

Henryk KRAWCZYK
Politechnika Gdańska

DWUPOZIOMOWE PROCESY WYTWARZANIA INTELIGENTNYCH PLATFORM I USŁUG CHMUROWYCH

Streszczenie. Dzięki rozwojowi chmur obliczeniowych i metod sztucznej inteligencji zmieniają się zarówno metody wytwarzania jak i możliwości funkcjonalne usług IT. Dominującą metodą wytwarzania jest obecnie DevOps (Development/ Operations) wspomagana techniką CI/CD (Continuous Integration/Continuous Development). W artykule przedstawiono propozycję budowy platformy chmurowej CAISE (Cloud Artificial Intelligence Service Engineering) wspomagającej procesy wytwarzania, uruchamiania i wdrażania usług inteligentnych, przydatnych w różnych obszarach zastosowań. Zakłada się że procesy te są realizowane równolegle na dwóch poziomach. Procesy nadrzędne dotyczą cykli odpowiadających uzyskiwanie kolejnych poziomów gotowości technologicznej platformy CAISE, zaś procesy podrzędne związane są z rozwojem wymaganej funkcjonalności tej platformy. Dodatkowo oryginalnym podejściem jest również włączenie w te procesy wytwarzania procedur kontroli jakości, tak by finalny produkt osiągnął zarówno odpowiednie wymagania funkcjonalne jak i jakościowe.

TWO-TIER PROCESSES FOR DEVELOPING INTELLIGENT CLOUD PLATFORMS AND SERVICES

Summary. Thanks to the development of cloud computing and artificial intelligence methods, both the methods of design and the functional capabilities of IT services are changing. The dominant method of cloud product development is currently DevOps (Development/ Operations) supported by the CI/CD (Continuous Integration/Continuous Development) technique. The article presents a proposal for the construction of a CAISE (Cloud Artificial Intelligence Service Engineering) platform supporting the processes of manufacturing, launching and implementing intelligent services, useful in various areas of applications. It is assumed that these processes are carried out in parallel at two levels. Parent processes refer to the cycles corresponding to obtaining successive levels of technological readiness of the CAISE platform, while child processes are related to the development of the required functionality of this platform. In addition, an original approach is also to include quality control procedures in these manufacturing processes, so that the final product meets both the appropriate functional and quality requirements.

1. Wprowadzenie

Wraz z rozwojem technologii informatycznych (IT) wzrasta znaczenie usług IT, które wspomagają działalność człowieka w jego różnych obszarach życia [1, 2]. Przez usługę IT określa się każdy komponent oprogramowania, mogący działać niezależnie od innych oraz posiadający zdefiniowany interfejs, za pomocą którego udostępnia swoje realizowane funkcje. Tak więc sposób działania każdej usługi jest w całości zdefiniowany przez interfejs, zaś szczegóły implementacyjne są niewidoczne i nieistotne z punktu widzenia klientów. Dodatkowo, istnieje wspólne, dostępne dla wszystkich usług medium komunikacyjne, umożliwiające swobodny przepływ wymaganych danych.

