

Odważa spojrzenia w siebie, czyli o ocenie procesu wytwórczego oprogramowania

Efektywność procesów wytwórczych oprogramowania jest przedmiotem szczególnej uwagi w przypadku dostawców i integratorów rozwiązań IT. Ma ona też istotne znaczenie dla tych podmiotów, w których IT jest immanentnym elementem przyjętego modelu biznesowego, zwłaszcza w firmach sektora e-commerce, bankowości, ubezpieczeń czy telekomunikacji. Podmioty te są zdecydowanie mniej skłonne do powierzania organizacjom zewnętrznym zadań związanych z wdrażaniem i rozbudowywaniem systemów informatycznych, ze względu na istotność tych aktywów w budowaniu przewagi konkurencyjnej.

Oczywiście przekonanie takie nie wyklucza zaangażowania zewnętrznych dostawców usług IT. Jednak jest ono podstawą do konstatacji, iż dla skutecznego działania organizacji niezbędne jest utrzymywanie, skrojonego na miarę potrzeb, własnego zespołu deweloperskiego, a przede wszystkim wdrożenie efektywnego, zarządzanego i udokumentowanego procesu wytwórczego. Nie ulega bowiem wątpliwości, że dojrzałość procesu wytwórczego przekłada się jednoznacznie na sukces projektów i jakość tworzonego oprogramowania.

PODSTAWA DOBREGO PROCESU WYTWÓRCZEGO

Jest nią usankcjonowanie w organizacji jednolitych standardów analitycznych, projektowych i dokumentacyjnych dla przedsięwzięć informatycznych. Powinny one w szczególności objąć:

- standardy modelowania (UML, BPMN);
- zasady prowadzenia analizy biznesowej, zwłaszcza zasady modelowania procesów i zarządzania wymaganiami;
- zasady prowadzenia analizy systemowej, w tym zasady modelowania przypadków użycia oraz budowania modelu logicznego;
- standardy projektowania, w tym:
 - standard zarządzania projektem,
 - standard interfejsu użytkownika,
 - standard kodowania, w tym konwencje kodowania,
 - standard projektowania interfejsów,
 - model platformy,
 - standard schematu danych,
 - standard zarządzania zmianami,
 - wymogi dokumentacyjne,
- zasady prowadzenia prac w repozytorium analitycznym.

Od problemów do rozwiązań

Proces wytwórczy obejmuje zestaw metod, technik i narzędzi używanych do budowy, wdrożenia i eksploatacji systemów informatycznych. Metody te dotyczą sposobów analizy, projektowania i kodowania, a także planowania i kontroli projektu oraz zapewniania jakości. Praktyka wskazuje, że kierownictwo organizacji tworzących oprogramowanie – czy to na potrzeby własne, czy swoich klientów – wielokrotnie otrzymuje informacje będące symptomami problemów dotyczących realizacji projektów IT. Zagadnienia te związane są z przestojami w działaniu aplikacji, utratą danych, pracochłonnością całego procesu, ale też niespełnionymi oczekiwaniami biznesu. W większości sytuacji brakuje informacji niezbędnych do podjęcia odpowiednich działań, które mają rozwiązać pojawiające się problemy lub zapobiec im w przyszłości. Zamierzonym celem każdej organizacji powinna być tym samym precyzyjna identyfikacja problemów związanych z realizacją projektów informatycznych, tak aby możliwe było podjęcie niezbędnych działań naprawczych. Praktyka wskazuje, że obszary szczególnej uwagi powinny dotyczyć kwestii związanych z:

- zapewnieniem właściwej komunikacji – pomiędzy osobami zamawiającymi oprogramowanie a osobami je realizującymi;
- zdolnością planowania i rozliczania prac analitycznych, programistycznych, a także prac dotyczących realizacji procesów wsparcia klienta;
- skutecznością procesu zarządzania zmianą – w rozumieniu zdolności do implementacji zmian w funkcjonującym oprogramowaniu;
- właściwym dokumentowaniem prac analitycznych i programistycznych.

