są znane i możliwe do precyzyjnego zdefiniowania przed jego rozpoczęciem. Metodyki zwinne przewidują sytuację, że użytkownik nie jest w stanie określić wszystkich wymagań, które są „doprecyzowywane" już w czasie trwania projektu i przedstawia postulaty, zasady i reguły postępowania, gdy istnieje taka sytuacja. Przyjmowanie pewnych wymagań już na początku podjęcia projektu może być czasem nawet niepożądane z uwagi na przewidywane zmiany technologii czy wymagań użytkownika. Metodyki zwinne są dostosowane do takich sytuacji przewidywanej zmienności i zakładają jej zachodzenie w trakcie realizacji projektu nie jako zakłócenie, ale naturalną cechę w realizacji projektów wytwarzania oprogramowania, i proponują odpowiednie sposoby postępowania w warunkach zmienności. Za pomocą różnorodnych miar, m.in. przedstawionych w artykule, można dokonać ostatecznej oceny wykonania projektu za pomocą metodyki zwinnej. Jak wskazano w artykule, projektowanie zwinne nie musi odbywać się za pomocą jednej z „klasycznych" metodyk zwinnych, a może znaleźć zastosowanie własna metodyka, której poziom zwinności zostanie oceniony jako wystarczający. Omówione zostały również sposoby przeprowadzenia oceny poziomu zwinności zastosowanej metodyki. Bardzo ważna jest prawidłowa interpretacja postulatów, zasad i reguł zwinności, bo tylko wówczas możliwe jest zgodne z nimi postępowanie podczas realizacji projektu i zakończenie go sukcesem. Metodyki zwinne są warte zaimplementowania także tylko wówczas, gdy zachodzi możliwość kształtowania, umożliwiającego wykorzystanie ich zalet i wynikających z ich zastosowania korzyści, „zwinnego środowiska", tj. zwinnej infrastruktury informatycznej i zwinnej kultury organizacyjnej, na których zaistnienie same metodyki mogą mieć wpływ, jeżeli zostanie to im umożliwione.

Wyniki przeprowadzonych przez nas analiz literaturowych i wnioski z badań własnych uzyskanych na podstawie doświadczeń z udziału w projektach, w których były stosowane postulowane reguły zwinnego wytwarzania oprogramowania, dość wyraźnie wskazują na kształtowanie się nowego podejścia do zarządzania projektami tworzenia oprogramowania i pozwalają zaryzykować stwierdzenie o ich znaczącym wpływie na rozwój inżynierii oprogramowania jako nauki.

## Literatura

Abrahamsson P., Salo O., Ronkainen J., Warsta J., 2002, *Agile software development methods. Review and analysis*, VTT Electronics, University of Oulu.

Adamus A., 2013, *Zastosowanie metodyk zwinnych w produkcji oprogramowania przez firmy „software'owe"*, praca magisterska, Wydział Nauk Ekonomicznych, Uniwersytet Warszawski, Warszawa.

Arbogast T., Larman C., Vodde B., *Agile contracts primer*, Online, http://www.agilecontracts.org (10.07.2013).

Arell R., Coldewey J., Gat I., Hesselberg J., 2012, *Characteristics of Agile Organizations*, Agile Alliance.

Calo K., Estevez E., Fillottrani P., 2010, *A Quantitative Framework for the Evaluation of Agile Methodologies*, JCS&T, vol. 10.

Cobb C., 2012, *Zrozumieć Agile Project Management. Równowaga kontroli i elastyczności*, APN Promise, Warszawa.

Cockburn A., 2008, *Agile Software Development. Gra zespołowa*, Wydawnictwo Helion, Gliwice.

Grabska A., Klimczuk-Kochańska M., 2011, *Analiza gospodarczych obszarów wzrostu i innowacji województwa podlaskiego. Sektor produkcji oprogramowania komputerowego*, Wojewódzki Urząd Pracy w Białymstoku, Białystok.

Highsmith J., 2005, *APM*: *Agile Project Management. Jak tworzyć innowacyjne produkty*, MIKOM, Warszawa.

*Iterative and incremental development*, Mera 2013, http://www.meranetworks.com/services/processes/iterative (21.05.2013).

Kasunic M., 2008, *A Data Specification for Software Project Performance Measures*: *Results of a Collaboration on Performance Measurement*, Report prepared for the SEI Administrative Agent, July.

Larman C., Basili V., 2003, *Iterative and Incremental Development*, A Brief History, June.

Manifest Zwinnego Wytwarzania Oprogramowania 2001, http://agilemanifesto.org/iso/pl (6.04.2013).

Miłosz M., Borys M., Plechawska-Wójcik M., 2011, *Metodyki zwinne wytwarzania oprogramowania,* Politechnika Lubelska, Lublin.

Sacha K., *Inżynieria oprogramowania*, Wydawnictwo Naukowe PWN, Warszawa 2010.

Sahota M., 2012, *An Agile Adoption and Transformation Survival Guide*, Working with Organizational Culture.

Schwaber K., Sutherland J., 2011, *The Scrum Guide. Przewodnik po Scrumie*: *Reguły gry*, Scrum.org.

Stepanek G., 2005, *Software Project Secrets*: *Why Software Projects Fail*, Apress, New York.

Tomasini A., Kearns M., 2012, *Agile Transition. What you Need to Know Before Starting*, INFO Q Enterprise Software Development Series.

## WHEN IS IT WORTH TO USE AGILE METHODOLOGIES IN SOFTWARE DEVELOPMENT PROJECT MANAGEMENT PRACTICE?

**Summary:** The article presents increasingly popular agile methodologies that are more and more often used for projects management of software development. The aim of this article is to analyze the concept of agility and agile methodologies, as well as to find an answer to the question when it is worth to use such methodologies which require organizational effort as-

sociated with their use, acquisition of the necessary skills and covering the necessary costs. We consider agile methodologies as the specific project management methodologies in software development. Therefore the article describes their role in the development of agile IT infrastructure, agile organizational culture, and issues related to the possibility of assessing the level of methodologies' agility, as well as the possibility of evaluation the purposefulness and benefits of using them in software development practice.

**Keywords:** project management, software development, agile methodologies, evaluation of software design methodologies, evaluation of IT projects.