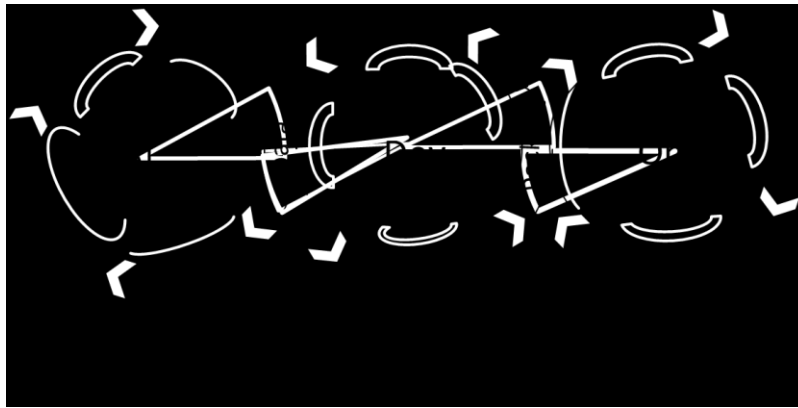
Każda chmura obliczeniowa [2] zapewnia dostęp do usług IT w warstwie SaaS (*ang. Software as a Service*), na ogół przez przeglądarkę internetową. Same usługi są implementowane na bazie różnych technologii, osadzone na maszynach wirtualnych bądź w kontenerach i udostępniane za pomocą niezależnego protokołu komunikacyjnego. Typowe obliczenia chmurowe obejmują zazwyczaj usługi związane z bazami danych, biznesem elektronicznym, czy analityką danych. Warstwa chmury obliczeniowej PaaS (*ang. Platform as a Service*) jest środowiskiem do wytwarzania i wdrażania usług i aplikacji. Wspiera cały cykl ich życia i pomaga użytkownikom w tworzeniu, testowaniu, wdrażaniu, zarządzaniu i aktualizowaniu wszystkich aplikacji na jednej platformie. Obejmuje również narzędzia deweloperskie, w tym rozwiązania CI/CD do monitorowania i testowania. Z kolei warstwa IaaS (*ang. Infrastructure as a Service*) jest środowiskiem wykonania usług i pozwala na wykorzystanie podstawowych zasobów infrastruktury chmury (węzły obliczeniowe - rdzenie i pamięć operacyjna, dyski przechowujące dane oraz porty komunikacyjne do wymiany informacji). Użytkownicy usług chmurowych nie muszą przechowywać plików, programów czy danych na dysku własnego komputera lub na serwerze dedykowanym. Są one utrzymywane w centrach danych, a klienci chmury mają do nich dostęp zdalny z dowolnego miejsca na świecie. Co więcej każdy użytkownik może dysponować wielkościami przestrzeni dyskowej i mocy obliczeniowej, jaka jest mu w danej chwili potrzebna (cecha elastyczności), co oznacza zdolności chmury do zmniejszenia lub zwiększania wielkości przydzielanych zasobów.

Możliwości funkcjonalne chmury mogą być rozbudowywane w zależności od potrzeb i wykorzystywanych narzędzi i środowisk. W dużej mierze decyduje o tym specyfika budowanych usług i aplikacji, te ostatnie na ogół reprezentują odpowiednie scenariusze wykonania usług. Przyjmujemy, że naszym celem jest wykorzystanie platformy chmurowej do wspomagania budowy usług inteligentnych związanych z analizą dokumentów cyfrowych. Jak wiadomo tego typu dokumenty, dzięki postępującej cyfryzacji, są wykorzystane we wszystkich dziedzinach działalności człowieka. W związku z tym podjęto się zadań rozbudowy warstwy chmurowej PaaS dotyczącej wspomagania procesów wytwarzania jak i wytworzenia reprezentatywnego zbioru takich usług. Istotnym wymaganiami jest zapewnienie najwyższego stopnia gotowości technologicznej takiej platformy [4] jak i odpowiedniej jakości [5] wytworzonych usług, przy założeniu, że zarówno platforma jak i usługi są rozwijane równolegle. W artykule wyróżniono dwa poziomy wytwarzania: jeden związany z procesami budowy takiej platformy (poziom technologiczny) oraz drugi z procesami

budowy usług inteligentnych (poziom usług), przy założeniu cyklicznej kontroli poziomu gotowości technologicznej platformy oraz jakości wytwarzanych usług. Takie rozwiązanie przyspiesza procesy wytwarzania i jest realizowane w projekcie o akronimie CAISE (*ang. Cloud Artificial Intelligence Service Engineering*) w ramach programu europejskiego Important Project of Common European Interest (IPCEI) Cloud Infrastructure and Services (CIS). W artykule przedstawiono jedynie nieformalne podejście, które może być następnie podstawą poszukiwania podstaw formalnych, mających na celu optymalizację efektywności takiego zestawu kooperujących procesów.