Koncentracja na tych czterech obszarach powinna pozwolić na zwiększenie produktywności zespołów wytwórczych, poprawę jakości produktów oraz zwiększenie terminowości realizacji projektów. Zasadnym jest, aby szczególną troską otoczyć aspekt wypracowania mechanizmu uzależnienia wynagrodzenia zespołu od skuteczności prac projektowych, który w przypadku większości dojrzałych integratorów przesądza o skuteczności ich funkcjonowania.

Na przestrzeni ostatnich dwóch dekad wyewoluowały dwa modele oceny procesów wytwórczych: CMM (Capability Maturity Model) opracowany przez amerykańską organizację Software Engineering Institute (SEI) oraz SPICE (Software Process Improvement and Capability dEtermination) będący efektem prac w ramach projektu powołanego przez International Standard Organization (ISO). Oba

modele znajdują swoje odzwierciedlenie w aktualnie obowiązujących rewizjach modeli: CMMI Capability Maturity Model Integration for Development oraz ISO TR 33015: Information Technology – Process Assessment – Guidance for Process Risk Determination, która zastąpiła wcześniej obowiązującą normę ISO/IEC 15504.

Modele oceny

Na przestrzeni ostatnich dwóch dekad wyewoluowały dwa modele oceny procesów wytwórczych: CMM (Capability Maturity Model) opracowany przez amerykańską organizację Software Engineering Institute (SEI) oraz SPICE (Software Process Improvement and Capability dEtermination) będący efektem prac w ramach projektu powołanego przez International Standard Organization (ISO). Oba modele znajdują swoje odzwierciedlenie w aktualnie obowiązujących rewizjach modeli: CMMI Capability Maturity Model Integration for Development oraz ISO TR 33015: Information Technology – Process Assessment – Guidance

ce for Process Risk Determination, która zastąpiła wcześniej obowiązującą normę ISO/IEC 15504.

W praktyce można się spotkać również z wykorzystaniem ram oceny zgodności procesów IT Governance COBIT (Control Objectives for Information and related Technology) opracowanego przez IT Governance Institute, obejmującego wszystkie niezbędne działania związane z nadzorem i zarządzaniem technologiami IT, nie mniej brak im ukierunkowania na proces wytwórczy, tak jak to jest w przypadku modeli opisanych normą ISO TR 33015 oraz CMMI.

Kontekst oceny

Niezależnie od przyjętej, sformalizowanej metody oceny, dla pełnego zrozumienia procesu wytwórczego funkcjonującego

w organizacji, istotne jest uwzględnienie kontekstu środowiskowego, na który składają się cztery podstawowe obszary:

- **produktu** – determinującego specyfikę oprogramowania, które jest tworzone,
- **kompetencji zespołu** – wykorzystywanego do produkcji oprogramowania,
- **narzędzi** – stosowanych w procesie wytwórczym,
- **procesu** – czyli przyjętego sposobu tworzenia oprogramowania.

Każdy ze wspomnianych aspektów wpływa wprost na ocenę procesu w kategoriach oczekiwanej efektywności oraz zakres rekomendowanych rozwiązań.

W obszarze produktowym w pierwszej kolejności ważne jest określenie podstawowej charakterystyki projektów prowadzonych przez organizację dotyczących przyjętych technologii wytwarzania, typowej architektury, rozmiaru tworzonego oprogramowania oraz jego przewidywanego czasu życia. Takie rozpoznanie pozwala w sposób płynny przejść do głębszej analizy związanej z weryfikacją adekwatności relacji kosztów i pracochłonności oprogramowania w stosunku do jego rozmiarów, przyjętymi metodami testowania, w szczególności zakresem wykorzystania testów automatycznych, liczbą błędów rejestrowanych na etapie testowania oraz po uruchomieniu eksploatacji, a także stopniem udokumentowania kodu oraz poziomem jego re-używalności. Spojrzenie na zespół umożliwia zidentyfikowanie podziału ról i obowiązków w ustalonej strukturze organizacyjnej oraz określenie stopnia zaangażowania członków zespołu w kluczowe zadania w procesie wytwórczym: od analizy wymagań, poprzez projektowanie architektury, tworzenie kodu, testowanie, dokumentowanie, aż do całościowego zarządzania projektem. Liczebność zespołu należy odnieść do wolumenu podejmowanych przez firmę zadań w poszczególnych projektach oraz liczby prowadzonych równolegle inicjatyw. Nie można pominąć oczywiście aspektu komunikacji oraz zarządzania wiedzą w zespole. Wielokrotnie u źródeł wszystkich problemów leży jakość prac programistycznych. Łamanie podstawowych zasad utrzymywania estetyki oraz struktury kodu źródłowego negatywnie wpływa na utrzymanie oprogramowania, jego aktualizację

