

SCRUM W POLSCE: AENIMA

Scrum w Polsce

Celem programu “Scrum w Polsce” jest przekazanie **realnych doświadczeń** zespołów stosujących Scrum i powiązane metody z rodziny Agile innym firmom, które rozważają jego zastosowanie lub już rozpoczęły swoje wdrożenia.

O Aenima

Aenima to założona w 2008 roku firma tworząca aplikacje internetowe (głównie w technologii Ruby on Rails) oraz aplikacje mobilne dla polskich i zagranicznych klientów. Na tym rynku z powodzeniem konkuruje z kilkoma większymi oraz setkami mniejszych usługodawców.

Aenima realizuje jednocześnie wiele małych projektów dla różnych klientów: typowy projekt opiera się o pracę jednej lub dwóch osób i trwa maksymalnie 3 miesiące. Oprócz tego realizowane są też większe projekty: trwające nawet ponad rok i angażujące, w zależności od fazy, od 4 do 8 osób.

Aenima we wszystkich projektach stara się wykorzystywać **praktyki Agile**: część projektów realizowana jest w pełnej zgodzie z metodyką Scrum, część korzysta jedynie z wybranych praktyk zwinnych, a część zorganizowana jest pod bezpośrednią kontrolą klienta według jego standardów. Stale są też prowadzone eksperymenty z nowymi praktykami i te, które się sprawdzają są dołączane do standardowego sposobu pracy – w ten sposób organizacja Aenima ciągle ewoluuje.

Dobra praktyka: Niezależnie od konkretnego zestawu metod, narzędzi i praktyk stosowanych w danym projekcie prowadzący Aenimę Marek i Robert dbają o podstawowe **wartości** i fundamentalne **zasady** agile (częsta inspekcja i adaptacja, zaangażowanie klienta, itd.). W większości przypadków spotyka się to z dużym zadowoleniem klienta, gdy doświadczy on korzyści, jakie daje mu zwinna praca. Od czasu do czasu reakcją klienta jest niezrozumienie – wtedy płynna współpraca jest większym wyzwaniem, a projekty są

Co to jest Scrum?

Scrum to **metoda organizacji pracy zespołów** tworzących nowe rozwiązania (informatycznych i nie tylko).

Jako **adaptacyjna** metoda zarządzania Scrum działa szczególnie dobrze tam, gdzie trudno jest z góry przewidzieć wszystkie szczegóły projektu: od prawdziwych potrzeb klienta po nieuniknione problemy techniczne.

Więcej informacji o Scrum w pasku bocznym poniżej oraz w [dziale Artykuły na stronie FluidCircle.net](#).

Jak działa Scrum?

Scrum to szkielet procesu ustalający **role, wydarzenia i narzędzia** które, uzupełnione o odpowiednie praktyki techniczne, pozwalają zbudować efektywny proces wytwórczy.

W Scrum *interdyscyplinarny, samoorganizujący się Zespół* w rytmie **Sprintów** (odcinków czasu o stałej długości – od 1 do 4 tygodni) dostarcza kolejne przybliżenia **Produktu** zgodnie z wymaganiami określonymi przez **Właściciela Produktu**.

Nad prawidłowym przebiegiem procesy dba **Scrum Master**, którego zadaniem jest też aktywne usuwanie przeszkód spowalniających pracę Zespołu.

(kontynuacja na następnej stronie)

trudniejsze w realizacji.

Małe projekty

Jednoosobowe projekty są mniej atrakcyjne z punktu widzenia firmy: w takich przypadkach brakuje członków zespołu, którzy mogliby się nawzajem uzupełniać – całość projektu spoczywa na barkach jednej osoby.

Dobra praktyka: Realizując portfolio małych projektów łatwo też o nadmierną fragmentację i rozszynchronizowanie metod i standardów pracy. Aby zminimalizować ten efekt Aenima stosuje wspólne, okresowe spotkania **Coding Dojo**, na których regularnie ćwiczone są najlepsze praktyki developerskie.

W małych projektach tablica scrum'owa (**Scrum Board**) ma bardzo prostą formę i składa się z trzech kolumn: gotowe-do-realizacji, w trakcie-realizacji i zaakceptowane. W takich projektach całość prac planowana jest podczas jednej sesji a ewentualne poprawki do planu wprowadzane są na bieżąco. Stan projektu śledzony jest za pomocą **wykresu burndown**, dzięki któremu wyraźnie widoczne są postępy (lub ich brak).