2. Metoda wytwarzania platformy i usług

Sztuczna inteligencja (AI) [6] odgrywa coraz większą rolę w różnych obszarach życia codziennego, gdzie tradycyjne usługi IT zastępuje się inteligentnymi usługami wykorzystującymi różne modele ML (*ang. Machine Learning*) [7]. Przygotowanie takich usług jest złożonym przedsięwzięciem badawczo-rozwojowym gdyż wymaga dużej wiedzy z zakresu AI oraz środowiska w którym taka usługa będzie utworzona. Na ogół różne firmy IT wykorzystują różne środowiska wytwarzania zorientowane na dany obszar zastosowań. To oznacza, że wiele prac przez te firmy jest powielanych, zwłaszcza dotyczących wykreowania dedykowanej platformy, co jest kosztowne i czasochłonne. Wyjściem naprzeciw takim rozwiązaniom jest budowa uniwersalnej platformy, umożliwiającej budowę usług inteligentnych w różnych sektorach działalności ludzkiej. Takie działanie zainicjował właśnie program europejski IPCEI-CIS [8], którego najważniejszym przedsięwzięciem jest zbudowanie Europejskiej przestrzeni cyfrowej, tzw. Cloud Continuous opartej o technologie chmurowe i IoT. Możliwości sztucznej inteligencji są wykorzystane również w metodach inżynierii oprogramowania [9]. Znana i popularna metodyka DevOps (*ang. Development /Operation*) jest rozbudowana do AI+DevOps [10] (rys. 1). Stanowi to istotną innowację (I), gdzie cykle R+D (*ang. Research and Development*) przeplatają się z cyklami wdrożeniowymi FID (*ang. First Investment Deployment*). Metodyka DevOps współdziała z techniką CI/CD [11] zapewniając nową metodę wytwarzania platform i aplikacji chmurowych. W wielu raportach firm podaje się, że przyjęcie już takiego sposobu działania daje szansę na szybsze wdrażanie produktów oraz niższy wskaźnik awarii (patrz raport DORA 2019 State of DevOps). Zespoły wytwarzające produkty oprogramowania są informowane o wszelkich anomaliach (błędach) lub incydentach (np. sprzecznościach) pojawiających się podczas procesu wytwarzania. Wykorzystując klasyczne metody wytwarzania zespoły mogą optymalizować swoją pracę albo pod kątem szybkości realizacji albo jakości wykonania. Ciągłe informacje zwrotne stanowiące istotne elementy metodyki DevOps, umożliwiają osiągnięcie obu tych celów jednocześnie. Ciągłe doskonalenie tej metodyki jest również istotnym wyzwaniem i sprowadza się na ogół do sformułowania tzw. dobrych praktyk wytwarzania oprogramowania. Istotny jest elastyczny dobór cykli wytwarzania, bowiem krótsze cykle wytwarzania ułatwiają sprawniejsze planowanie i zarządzanie ryzykiem, ale z uwagi na to że postęp projektu jest przyrostowy, to może to ograniczać stabilność rozwiązań. Dlatego cykliczna kontrola procesu wytwarzania jest również sprawą istotną.

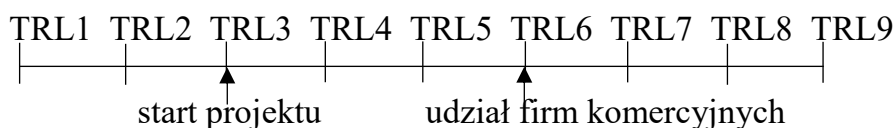


Rys. 1. Graficzna ilustracja metody AI+DevOps

Proponowana metoda wytwarzania zakłada, że wytwarzanie platformy oraz usług odbywa się cyklicznie, metodą iteracyjno-przyrostową, na dwóch poziomach; przy czym cykle technologiczne są nadrzędne w stosunku do cykli usługowych. Rysunek 2 stanowi rozwinięcie w czasie cykli prezentowanych na rysunku 1. I tak w cyklach technologicznych następuje rozbudowa platformy związana z zapewnieniem kolejnych poziomów gotowości technologicznej. Zakłada się przy tym, że start projektu rozpoczyna się od poziomu gotowości technologicznej TRL3, zaś od osiągnięcia poziomu TRL 6 dalsze udoskonalenie platformy będzie realizowane przy współpracy firm/kooperantów. W cyklach usługowych dokonuje się sprawdzanie na bieżąco funkcjonalności platformy poprzez realizowanie w sposób sekwencyjno-równoległy różnych jej usług; w tym zarządzania platformą jak też analizy treści dokumentów cyfrowych. Oczywiście jest, że w jednym cyklu technologicznym może być realizowane wiele takich usług, poprzez różne zespoły, a wraz z rozwojem platformy liczba wytworzonych usług oraz ich jakość powinna stale wzrastać.

