Możesz być zdezorientowany, dlaczego te same aspekty są stale wymieniane zarówno jako pozytywy, jak i negatywy podczas podsumowania na końcu każdego artykułu o Scrumie, KanBanie i Scrum-Banie. Jest w tym pewien cel, a tym celem jest zawsze rozważanie “to zależy” jako najlepszej możliwej pierwszej odpowiedzi na każde pytanie.

Scrum vs Agile vs Kanban czy Scrumban? – Krótkie spojrzenie na trwającą transformację Agile

Wiele osób łatwo gubi się w znaczeniu terminów związanych z procesami Agile. Dlatego zanim przejdziemy dalej, wyjaśnijmy sobie terminologię. Agile to nazwa metodologii charakteryzującej się podziałem pracy na etapy, w ramach których dostosowujemy ją do zmieniających się wymagań. Idea stojąca za Agile jest przekształcana w rzeczywistość w różnych ramach. Przykładami takich frameworków są Scrum, Kanban i Scrumban. Tak więc technicznie rzecz biorąc, takie porównania jak Scrum vs Agile vs Kanban nie powinny istnieć.

Dlaczego istnieje tak wiele frameworków? Najlepszą odpowiedzią na to pytanie byłoby prawdopodobnie: ponieważ istnieje wiele sposobów, w jakie ludzie lubią pracować. Agile wierzy w priorytet efektywności. Dlatego transformacja Agile jest procesem ciągłym. Jest jeden wspólny cel wszystkich frameworków. Prowadzą one do opracowania najlepszego możliwego produktu dla klienta. Zilustrujmy ten cel, przechodząc kolejno przez zasady Scrum, Kanban i Scrumban.

>

Który framework Agile wybrać?

Który Agile Framework jest NAJLEPSZY (i dlaczego nie można

odpowiedzieć na to pytanie)?

Odpowiedź: “To zależy od: “

1) Co próbujesz osiągnąć poprzez zastosowanie określonego frameworka?

1) Co próbujesz osiągnąć poprzez zastosowanie określonego frameworka?

Może po prostu potrzebujesz kolejnej rundy “zmian organizacyjnych”, które usprawiedliwią obecny niezbyt dobry stan danego produktu? Mam nadzieję, że nie. O wiele lepiej będzie w pełni zaangażować się w określony framework po dokładnej analizie (proszę zaangażuj swoich programistów w te analizy!) & test na żywo w istniejącym projekcie.

2) Jaka jest obecna struktura rozwoju Twojego produktu?

Możliwe, że już rozwijasz produkt w mniej lub bardziej zwinny sposób ale nawet nie pomyślałeś o nazwaniu go w ten sposób. Jeśli jest tylko kilka rzeczy, które trzeba zmienić, aby dostosować

obecną produkcję/przepływ pracy do na przykład KanBan, po co iść i wymuszać na przykład Scrum?

Im mniej zmian wymusisz na całej organizacji na raz tym lepiej.

3) Jaki jest Twój produkt i dlaczego w ogóle chcesz rozważyć Agile jako wartościowe podejście do rozwoju tego produktu?

>

Jak wspomniano na samym początku, istnieją produkty (na przykład most), które muszą być rozwijane w podejściu kaskadowym. Po prostu nie odniesiesz korzyści z zastosowania frameworków Agile do rozwoju mostu. Może Twój produkt nie jest dokładnie mostem, ale podlega tej samej zasadzie? Jeśli tak, to wysiłek organizacyjny mający na celu narzucenie “Agile” będzie oczywistym marnotrawstwem.

>

4) Czy masz wizję swojego produktu ?

Niemal niemożliwe jest wyobrażenie sobie produktu, który nie skorzysta z jasno określonego planu rozwoju na przyszłość. Nawet jeśli te plany się zmienią (a prawda jest taka, że zmienią się bardzo często), tworzysz powszechne poczucie w całej firmie, że jako właściciel produktu/CEO/kierownik produktu/kierownik projektu (wybierz nazwę, którą chcesz) dbasz o produkt.