Dobra praktyka: Podczas akceptacji poszczególnych funkcji przez Klienta często zdarzają się rozbieżności zdań, ale paradoksalnie jest to traktowane jako oznaka zdrowego projektu: jeśli klient akceptuje wszystkie produkty bez żadnych uwag jest to sygnał, że nie są dla niego zbyt ważne. Właściwie kontrolowany, **kreatywny konflikt** jest wartościowym narzędziem, dzięki któremu finalny produkt jest znacznie lepiej dopasowany do prawdziwych potrzeb Klienta.

Duże projekty

Dobra praktyka: W większych projektach Aenima łączy Scrum z **Kanban**'em czyli tablica, na której śledzona jest praca (Scrum Board lub w tym przypadku Kanban Board) ma kilka dodatkowych cech:

- 1) wszystkie elementy pracy muszą być **zwizualizowane**, a struktura tablicy musi odzwierciedlać rzeczywisty proces,
- 2) stosowany jest twardy **limit liczby zadań** w poszczególnych kolumnach – jeśli poprzednie zadania nie są zakończone, nowe nie mogą być rozpoczęte,
- 3) organizacja pracy optymalizowana jest pod kątem jak **najszybszego przepływu zadań** przez całą tablicę.

Zastosowanie Kanban ma szereg zalet. Np.:

★wszystkie problemy wstrzymujące dane zadanie są

(kontynuacja z poprzedniej strony)

W trakcie sprintu zespół na bieżąco dopasowuje plan działania podczas codziennych, krótkich **Spotkań Scrum**.

Po upływie ustalonego czasu odbywa się **Przegląd Sprintu**, na którym Zespół prezentuje Właścicielowi Produktu właśnie zakończony fragment, realizujący uzgodnione wcześniej funkcje oraz spełniający ustaloną **Definicję Zakończenia Prac** (*definition of done*).

Przed rozpoczęciem kolejnego sprintu Właściciel Produktu weryfikuje przyjęte wcześniej założenia na podstawie rzeczywistego, działającego fragmentu Produktu i na tej podstawie aktualizuje **Backlog** zawierający wymagania co do dalszych prac.

Z kolej zespoł, we własnym gronie, przeprowadza **Retrospekcję**, której celem jest ulepszenie aktualnie stosowanych praktyk oraz poprawę warunków pracy.

Scrum Board to fizyczna tablica, na której zespół na bieżąco śledzi postępy własnych prac.

Struktura tablicy Scrum jest zawsze dopasowana do potrzeb zespołu.



rozwiązywane natychmiast – w razie potrzeby przez cały zespół – i nie mają okazji rozrosnąć się do większych kłopotów

★stały rytm prac sprzyja utrzymaniu wysokiej jakości i ułatwia uniknięcie wypalenia wynikającego z nierównomiernego obciążenia poszczególnych pracowników czy zespołów.

Kolumny na tablicy odzwierciedlające kroki stosowane przez Aenima w większych projektach to zwykle: Ready, Bugs, Dev, Review, Accept, Deployed: (Staging, Production) oraz dodatkowo Backlog i Archive.

Planowanie prac

Planowanie w Aenima jest luźno ograniczone w czasie do 4~5h i polega na tworzeniu i omawianiu *ticketów*. Omówienie polega na wyjaśnieniu przez Właściciela Produktu, na czym ma polegać dana funkcjonalność oraz jaka jest jej wartość biznesowa. Omawiane są też przypadki brzegowe i inne kryteria akceptacji, a następnie zespół szacuje złożoność/pracochłonność zadania za pomocą metody **Planning Poker**.

Dla wszystkich bardziej złożonych funkcjonalności zespół przygotowuje projekt implementacji: rozstrzyga od razu wszystkie ważne decyzje architektoniczne, analizuje główne algorytmy, które zostaną zastosowane itd.