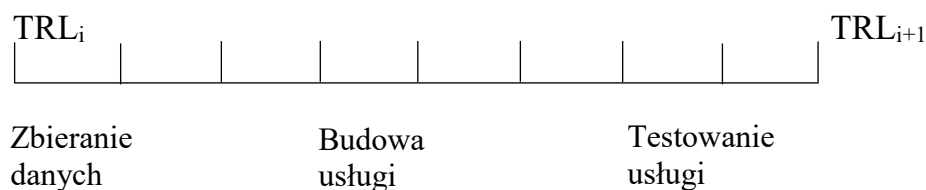
a) Cykle technologiczne:

Poziomy gotowości technologicznej:



b) Cykle usługowe dla

i-tego poziom gotowości technologicznej:



Rys. 2. Ilustracja technologicznych i usługowych cykli rozwoju platformy CAISE

3. Ocena gotowości technologicznej platformy

Przyjęto, że w cyklach technologicznych będzie następowała rozbudowa platformy związana z uzyskiwaniem jej kolejnych poziomów gotowości technologicznej. Definicja poziomów gotowości technologicznej podaje tabela 1. Oznacza to również stopniowe zwiększanie podstawowych charakterystyk jakościowych sygnalizowanych również w tej tabeli. W cyklach funkcjonalnych dokonuje się rozbudowa funkcjonalności platformy CAISE, w których realizowane są w sposób sekwencyjny bądź równoległy kolejne jej usługi. Po włączeniu się w proces wytwarzania platformy firm komercyjnych, dzięki dostarczonym zbiorom danych, następuje wytwarzanie usług ściśle związanych z przetwarzaniem dokumentów cyfrowych związanych z danym sektorem działalności publicznej. Poza tym rozpatrywane i implementowane będą różne przypadki użycia takich usług w celu dalszego testowania i oceny jakości rozwijanej platformy.

Tabela 1

Charakterystyka cykli technologicznych przy realizacji platformy CAISE

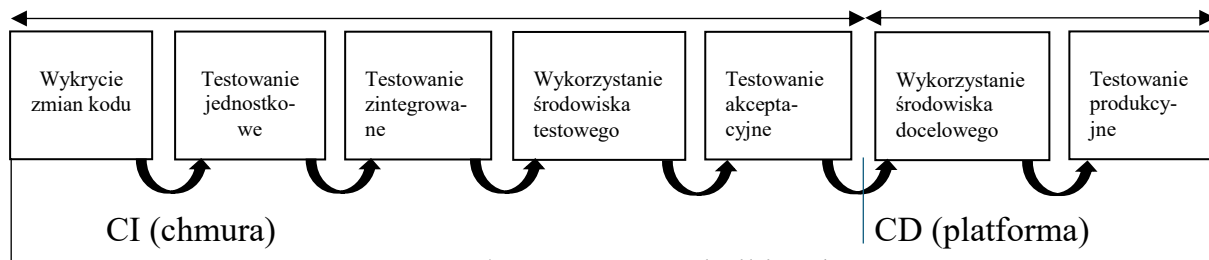
| Przejście z poziomu TRL _i do TRL _{i+1} | Charakterystyka ciągłych działań | Uwagi |
|--|---|---|
| TRL1 → TRL2 | Badania naukowe, prace magisterskie i doktoraty, koncepcja platformy CAISE | Działania badawcze przed realizacją platformy i usług. Opracowanie koncepcji platformy. |
| TRL2 → TRL3 | Wykorzystanie otwartych rozwiązań programistycznych integrujących technologie: HPC, Cloud Computing, Big Data, AI. | |
| TRL3 → TRL4 | Adaptacja chmury do otwartych rozwiązań wspomagających funkcjonowanie platformy-usługi na poziomie IaaS oraz PaaS. Uwzględnienie wymagań funkcjonalności i integralności. | Wykorzystanie otwartego oprogramowania typu OpenStack czy Kubernetes> Laboratoryjna wersja platformy. |
| TRL4 → TRL5 | Rozwój różnego typu usług na chmurze i adaptacja ich do rozwijanej platformy Wykorzystanie DevOps oraz technik CI/CD. Zapewnienie automatyzacji i efektywności działania. | Wspomaganie i udoskonalenie metod wytwarzania platformy i usług. |
| TRL5 → TRL6 | Dalszy rozwój usług z uwzględnieniem obserwowalności procesów wytwarzania jak i dostępności do danych i usług. | Repozytoria usług oraz danych oraz badanie platformy w środowisku symulowanym. |
| TRL6 → TRL7 | Rozwój usług dla wskazanych sektorów zastosowań z zapewnieniem uniwersalności rozwiązań i akceptowalności rynkowej. | Wzbogacenie danych uczących i testujących oraz badanie platformy w środowisku operacyjnym. |
| TRL7 → TRL8 | Realizacja i analiza potoków usług w celu potwierdzenia wymaganej funkcjonalności, efektywności i bezpieczeństwa. | Badania platformy w środowisku rzeczywistym, tzn. dla wskazanej analizy dokumentów cyfrowych. |
| TRL8 → TRL9 | Udostępnienie platformy dla zainteresowanych budową klientów rynkowych. Ocena dojrzałości oraz użyteczności platformy. | Faza pierwszego wdrożenia i oceny użyteczności platformy. |

