Agile, po prostu.

Przewodnik ze sprawdzonymi zasadami na temat zwinnego dostarczania mierzalnej wartości, bez opóźnień i niepotrzebnych stresów.

projectmakers.pl

Spis treści

Rozdział 1

Co jest najważniejsze w zwinności?

Według Manifestu Agile, w zwinnym zarządzaniu projektami kluczowe są 4 aspekty:

1. Ludzie i Interakcje (bardziej niż procesy i narzędzia).
2. Działające Oprogramowanie (bardziej niż obszerna dokumentacja).
3. Współpraca z Klientem (bardziej niż negocjowanie szczegółów umowy).
4. Reagowanie na Zmiany (bardziej niż podążanie za początkowym planem).

Na tej liście jest jeden punkt, który według mnie jest decydujący. Bez niego reszta nie ma znaczenia. To jest...**Działające Oprogramowanie.**

Celem projektu informatycznego jest uzyskanie konkretnego **efektu** poprzez wdrożenie oprogramowania, które działa prawidłowo i spełnia oczekiwania jakościowe. Takim efektem może być na przykład zwiększenie zadowolenia klientów o 25% dzięki dodaniu nowych, intuicyjnych funkcjonalności.

W tym opracowaniu skupiam się więc na **dostarczaniu działających rozwiązań w sposób zwinny.** Pokażę Ci dobre praktyki, które są znane od dziesięcioleci. Bo agile w IT narodził się kilka dekad przed Manifestem z 2001 roku. O początkach agile piszę więcej na końcu.

Będą też historie projektowe. I zaraz właśnie pokażę Ci na przykładzie, dlaczego punkt nr 2 z Manifestu Agile jest tak istotny.

Historia nagle wstrzymanych prac

Jacek właśnie dołączył do trwającego projektu.
Celem jest dokończenie i wdrożenie rozwiązania informatycznego wspierającego wewnętrzne procesy w organizacji.

Zespół jest nieduży, ale świetnie zgrany. Każdy rozumie co jest do zrobienia. Punkt z Manifestu dotyczący ludzi i interakcji jest całkowicie spełniony.

Zespół Jacka współpracuje bezpośrednio z klientem wewnętrznym. Pętla informacji zwrotnej funkcjonuje sprawnie. Programiści regularnie prezentują efekty pracy i wdrażają poprawki. Punkt nr 3 z Manifestu, czyli bliską współpracę z klientem odhaczam jako spełniony.

Zmiana w tym projekcie to codzienność. Klient konsultuje temat z różnymi specjalistami, co powoduje regularne zmiany priorytetów. Wypracowany rytm pracy i organizacja backloga dają sobie z tym radę. Punkt nr 4 też działa.

Pewnie już widzisz, że pominąłem punkt nr 2.
No właśnie, co z **Działającym Oprogramowaniem?**

Oprogramowanie zwyczajnie **nie działa**. To znaczy, można je testować, ale nie można go jeszcze wdrożyć.

Są pewne błędy, których naprawa schodzi na drugi plan, bo wymyślanie i dostarczanie nowych funkcjonalności jest dużo ciekawsze, zarówno dla zespołu jak i dla klienta. “Odłóżmy to do backloga” - to popularna reakcja na skomplikowane problemy.

STOP! Jacek wstrzymuje dodawanie zaplanowanych funkcjonalności. Bez **wdrożenia oprogramowania** i oddania go w ręce rzeczywistych użytkowników klient nie dowie się, czy to, co zaplanował faktycznie działa. Zespół będzie reagował na zmiany, które być może nie są konieczne. Klient w końcu też przestanie być zadowolony ze współpracy, bo nie będzie widział realnych efektów.

Jacek potrzebuje jak najszybciej dostarczyć **Działające Oprogramowanie** na produkcję. To jest jedyny priorytet.

W tej historii dostrzegam, nie po raz pierwszy, bardzo ważną zależność.

Wszelkie zwinne interakcje czy produkty (artefakty) powstające zgodnie z zasadami Scruma są pomocne. Jeżeli jednak ten wysiłek nie przekłada się na jak najszybsze przekazanie działającego produktu użytkownikom, to znaczy, że takie działania wymagają natychmiastowych usprawnień.

Prawda być może jest nieco brutalna. Ostatecznie to korzyść biznesowa mierzona zwykle w PLN liczy się najbardziej. Zlepek losowo dobranych funkcjonalności może, ale nie musi dostarczyć taką wartość.

Najlepszą metodą realizacji projektu jest więc ta, która **doprowadzi do wdrożenia Działającego Oprogramowania**. Oprogramowania, które szybko wygeneruje konkretną wartość. A skutecznym narzędziem, które pomaga zarządzać projektami zwinnymi jest zwykle **zdrowy rozsądek.**

W tym opracowaniu znajdziesz sporo wskazówek bazujących na tym, co ma sens i działa. Wybierz z tego to, co uznasz za wartościowe.

Rozdział 2

Czym jest agile i kiedy go nie stosować? (definicja)

Definicji zwinności jest bardzo wiele.

Zobacz poniższe nagranie, na którym **w 1 minutę** wyjaśniam jak ja rozumiem Agile.

Teoretycznie jest to proste.