W obszarze produktowym w pierwszej kolejności istotne jest określenie podstawowej charakterystyki projektów prowadzonych przez organizację dotyczących przyjętych technologii wytwarzania, typowej architektury, rozmiaru tworzonego oprogramowania oraz jego przewidywanego czasu życia. Takie

rozpoznanie pozwala w sposób płynny przejść do głębszej analizy związanej z weryfikacją adekwatności relacji kosztów i pracochłonności oprogramowania w stosunku do jego rozmiarów, przyjętymi metodami testowania, liczbą rejestrowanych błędów, a także stopniem udokumentowania kodu oraz poziomem jego re-używalności.

i możliwości rozbudowy. Brak adekwatnego udokumentowania kodu, w postaci czytelnych komentarzy, nie mówiąc już o testach automatycznych, nie jest niestety niczym wyjątkowym. Uzasadnieniem dla takiego stanu rzeczy jest najczęściej presja na krótkoterminowe efekty lub po prostu brak kompetencji. Próby porządkowania kodów źródłowych post factum zazwyczaj kończą się fiaskiem.

Komplementarnym dla spojrzenia na aspekt produktowy i zespołowy jest analiza narzędzi wykorzystywanych przez zespół w procesie wytwórczym, w szczególności narzędzi do gromadzenia i śledzenia realizacji wymagań, zarządzania wersjami oprogramowania, zgłaszania i śledzenia usuwania błędów w oprogramowaniu, komunikacji i dzielenia

ZAKRES MOŻLIWYCH ZMIAN W PROCESIE WYTWÓRCZYM OPROGRAMOWANIA MOŻE DOTYCZYĆ:

- planowania prac w zakresie budżetu, harmonogramu i zasobów na podstawie obiektywnych wskaźników,
- monitorowania postępu prac na bazie obiektywnych wskaźników,
- optymalizacji kosztów zamawiania prac u podmiotów zewnętrznych przy wykorzystaniu obiektywnych danych statystycznych własnych i rynkowych,
- sposobu dokumentowania przebiegu wytwarzania oprogramowania pod kątem możliwości jego zrozumienia, a także utrzymania i rozwoju opisanych w dokumentacji rozwiązań w razie odejścia z organizacji osób odpowiedzialnych za to oprogramowanie lub rezygnacji ze współpracy z podmiotami zewnętrznymi za nie odpowiedzialnymi.

się wiedzą, wspomaganie realizacji testów manualnych i automatycznych, a także specjalistyczne oprogramowanie do tworzenia i utrzymywania kodu.

Dopiero mając wiedzę o charakterystyce tworzonego oprogramowania, zespole go tworzącym oraz wsparciu narzędziowym procesu wytwórczego, można podejść do pełnej analizy samego procesu. Pierwszym krokiem jest zidentyfikowanie poziomu sformalizowania procesów, zweryfikowanie, na ile struktura procesowa zespołu odzwierciedla właściwe role jego członków oraz faktyczne kanały komunikacyjne. Na tej podstawie można określić optymalny stopień formalizacji procesu, jaki należałoby zastosować przy ustalonej specyfice produktu oraz zaproponować dodatkowe narzędzia mogące usprawnić proces. Nie można oczywiście zapomnieć o istotnych aspektach zapewnienia najlepszego po-

AUDYT PROCESU WYTWÓRCZEGO MOŻE POMÓC W ZWIĘKSZENIU JEGO EFEKTOWNOŚCI

Minimalny zakres audytu powinien dotyczyć podstawowych procesów związanych z zarządzaniem wymaganiami, analizą i projektowaniem rozwiązań, ich implementacją, testowaniem i zapewnianiem jakości, zarządzaniem projektem oraz całym procesem wytwórczym. Powinien on objąć w szczególności:

- identyfikację silnych i słabych stron obecnego procesu wytwórczego,
- poznanie problemów i zidentyfikowanie ich źródeł,
- opracowanie rekomendacji w zakresie poprawy procesu wytwórczego i organizacji pracy,
- zaplanowanie roadmapy działań zmierzającej do wdrożenia wskazanych rekomendacji,
- określenie krótkoterminowych działań naprawczych, m.in. usprawnienie poszczególnych czynności, szkolenia, narzędzia.

ziomu jakości produkowanego oprogramowania oraz odpowiedniego poziomu bezpieczeństwa.