Poziomów TRL systemu nie wiąże się bezpośrednio ze wskaźnikami wymiernymi, które przyjmują konkretne wartości, a tworzy się listę sprawdzającą (check list) złożoną z konkretnych pytań dla każdego poziomu. Proponowane są takie zestawy pytań dla różnego typu budowanych systemów. Pozytywne odpowiedzi na te pytania stanowią podstawę do akceptacji osiągnięcia danego poziomu TRL.

Przykładem takiego podejścia jest TRL Worksheet firmy NASA (https://esto.nasa.gov/files/TRL_Worksheet_11-30-10.xls), który ocenia poziomy TRL z dołu do góry, rozpoczynając od komponentów, a następnie koncentrując się na systemie złożonym z tych komponentów. Każdemu poziomowi odpowiada zadany zestaw pytań sprawdzających spełnienie odpowiednich kryteriów akceptacji. Na poziomie TRL3 sprawdzane jest czy funkcje krytyczne komponentów działają prawidłowo i czy akceptowane są ich parametry jakościowe (wydajnościowe) w środowisku eksperymentalnym. Dotyczy to na ogół dostępnych pakietów otwartego oprogramowania. Na poziomie TRL4 sprawdza się czy funkcje złożone (wynikające z integracji tych pakietów) spełniają założone wymagania w środowisku laboratoryjnym. Dostępne funkcje platformy oraz jakość ich wykonania sprawdza się na poziomie TRL5 już w środowisku symulacyjnym zbliżonym do rzeczywistego. Na poziomie TRL6 czyni się to na podstawie działającego już prototypu platformy. Akceptacja poziomu TRL7 związana jest z kolei z oceną działania platformy w finalnym środowisku operacyjnym. Poziom TRL8 weryfikuje się w środowisku rzeczywistym (rynkowym) funkcjonowania platformy. Stąd potrzeba wykorzystania firm komercyjnych w procesie jej wytwarzania. Reasumując dla akceptacji danego poziomu gotowości technologicznej istotne jest więc przeprowadzenie badań funkcjonalności w zadanym środowisku, a wyniki oceny zachowania się platformy w takich środowiskach stanowią podstawę akceptacji jej danego poziomu gotowości technologicznej. Co więcej, wraz z rozwojem środowiska platformy wzrasta również jej funkcjonalność, stąd proces weryfikacji i walidacji staje się bardziej złożony. W rezultacie powstaje zbiór zrealizowanych usług, również na coraz wyższym poziomie jakości.