Planowanie realizacji złożonych funkcjonalności obowiązkowo obejmuje projektowanie rozwiązania - pominięcie projektowania powodowało w przeszłości kłopoty podczas samego sprintu. Dodatkowo wspólne projektowanie zwiększa zbiorową świadomość projektu, poczucie współodpowiedzialności wszystkich członków zespołu za produkt oraz sprzyja sprawnemu rozwiązywaniu ew. konfliktów – temat jest wątkowany aż do osiągnięcia konsensusu. Jeśli jego osiągnięcie w zadanym czasie nie jest możliwe, to rozstrzygnięcie konfliktu następuje w ciągu najbliższych dni po planowaniu.

Pair programming (**Dobra praktyka**)

Aenima podjęła trzy próby zastosowania programowania w parach jako stałej części procesu twórczego, z czego trzecia próba przyniosła jednoznacznie pozytywne rezultaty i od tego czasu pair programming jest często stosowany, gdy tworzony jest szczególnie trudny fragment kodu lub gdy zachodzi potrzeba wyrównania wiedzy między członkami zespołu.

Doświadczenie Aenimy pokazuje, że z jednej strony do programowania w parach niezbędny jest pewien poziom dojrzałości i samodyscypliny wśród członków zespołu, a z drugiej jest to świetne narzędzie naświetlające problemy, które inaczej mogłyby pozostać ukryte. Mówiąc dosadnie: pracując w parze trudno marnować czas na surfowanie, gdy pracuje się na jednym ekranie z partnerem.

Dla sukcesu programowania w parach konieczne jest by para pracowała przy jednej stacji roboczej. Dobrze, jeśli jest ona wyposażona w oddzielną klawiaturę i myszkę dla obu pracujących, ale najważniejsze, by nie pozwalała na oddzielenie działań obu osób – taka możliwość zwykle prowadzi do rozpadu wspólnej koncentracji nad zadaniem.

Najważniejsze **korzyści** ze stosowania Scrum w Aenima:

- ★ zadowolenie klientów dzięki ich zaangażowaniu w proces twórczy i dopasowanie produktów do ich realnych potrzeb,
- ★ wgląd w realny stan projektu – szansa na szybkie zaadresowanie problemów
- ★ stały rozwój efektywności zespołu i całej firmy

Pracując w parze dobrze jest stosować odpowiednie praktyki techniczne: np. znany z TDD (test driven development) proces red-green-refactor w modelu ping-pong polegającym na cyklicznej zamianie ról: pierwszy programista pisze test (którego system jeszcze nie spełnia – status czerwony), drugi tworzy właściwą funkcjonalność pozwalającą zaspokoić test (status zielony), a pierwszy następnie refakturuje powstały kod.

Stała gotowość do wdrożenia **(Najlepsza praktyka)**

Jedną z praktyk, która ma największy wpływ na jakość projektów w Aenima jest dbanie o to, żeby **każda** wersja produktu była **gotowa do wdrożenia**.

Integracja całego produktu odbywa się kilka razy dziennie i obejmuje instalację systemu w środowisku testowym i uruchomienie baterii zautomatyzowanych testów. Wdrożenia na środowisko produkcyjne odbywają się średnio raz w tygodniu – rano, w środku tygodnia, aby szybko reagować na ew. problemy. Zespoły w Aenima dokładają starań, by nawet te wersje, które nie będą wdrażane, nie zawierały widocznych dla użytkownika, niedokończonych lub błędnych funkcji.

Aby to osiągnąć bez zakłócania pracy, niegotowe funkcjonalności są ukrywane za pomocą zwykłej konstrukcji `if (aktywneFunkcje[funkcja]) {<implementacja funkcjonalności>}`. Dzięki temu można łatwo aktywować i ukrywać poszczególne funkcje za pomocą wpisów w tablicy `aktywneFunkcje` i zawsze pracować na pojedynczej, zawsze poprawnie działającej wersji systemu.

Korzyści z takiego podejścia to m.in.:

- ★szybsze wykrywanie problemów – uniknięcie długich okresów rozwoju danej funkcji w bocznej gałęzi bez konfrontacji z resztą systemu,
- ★łatwiejsze zarządzanie wersjami,
- ★zaskakująco mały narzut – zwykle do ukrycia funkcji wystarczy zaledwie kilka `if`ów.

Uwaga! Praktykę tę łatwo zlekceważyć ze względu na jej prostotę, ale okazuje się być ona źródłem ogromnego wzrostu efektywności prac. Wzrostu porównywalnego z zastosowaniem Kanban jako dodatku do bazowego procesu Scrum.