Jest to bardzo ważny sygnał dla:

- **deweloperów – daje im to cel, aby skupić się na tym, w czym są najlepsi – rozwoju! Jeśli zobaczą, że masz jasny cel produktowy i plan jego dostarczenia, pójdą za tobą!**
- **przyszli pracownicy – czy tego chcesz, czy nie, obecni członkowie zespołu dzielą się opiniami na temat Twojej firmy/zespołu przez cały czas. Stworzenie wizji produktu to po części także dobry PR na rynku pracy.**
- **obecni i przyszli partnerzy biznesowi – jeśli jasno wyrazisz wizję swojego produktu, będziesz w stanie utrzymać te partnerstwa, które są cenne dla Twojej organizacji.**

5) Czy jesteś pewien, że kultura Twojej organizacji będzie zgodna z zasadami Agile

Jak wspomniano kilka razy w tym dokumencie, zmiana kultury całej organizacji jest bardzo trudna. Wdrożenie jakiegokolwiek zwinnego frameworka nie ma sensu, jeśli organizacja nie jest już przynajmniej trochę przejrzysta i gotowa do inspekcji i adaptacji (a więc gotowa zaakceptować porażkę jako naturalny wynik zmiany „jak pracujemy”). Nawet najlepszy możliwy produkt do wdrożenia zwinnego frameworka nie przyniesie korzyści bez odpowiednich wartości jako fundamentów.

Jeśli prowadzisz małą firmę (lub jeden zespół lub dział zespołów wewnątrz większej firmy), masz największy wpływ na narzucenie zmian kulturowych poprzez jasne wyrażenie przez siebie wszystkich tych wartości, które chcesz narzucić członkom swojego zespołu. Jeśli to się uda i znajdziesz członków zespołu chętnych do przestrzegania tych wartości (czasami będziesz musiał zwolnić kogoś, kto nie chce się dostosować) ORAZ wartości te są zgodne z wartościami Agile’, powinieneś spróbować użyć

zwinnego frameworka. Wtedy jest prawie gwarantowane, że twój zespół/mała firma skorzysta na użyciu określonego frameworka, a tym samym będzie się rozwijać i wzbudzać ciekawość na rynku/wewnątrz większej firmy.

>

W tym momencie jesteś:

- **prowadzisz małą firmę i właśnie masz trudne zadanie utrzymać tę “właściwą kulturę” przez okres bolesnego rozwoju firmy;**
- **prowadzenie zespołu lub oddziału w większej firmie i masz bardzo trudne zadanie przekonania kierownictwa wyższego szczebla/CEO/ zarządu do zastosowania kultury twojego zespołu’w całej firmie.**

Co to jest Scrum?

Jak wspomniano wcześniej, Scrum jest frameworkiem używanym do wdrażania metodologii Agile. Główną cechą tego frameworka jest podział projektu na ograniczone czasowo przyrosty, zwane Sprintami. Sprint trwa

do 4 tygodni i kończy się dostarczeniem wcześniej uzgodnionej części projektu. Głównym celem frameworka Scrum jest ciągłe doskonalenie Sprintów, celu i całego procesu rozwoju produktu. Istnieją inne cechy charakterystyczne Scrum, takie jak:

>

- regularne planowanie odbywa się na początku Sprintu
- szacowanie czasu potrzebnego na wykonanie zadań odbywa się przed rozpoczęciem Sprintu

.

- zadania do wykonania w ramach Sprintu powinny być małe, jeśli są większe, powinny zostać poddane dalszemu podziałowi
- wszelkie zmiany zakresu powinny poczekać do następnego Sprintu
- Istnieją cztery główne rodzaje spotkań: Planowanie Sprintu, codzienny Scrum, Przegląd Sprintu i Retrospektywa Sprintu

.

- rozróżnia się konkretne role, takie jak Scrum Master, Product Owner, Development Team
- własność należy do Właściciela Produktu

.

Co to jest Kanban?

Ta struktura również dzieli projekty na małe i łatwe do zarządzania „kawałki”. Czym zatem jest Kanban i dlaczego go potrzebujemy? Czym różni się od Scruma? Kanban nie skupia się na czasie jako najważniejszym czynniku w ukończeniu etapów. W Kanban praca nad projektem przypomina ciągły przepływ. Tak więc, na przykład, praca w ramach przyrostu będzie kontynuowana, dopóki zespół nie poczuje, że dodał znaczącą wartość do końcowego wyniku. To właśnie wtedy zakończą Sprint. Poniżej można znaleźć kilka innych aspektów charakteryzujących Kanban:

- zwykle zarządza się nim za pomocą tablicy Kanban, która pomaga zobaczyć, jak zadania przechodzą przez przepływ pracy

- planowanie odbywa się na żądanie – nie ma precyzyjnej procedury planowania

.