4. Ocena jakości usług platformy

W ocenie jakości usług istotną rolę odgrywa współdziałanie metodyki AI+DevOps (rys. 1) z techniką CI/CD. Metoda CI/CD odnosi się przede wszystkim do testowania kodu implementowanych czynności (przede wszystkim usług chmurowych) w środowisku wytwarzania, w szczególności w warstwie PaaS (rys. 3). Ostatecznie walidacja usługi sprawdza się w środowisku docelowym. Do tego celu wykorzystuje się oprogramowanie otwarte GitLab oraz aktualną funkcjonalność platformy CAISE w postaci dodatkowych usług związanych z zarządzaniem i monitorowaniem procesów wytwarzania. Wyniki tych testów stanowią podstawę do oceny zarówno wytworzonej usługi jak i realizowanej platformy. Dodatkowo wykorzystane są dostępne otwarte narzędzia monitorująco-raportujące np. Prometheus, narzędzia zarządzania użytkownikami, np. Keycloak, zarządzania potokami usług, np. Argo czy raportowania błędów, np. Zammad. Dzięki możliwościom chmury TASKcloud możemy również dobrać odpowiednie zasoby fizyczne chmury (rdzenie, pamięć, dyski, sieć) tak by osiągnąć najlepszą wydajność funkcjonowania usług i platformy.



Rys. 3. Główne etapy techniki CI/CD

Powyższe rozwiązania wymagają również zapewniania odpowiedniej obserwowalności realizowanych zadań i procesów wykonywanych na chmurze obliczeniowej wraz z zintegrowaną z nią platformą. Dotyczy to przede wszystkim procesów monitorowania zarówno powstającego kodu jak i działań ludzkich, które mają duży wpływ na efektywność procesów wytwarzania. Do oceny jakości platformy oraz usług wykorzystane mogą być Kluczowe Wskaźniki Efektywności KPI (*ang. Key Performance Indicators*) [12], określające mierzalne wartości, które pokazują postęp w realizacji projektu, w takich obszarach takich jak przyspieszenie cykli wydawania produktu, poprawa jego jakości czy sterowanie jego wydajnością. Dobór odpowiednich wskaźników jest sporym wyzwaniem i może być modyfikowany nawet w trakcie rozwoju produktów (w naszym przypadku usług oraz platformy jako środowiska ich wykonania). Tego typu wskaźniki będą rejestrowane i śledzone przy ocenie porównawczej procesów wytwarzania różnych usług, bądź kolejnych etapów rozwoju platformy. Z reguły przyjmuje się następujące wskaźniki:

- dla usług :
 - czas cyklu wytwarzania, inaczej okres od zatwierdzenia usługi do realizacji po wdrożeniu jej kodu,
 - czas przywracania sprawności jako czas potrzebny na przywrócenie usługi do działania od momentu wystąpienia przerwy w jej działaniu (pojawienie się anomalii).
- dla platformy:
 - czas uruchomienia danego środowiska,
 - czas integracji jego z innym środowiskiem,
 - czas obsługi anomalii czy incydentów występujących w czasie wytwarzania usług,
 - średni czas przywrócenia do stanu sprawności po wystąpieniu anomalii lub incydentu,
 - czas dostrojenia platformy do wymagań wydajnościowych czy energetycznych.

W projekcie CAISE proponuje się bardziej zaawansowaną metodę oceny jakości wytwarzanej platformy i usług, w tym celu wykorzysta się model jakości przedstawiony w tabeli 2. Wyróżniono trzy rodzaje zaproponowanych metryk jakości:

1. Ilościowe – opisujące wielkość wykorzystywanych zasobów chmury, liczbę dostępnych usług, liczbę pojawiających się incydentów, liczbę wyeliminowanych błędów, jakość danych, modeli, itp.

2. Czasowe – związane z czasem wykonywania różnych czynności takich jak: czasy realizacji cykli, czasy tworzenia i testowania usług, czasy wprowadzania zmian, czy czasy pomiędzy pojawiającymi się incydentami, itp.
3. Ankietowe – wyznaczanie wartości metryk na podstawie przeprowadzanych ankiet, np. stopień automatyzacji procesu wytwarzania usług, czy stopień integracji narzędzi chmurowych wspomagających ich wytwarzanie, itp.