Inne praktyki stosowane przez Aenima to m.in.:

★**Czeklista kryteriów jakości do przeglądu kodu** – przydatna by zastymulować głębszą refleksję nad przeglądany kodem i by przeglądającemu trudniej było prześlizgiwać się po jego powierzchni. Po kilku użyciach lista wchodzi w krew przeglądającym. Lista jest też okresowo aktualizowana gdy zespół odkryje nowe, przydatne kryteria jakości lub w inny sposób podniesie swe kompetencje.

★**Technika Pomodoro** – technika polegająca na pracy w stałych blokach (np. 25min) podzielonych obowiązkowymi krótkimi przerwami po każdym bloku i dłuższą przerwą po czterech blokach. Dzięki tej technice Aenima unika spadku realnej efektywności jaki często towarzyszy długim, nieprzerwanym sesjom pracy. Więcej informacji o technice pomodoro można znaleźć pod adresem: <http://www.pomodorotechnique.com/>

Scrum jest częścią ruchu *agile*, którego wartości najlepiej ujęte zostały w opublikowanym w 2001 roku **Manifeście Agile**, w którym autorzy deklarują, że:

1. **Ludzie i ich współdziałanie**
jest ważniejsze niż procedury i narzędzia.
2. **Działające oprogramowanie**
jest ważniejsze niż wyczerpująca dokumentacja.
3. **Współpraca z klientem**
jest ważniejsza niż negocjacja kontraktów.
4. **Reagowanie na zmiany**
jest ważniejsze niż realizacja planów.

Autorzy manifestu w pełni doceniają to, co wymieniono po prawej, jednak wyżej cenią to, co zostało wymienione po lewej.

★**Coding Dojo** – to wspomniane wyżej spotkania, na których członkowie wszystkich zespołów wspólnie ćwiczą technikę programowania na specjalnie przygotowanych ćwiczeniach. Na takich spotkaniach wszyscy patrzą na jeden ekran (wyświetlony za pomocą projektora) a kontrola nad klawiaturą przekazywana jest kolejnym osobom. Dodatkową zaletą takich spotkań jest trening uważności i koncentracji – każdy uczestnik powinien być w dowolnym momencie gotowy kontynuować pracę. Nawyk ten ta przydatny jest potem np. podczas programowania w parach.

★**Unikanie wielozadaniowości** – zgodnie z opisanymi wyżej zasadami Kanban zespół nie może rozpocząć zbyt dużo zadań jednocześnie. Na limit akceptowalnej liczby współbieżnych zadań wpływa m.in. ich średni rozmiar: jeśli projekt składa się z dużej liczby małych zadań to limit pracy w toku może być nieco większy, jeśli zaś zadania są duże i skomplikowane to większy nacisk warto położyć na domknięcie poprzedniego zadania przed rozpoczęciem następnego.

★ **Współwłasność kodu** – cały zespół jest *współodpowiedzialny* za całość produktu. Aenima stara się unikać sytuacji, w których pojedyncza osoba jest jedyną, która dobrze rozumie wybrany fragment kodu. Wiedza zawsze powinna być mniej lub bardziej współdzielona przez cały zespół. Czasem cel ten jest podważany przez Klientów żądających możliwości mikrozarządzania pracą poszczególnych programistów, lecz Aenima dba by przypadki te były minimalizowane.

★**Retrospekcje** – okresowe spotkania pozwalają każdemu członkowi zespołu oraz zespołowi jako całość odpowiedzieć na pytania: Co udało Ci się nauczyć od ostatniej retrospekcji: pod kątem organizacji pracy? pod kątem wiedzy i umiejętności technicznych? Spotkania te pozwalają zespołowi oderwać się od bieżących problemów i spojrzeć na efektywność pracy z dalszej perspektywy. Udane retrospekcje są głównym motorem ciągłego doskonalenia całego zespołu.

★**Rotacyjny Scrum Master** – rola Scrum Mastera w większych zespołach jest przekazywana między jego członkami. Z nielicznymi wyjątkami każdy członek zespołu powinien przez jakiś czas pełnić taką rolę, by lepiej zrozumieć dynamikę całego zespołu oraz potrzeby innych jego członków jak też uwarunkowania istniejące w otoczeniu zespołu.