- szacowanie czasu jest opcjonalne – zwykle zespół rozpoczyna nowe zadanie od *listy rzeczy do zrobienia gdy jest na to miejsce: 400;”> gdy jest na nie miejsce w ramach pozycji na liście w toku*

.

- liczba zadań oznaczonych jako *w toku* jest ograniczona

.

- zmiany zakresu prac są dodawane w razie potrzeby
- role są przypisywane w razie potrzeby
- nie są wymagane żadne spotkania
- własność zależy od zdefiniowanych ról i potrzeb.

Co to jest Scrumban?

Jak można się spodziewać, Scrumban jest połączeniem zasad Scrum i Kanban. Znajduje się

gdzieś pomiędzy sztywnym frameworkiem Scrum a zbyt elastyczną strukturą Kanban. Jak to zatem wygląda? W Scrumban:

- podejście Scrum jest używane do planowania, przydzielania i priorytetyzacji zadań
- tablica Kanban służy do lepszej wizualizacji pracy i postępów

.

- planowanie odbywa się wtedy, gdy jest potrzebne
- szacowanie czasu jest opcjonalne

.

- zmiany zakresu prac są dodawane w razie potrzeby
- role są następujące: Scrum Master, Product Owner, Development Team

.

- prowadzone są codzienne spotkania Scrum; zwykle dodawane są również inne spotkania Scrum – w zależności od potrzeb
- własność należy do Właściciela Produktu, ale może

również zależeć od zdefiniowanych ról.

Jak zatem określić, którego frameworka/narzędzia powinienem użyć?

Wiedząc, że nie ma jednoznacznej odpowiedzi na pytanie “który framework jest najlepszy”, musimy zadać sobie inne pytanie “Który framework/narzędzie najlepiej pasuje do mojego produktu?”

Ważne podobieństwa Scrum, ScrumBan & KanBan:

- **stosunkowo łatwe do**

wdrożenia, ale trudne do opanowania ze względu na doskonałość jako cel końcowy;

- skupiają się na redukcji marnotrawstwa, ciągłym dostarczaniu wartości i tworzeniu pętli sprzężenia zwrotnego;
- wymagają, aby zespół programistów był wielofunkcyjny;
- wymaganie odpowiedniej kultury organizacyjnej, aby przynieść jakiekolwiek korzyści;

Co może być trudne przy

wyborze odpowiedniego frameworka/narzędzia:

- **Niektóre aspekty każdego frameworka/narzędzia mogą przynieść korzyści lub zaszkodzić procesowi rozwoju produktu, więc wybieraj mądrze;**
- **zachowaj ostrożność przy wdrażaniu odgórnie jednego frameworka/narzędzia do wszystkich procesów związanych z zarządzaniem produktem (dział wsparcia może potrzebować innego podejścia niż dział R&D).**

- **Jeśli Twój produkt jest duży i skomplikowany, prawie zawsze lepiej jest dopasować dany framework/narzędzie oddolnie ;**

  .

- **Zawsze zbieraj informacje zwrotne od deweloperów na temat kierunku bieżących zmian podczas zwinnej transformacji;**
- **pamiętaj, że charakter Twojego produktu może zmieniać się w czasie, więc żadne rozwiązanie nie będzie**

wieczne;

Różnice i prawdopodobne najlepsze zastosowania

Scrum	Scrum-Ban	KanBan
Framework	Ramework	Narzędzie (może być używane jako część dowolnego frameworka lub niezależnie)
Definiuje role, obowiązki, artefakty i zdarzenia	Definiuje role, obowiązki, artefakty i wydarzenia, ale można je zmienić/wyciąć, jeśli nie przynoszą wystarczającej wartości	Dopasowanie do bieżącego przepływu pracy (bez zmian w rolach, zdarzeniach itp.) – tylko tablica KanBan
Czas w ramce	Nie musi być ograniczone czasowo	Bez ograniczeń czasowych
Nienakładanie limitów systemu pull & WiP	Nałożenie limitów systemu ciągnięcia i WiP	Nałożenie limitów pull-system & WiP
Możliwe najlepsze zastosowania		