Tabela 2

Opis modelu jakości platformy i usług

| | Atrybuty | Charakterystyki | Metryki |
|--|---|--|--|
| Model jakości platformy (w trakcie jej wytwarzania i eksploatacji) | Funkcjonalność: wykaz funkcji dostępnych dla użytkowników platformy | <ul style="list-style-type: none"> – Cechy środowiska wytwarzania usług inteligentnych Y – Własności usług Z wspomagających funkcjonowanie platformy – Możliwości chmury-środowiska działania platformy usług X | <ul style="list-style-type: none"> – Lista dostępnych usług i danych (wielkości repozytoriów) – Stosunek liczby wykorzystywanych usług do liczby oferowanych usług – Stopień integralności wykorzystanych narzędzi i oprogramowania – Stopień automatyzacji procesu wytwarzania |
| | Produktywność: efektywność wytwarzania i wykonywania usług | <ul style="list-style-type: none"> – Dobre praktyki metodyki wytwarzania usług – Zakres skalowalności procesu wytwarzania – Jakość wytworzonych usług – Wydajność i energooszczędność | <ul style="list-style-type: none"> – Średnie czasy trwania cyklu technologicznego i funkcjonalnego – Min i max czasy wytwarzania (wdrożenia) usług oraz ich wykonania – Min i max wielkości zasobów wymagane do funkcjonowania platformy, wytwarzania oraz wykonania usługi – Średni czas dokonywania zmian platformy (<i>lead time for changes</i>) – Jakość danych, jakość modelu wyrażone poprzez jakość usługi |
| | Wiarygodność: funkcjonowanie platformy - dostępność realizowanych funkcjonalności | <ul style="list-style-type: none"> – Rozwiązywalność pojawiających się problemów w procesie wytwarzania usług – Zdolność eliminacji błędów podczas funkcjonowania platformy – Poziom bezpieczeństwa platformy | <ul style="list-style-type: none"> – Liczba pojawiających się problemów w cyklu funkcjonalnym i technologicznym – Średni czas eliminacji incydentów czy błędów pojawiających się podczas wytwarzania usługi czy rozwoju platformy (<i>lead time</i>) – Częstość pojawiania się błędów podczas wykorzystania platformy – Liczba zmian (kodu) zakończona niepowodzeniem (<i>Change Failure Rate</i>) – Współczynnik zaniedbania defektów (<i>Defect Escape Rate</i>) – Liczba zmian w zadanym przedziale czasu (<i>Deployment Frequency</i>) |

| | | | |
|--|---|---|--|
| | <p>Dojrzałość platformy: jej akceptowalność przez różnych klientów oraz w różnych sektorach zastosowań</p> | <ul style="list-style-type: none"> – Poziom gotowości technologicznej – Akceptowalność przez klientów – Przenoszalność na inne środowiska chmurowe | <ul style="list-style-type: none"> – Faktyczny poziom gotowości technologicznej – Stopień użyteczności - stosunek liczby pozyskiwanych klientów do liczby zakładanych klientów – NPS (<i>Net Promoter Score</i>) - ilu klientów poleca produkt innym podmiotom (ankiety) – Osiągnięty poziom CMM (<i>Capability Maturity Model</i>) wg standardu – Częstość zgłaszanych uwag klientów w czasie realizacji usług na wdrożonej platformie |
|--|---|---|--|

Przedstawione metryki mogą przyjmować różne wartości, jednak dla dokonania analizy jakościowej powinny zostać zunifikowane do przedziału $\langle 0, 1 \rangle$, gdzie 0 oznacza najniższą wartość jakości, zaś 1 jej najwyższą wartość. Weźmy pod uwagę metrykę zaniedbania defektów, który pomaga określić skuteczność procesów testowania, a także ogólną jakość usługi. Idealnie byłoby jednak, gdyby wszystkie błędy były wychwytywane podczas faz rozwoju i testowania procesu (DevOps) wytwarzania platformy, a nie w procesach jej produkcji/wdrożenia. Wysoka wartość tego wskaźnika sugeruje, że procedury testowania wymagają poprawy i że potrzebna jest większa ich automatyzacja, podczas gdy niski wskaźnik (najlepiej bliski zeru) oznacza aplikację wysokiej jakości. W takim przypadku wymagana będzie redefinicja metryki by dopasować się do powyżej przyjętej interpretacji przedziału $\langle 0, 1 \rangle$. W tym przypadku nazwę metryki (gdzie ukryta jest bezpośrednio jej definicja) – „zaniedbania defektów” należy zmienić na metrykę „zadbania o eliminację defektów”.