★**Zero tolerancji dla wykrytych błędów** – każdemu wykrytemu błędowi przypisywany jest priorytet oznaczony kolorem czerwonym lub żółtym. Czerwone błędy naprawiane są natychmiast. Jeśli jest to konieczne, któryś z członków zespołu przerywa aktualne zadanie i naprawia błąd. Żółte błędy naprawiane są, gdy tylko którykolwiek z członków zespołu zakończy aktualne zadanie. Zespół nie może rozpocząć nowego zadania przed naprawieniem błędu. Jeśli zgłoszony błąd oceniony jest jako niewart naprawiania to nie jest w ogóle śledzony.

Umowy

Preferowanym modelem rozliczania się z klientami jest model Time & Materials. Czasem realizowane są też projekty ze z góry określonym zakresem, ale często kończą się one źle, gdy Klienci próbują nadużywać nieprecyzyjne sformułowania stanowiące treść umowy.

Czasem też Aenima realizuje i dostarcza cały uzgodniony zakres

Korzenie Scrum sięgają prac Williama Deminga, które pomogły Japonii, zdruzgotanej po Drugiej Wojnie Światowej, stać się światową potęgą przemysłową.

Teorie Deminga stały u podstaw systemu zarządzania Toyoty oraz całego ruchu *Lean Manufacturing* – znanych z wysokiej jakości i produktywności.

Idee stosowane w przemyśle przeniesli na grunt inżynierii oprogramowania twórcy Scrum: Ken Schwaber i Jeff Sutherland, którzy po raz pierwszy opublikowali swoją metodę w 1995 roku.

Obecnie idee zawarte w Scrum są już silnie zakorzenione w informatyce i zaczynają rozprzestrzeniać się także na inne obszary gospodarki, m.in. dzięki pracom Steve'a Denning'a pod hasłem Radical Management.

projektu, ale Klient nie jest zadowolony, ponieważ nie był zaangażowany w proces i nie miał okazji dopasować wstępnych wymagań do swoich prawdziwych potrzeb.

Wielu klientów na polskim rynku preferuje projekty z ustaloną ceną i zakresem – Aenima celowo rezygnuje z konkurencji o tych klientów, koncentrując się na tych, z którymi może zbudować relacje oparte o zaufanie pozwalające na swobodną współpracę w modelu Time & Materials. Relacje te tworzone są przez pozytywne referencje od innych klientów, długotrwałą współpracę oraz fakt, że Klient jest właścicielem kodu i może w każdej chwili zerwać współpracę i przekazać projekt innemu wykonawcy. Niektórzy klienci stosują też własny przegląd kodu by na bieżąco monitorować jakość odbieranych produktów.

Ciekawostka: Aenima stara się zaczynać projekt niedługo przed końcem miesiąca by mieć okazję szybko wystawić fakturę "rozgrzewkową" na małą kwotę. Dzięki temu ew. problemy z realizacją płatności przez danego klienta mogą być wykryte i zaadresowane już na początku relacji, gdy kwoty – a co za tym idzie poziom emocji – są jeszcze niskie.

Podsumowanie

Aenima od początku swojego istnienia stosuje Scrum i inne zwinne praktyki. Dzięki temu może z sukcesem konkutować z innymi dostawcami stosującymi mniej efektywne metody organizacji pracy.

Szczególnie przydatnymi praktykami, rozszerzającymi podstawowy szkielet Scrum są **Kanban** – pozwalający eliminować większość problemów zanim urosną do kłopotliwych rozmiarów oraz **Stała Gotowość do Wdrożenia**, która również pozwala zminimalizować liczbę problemów wykrywanych zbyt późno by poradzić sobie z nimi bez poważniejszego zakłócenia projektu.

Obie te praktyki są realizacją podstawowej zasady *agile* polegającej na **jak najszybszej weryfikacji założeń** i częstej kontroli realnej jakości i wartości produktów.

Wywiadu będącego podstawą przypadku udzielił współzałożyciel Aenima, Agile Manager/Scrum Master Marek Kirejczyk.

*Pierwsza publikacja: kwiecień 2012
Ostatnia aktualizacja: maj 2012*

Więcej o Aenima można dowiedzieć się ze strony: <http://aenima.pl/>.

Więcej przypadków oraz sumaryczne informacje o programie Scrum w Polsce można znaleźć pod adresem: <http://fluidcircle.net/case>.