Rozwój i utrzymanie produktów, które korzystają z **empiryzmu** (nie ma sensu używać tych frameworków, jeśli dokładnie wiemy, co będzie wynikiem naszej pracy i / lub nie jesteśmy w stanie zmienić produktu po otrzymaniu informacji zwrotnej od klienta końcowego)	Może być używany **albo** jako narzędzie do a) rozwijania produktów, które korzystają z empirii (rozwój oprogramowania) b) czystego usuwania marnotrawstwa i defektów (produkcja samochodów itp.) c) czysto płynnego przepływu elementów pracy poprzez wizualizację i rozwiązywanie wąskich gardeł (zespoły utrzymania i wsparcia)	
	Rozwój produktów, które czerpią największe korzyści z ciągłego dostarczania (brak wyraźnych terminów, wydanie, gdy jest gotowe)	
Rozwój produktów, które są w stanie chaosu z powodu częstych zmian wymagań klienta i braku jasnych definicji odpowiedzialności	Produkty, które są obecnie w stagnacji rozwojowej z powodu nadmiernej analizy przyszłych potencjalnych elementów pracy i braku koncentracji na dostarczaniu wartości	Produkty, które zostały opracowane przy użyciu frameworków Agile z rosnącym zrozumieniem koncepcji LEAN w całym zespole/organizacji
Rozwój produktów bez historii używania zwinnych frameworków	Rozwój produktów z pewną historią używania frameworka Scrum	Produkty, które zostały opracowane przy użyciu zwinnych frameworków z rosnącym zrozumieniem koncepcji LEAN w całym zespole/organizacji
Rozwój produktów, w których informacja zwrotna od klienta końcowego powinna zostać przekształcona w przyrost produktu nie szybciej niż po 1 sprincie	Rozwój produktów, w których informacje zwrotne od klienta końcowego powinny być przekształcane w przyrost produktu nie szybciej niż po 1 sprincie	Rozwój produktów, w których informacja zwrotna od klienta końcowego musi zostać przekształcona w przyrost produktu w jak najkrótszym czasie

Scrum vs Agile vs Kanban vs Scrumban – z czego powinieneś skorzystać?

Jak zapewne podejrzewasz, nie ma idealnego frameworka. Jednocześnie nie ma złego frameworka. To, który powinieneś wybrać do zarządzania projektem, zależy od okoliczności. Dlatego przygotowałem krótki przewodnik po typach sytuacji, w których najlepiej jest użyć

konkretnego frameworka:

- Używaj Scrum, gdy twój projekt wymaga szybkiej dostawy początkowej i szybkiego postępu (duże, długoterminowe projekty)
- Używaj Kanban w ciągłym wytwarzaniu produktu, na przykład podczas świadczenia usług, wsparcia, konserwacji, naprawiania błędów
- Używaj Scrumban w złożonych projektach, które mają funkcje produktu i wsparcia (startupy, projekty o szybkim tempie)

Wybierając framework, należy

jednak wziąć pod uwagę również członków zespołu Agile. Jeśli członkowie zespołu nie lubią ścisłych ram, ich wydajność może ucierpieć w przypadku wdrożenia podejścia Scrum. To samo może się zdarzyć, gdy zdecydujesz się zastosować podejście Kanban w zespole, który nie działa dobrze w nierestrykcyjnych ramach. W takich przypadkach najlepiej jest rozważyć użycie Scrumbana i połączenie elementów Scruma i Kanbana zgodnie z preferencjami zespołu. Agile nadaje priorytet wynikom projektu, ale także

ludziom. Optymalny rezultat nie zostanie osiągnięty, jeśli zespół nie będzie dobrze pracował.

Sailing Byte opanuje transformację Agile i dostarczy Twój projekt w mgnieniu oka!

Sailing Byte wierzy, że wybór Agile jest najlepszym

podejściem do rozwoju produktu. Nie ograniczamy się do Scrum lub Kanban, ale obejmujemy całe spektrum, które przynosi transformacja Agile. W końcu Sailing Byte i metodologie Agile mają wspólny cel,

którym jest dostarczenie naszym klientom najlepszego możliwego produktu. Dlatego zawsze, gdy projekt wymaga podejścia Scrum, używamy go. Ilekroć Kanban jest lepszą metodą, użyjemy jej. Wreszcie, jeśli

SAILINGBYTE

Scrumban pozwala na optymalny rozwój produktu, możesz być pewien, że Sailing Byte go użyje. Wszystko po to, aby urzeczywistnić Twoją wizję w najlepszej możliwej wersji. Zarezerwuj telefon już dziś, aby omówić,

SAILINGBYTE

którą metodologię Agile wykorzystamy w rozwoju Twojego oprogramowania.