5. Uwagi końcowe

W artykule przedstawiono realną metodę wytwarzania platform i usług chmurowych, która jest realizowana na chmurze TASKcloud funkcjonującej już w Centrum Informatycznym TASK Politechniki Gdańskiej od kilku lat. Przedstawiona metoda budowy pozwala zarówno na kontrolę poziomów gotowości realizowanej platformy jak też ocenę jakości tworzonych usług. To znacznie skraca czas jej wytworzenia, a jednocześnie zapewnia uzyskanie najwyższej gotowości technologicznej. Po uzyskaniu poziomu gotowości TRL 9 platforma może być wykorzystana już tylko do wytwarzania kolejnych usług przydatnych do analizy dokumentów cyfrowych. Liczba takich usług zależy tylko od pomysłowości twórców, ale będzie weryfikowana poprzez realną ich przydatności w praktyce.

W czasie realizacji przedstawionych procesów wytwarzania istnieje wiele problemów optymalizacyjnych, np. w jaki sposób zredukować czas wytwarzania czy wykonywania usług, w jaki sposób zmniejszyć zużycie energii (problem doboru infrastruktury chmurowej na poziomie IaaS). Cenne byłoby powiązanie metod oceny metryki jakości z osiąganym poziomem gotowości technologicznej. Innym zagadnieniem jest zapewnienie wymaganego poziomu wiarygodności czy bezpieczeństwa platformy jak najniższym kosztem. Otwartym problemem pozostaje rozwój metod

DevOps oraz CI/CD umożliwiającą wyższą automatyzację procesów wytwarzania czy określenie dobrych praktyk dla przyjętej metodyki wytwarzania oprogramowania. Przy rozwiązywaniu tych problemów konieczna byłaby formalizacja zaproponowanej metody wytwarzania lub co najmniej jej niewrażliwych elementów, tak by można byłoby na te pytania odpowiedzieć wcześniej niż dopiero po wykonaniu bardzo kosztownych eksperymentów ściśle związanych z przedstawioną w tym artykule (w skrócie) metodą wytwarzania platform i usług chmurowych.

Podziękowania: Praca została zrealizowana w ramach projektu pt. Chmurowa platforma (CAISE) do wytwarzania uniwersalnych usług inteligentnych dla różnych obszarów zastosowań, nr KPOD.05.10-IW.10-0005/24), program IPCEI-CIS.

LITERATURA

1. Stojanovic Z., Dahanayake A.: Service-Oriented Software System Engineering. Challenges and Practices, IGI Global Idea Group Publishing, 2005.
2. Krawczyk H., Kaczmarek S., Nowicki K.: Aplikacje i usługi a technologie sieciowe. Warszawa: Wydawnictwo Naukowe PWN, 2018. 414 s. ISBN 978-83-01-19959-3.
3. Manvi S., Shyam G. K.: Cloud Computing Concepts and Technologies, CRC Press, Taylor & Francis Group, LLC, 2021.
4. Poziom gotowości technologicznej: https://en.wikipedia.org/wiki/Technology_readiness_level.
5. Jelassi M., Ghazel Ch., Saidane L. A.: A survey on quality of service in cloud computing, International Conference on Frontiers of Signal Processing (ICFSP), 2017, <https://doi.org/10.1109/ICFSP.2017.8097142>.
6. Russell S., Norvig P.: Artificial Intelligence: A Modern Approach. Pearson Education Inc., 2021.
7. Bishop Ch.M., Bishop H.: Deep Learning: Foundations and Concepts, Springer, 2023.
8. IPCEI-CIS: <https://digital-strategy.ec.europa.eu/pl/news/ipcei-next-generation-cloud-infrastructure-and-services-boost-europes-digital-decade>.
9. Staron M.: Machine Learning Infrastructure and Best Practices for Software Engineers. Packt Publishing, (ebook).
10. Davis J., Daniels R.: Effective DevOps, O'Reilly Media, Inc., 2016.
11. Merode H.: Continuous Integration (CI) and Continuous Delivery (CD): A Practical Guide to Designing and Developing Pipelines, Apres, 2023.
12. Parmenter D.: Kluczowe wskaźniki efektywności (KPI) Tworzenie, wdrażanie i stosowanie, Helion, 2021.