Obawa przed spojrzeniem w siebie

Pomimo że nowoczesne metody zarządzania od wielu lat edukują nas w zakresie dobrych praktyk samodoskonalenia opartych na szczerej komunikacji i cyklicznych przeglądach procesów, nadal boimy się ocen i negatywnych ich konsekwencji. W tym przypadku, nawet gdy opór ten uda się pokonać, nie sposób zdiagnozować przyczyny problemów wyłącznie na podstawie uwidaczniających się kłopotów w prowadzonych projektach IT dotyczących terminowości, kosztów realizacji oraz niezaspokojonych potrzeb klienta.

Zastosowanie modeli oceny procesu wytwórczego obiektywizuje proces oceny, wskazując w sposób znormalizowany obszary procesów podlegające ocenie oraz poziomy dojrzałości. Osiągnięcie danego poziomu dojrzałości wiąże się z oceną poziomu wdrożenia praktyk, realizacji założonych celów czy też poziomu zdolności. Istotne jest zderzenie wymaganych ewidencji – dowodów dla wdrożonych praktyk z tymi faktycznie wdrożonymi w organizacji. Takim dowodem może być chociażby tworzenie Struktury Podziału Prac (WBS) w procesie szacowania zakresu prac w projekcie czy raportów z przeglądów planów projektowych w procesie uzgodnienia poziomu pracy i zasobów. Zazwyczaj bowiem dopiero brak zidentyfikowania takich artefaktów pozwala na realną oceną kompetencji naszej organizacji.

Oba wspomniane wcześniej modele oceny pozwalają na weryfikację poziomu dojrzałości całego procesu tworzenia oprogramowania, a także zdolności w ramach każdego z podejmowanych działań, m.in. takich jak identyfikacja wymagań, projektowanie czy zarządzanie konfiguracją. Na tej podstawie możliwe jest stworzenie profilu zdolności dla każdego z procesów, a w konsekwencji wybór tych z nich, które powinny podlegać optymalizacji, wraz ze skorelowaniem inicjatyw optymalizacyjnych tak, aby kolejność ich realizacji w najwyższym stopniu odpowiadała celom biznesowym organizacji.

Wielokrotnie u źródeł wszystkich problemów leży jakość prac programistycznych. Łamanie podstawowych zasad utrzymywania estetyki oraz struktury kodu źródłowego negatywnie wpływa na utrzymanie oprogramowania, jego aktualizację i możliwości rozbudowy. Brak adekwatnego udokumentowania kodu, w postaci czytelnych komentarzy, nie mówiąc już o testach automatycznych, nie jest niestety niczym wyjątkowym.

Uzasadnieniem jest najczęściej presja na krótkoterminowe efekty lub brak kompetencji. Próby porządkowania kodów źródłowych post factum zazwyczaj kończą się fiaskiem.

Oczywiście zasadne jest, aby punktem odniesienia do całościowej oceny procesu wytwórczego były również inne dobre praktyki i obowiązujące na rynku normy i standardy, metodyka wytwórcza RUP, metodyki zarządzania projektami, a także możliwe do wykorzystania wsparcie narzędziowe, obejmujące zintegrowane środowisko analityczne, środowisko do zarządzania projektami i kolaboracji oraz zintegrowane środowisko deweloperskie.

Michał Wiatr, CISA, TOGAF, COBIT, prezes zarządu firmy Softtutor Consulting

Softtutor Consulting wspiera organizacje w realizacji kompleksowych przedsięwzięć IT, w tym zwiększaniu efektywności procesów wytwórczych oprogramowania.