Agile najlepiej działa wtedy, **kiedy wiesz CO** (cel) chcesz osiągnąć i potrzebujesz **znaleźć odpowiedź** na pytanie **JAK** (rozwiązanie). Wdrożenie agile polega na **częstym wydawaniu** działającej wersji (inkrementy), przy jednoczesnym **zbieraniu i uwzględnianiu** informacji zwrotnej (iteracje).

W praktyce bywa z tym różnie...

Jak zrobić więcej bałaganu niż pożytku

Kasia dostała do poprowadzenia projekt, którego celem było zbudowanie nowej, **lepszej wersji działającego oprogramowania**, które wspiera kluczową działalność przedsiębiorstwa.

Takich historii są tysiące.

W jej branży funkcjonuje ustalony zestaw zasad i procesów, które każde takie narzędzie musi spełniać. Dopiero po zbudowaniu podstaw może zacząć się etap eksperymentowania i dokładania innowacyjnych rozwiązań.

Kasia znała więc dobrze początkowy zakres prac, który potwierdziła z wieloma decyzyjnymi osobami. Decydenci narzucili też ambitne, ale realistyczne terminy.

Wszystko wyglądało dobrze do czasu. Jakie było bowiem zdziwienie Kasi, gdy okazało się, że zespół wytwórczy chce realizować ten projekt od początku zwinnie. A może raczej trzeba napisać “zwinnie”, bo przecież zakres, czyli droga do osiągnięcia początkowych celów była znana. Projekt ten nie spełniał więc podstawowych kryteriów zwinności.

Kolejne tygodnie to był klasyczny **chaos organizacyjny**. Chociaż wiadomo było, co trzeba zrobić, to jednak Scrum Master nalegał na organizowanie regularnych sesji z całym zespołem, na których godzinami dyskutowano każdy szczegół z zakresu.

Położenie pola na ekranie, nazwa zmiennej czy reguły walidacyjne – nawet takie oczywiste zagadnienia wywoływały burzliwe, niekończące się dyskusje.

Czas leciał, a prace szły do przodu ślimaczym tempem.

Ostatecznie Kasia i jej zespół osiągnęli dokładnie taki sam efekt, jaki był początkowo zaplanowany. Tyle tylko, że wszelkie zbędne rytuały przełożyły się na trzy miesiące opóźnienia.

W historii Kasi odnajduję kilka ciekawych wątków.

1. **Agile nie jest do wszystkiego**. Jeżeli wiadomo, co jest do zrobienia, system jest zaprojektowany, są gotowe makiety ekranów itd., to trzeba to po prostu zrobić.
2. Jeżeli jest kilka dróg do celu i dopuszczane są **zmiany zakresu**, to Agile będzie dobrym rozwiązaniem. O ile został określony mierzalny cel, ale o tym później.
3. Ogólnie, proces wytwarzania i związane z nim aktywności powinny być dostosowane do potrzeb. Jeżeli jakieś działanie nie przynosi efektu, to warto je wyeliminować. Choćby na próbę, żeby sprawdzić, czy coś się zmieni. Pamiętaj, że celem jest **Działające Oprogramowanie**, a nie idealny proces. Np., “daily” może być 2-3 razy w tygodniu, jeżeli częstsze spotkania nic nie zmieniają.
4. **Łączenie podejścia tradycyjnego i zwinnego** to często dobry pomysł! Zwykle znajdziesz takie elementy projektu, które pasują do różnych sposobów wytwarzania. W projektach takich jak u Kasi, przygotowanie pierwszych wdrożeń tradycyjnie i stopniowe przechodzenie w kierunku zwinności może przynieść dobre efekty.

Rozdział 3

Zrób to koniecznie, zanim zaczniesz pierwszy sprint!

Żeby osiągnąć cel stawiany przed każdą zmianą, trzeba najpierw dobrze zrozumieć ten cel. Być może brzmi to, jak kolejny banał, ale temat jest dość złożony. Dobre opisanie celu to prawdziwe wyzwanie!

Wiesz już, że **cel projektu zwinnego** to dużo więcej niż samo wdrożenie funkcjonalności. Celem jest oczekiwana **korzyść** z wdrożenia pewnych funkcji, zwykle w połączeniu z osiągnięciem odpowiednich cech jakościowych. Na przykład, wymiana danych pomiędzy systemami (funkcja) zredukuje ilość pracy manualnej tylko wtedy, gdy będzie działać szybko i stabilnie (cechy jakościowe).

Cel odpowiada więc na **potrzeby biznesowe** i musi być możliwy do zmierzenia. “Większe bezpieczeństwo” to nie jest dobry cel! Przeczytaj poniższe artykuły, w których znajdziesz sporo definicji oraz przykładów. Po przeczytaniu tych materiałów będziesz w stanie zdefiniować **mierzalne cele dowolnego projektu**.
Cel a zakres projektu – 6 konsekwencji gdy pomylisz te pojęcia
Cel projektu IT – 22 praktyczne przykłady
Określnie celu jest kluczowe.
Tak samo ważne jest zebranie i potwierdzenie wszystkich informacji powiązanych z celami, typu:

- na kiedy cel ma zostać zrealizowany?
- jaki jest maksymalny budżet na realizację celu, oraz inne ograniczenia?
- kto jest źródłem szczegółowych informacji na temat celu itd.

Dlaczego to jest ważne?

Lepiej raz coś zapisać, niż dziesięć razy zgadywać

Marek dostał do poprowadzenia swój pierwszy poważny zwinny projekt.

Kiedy w trakcie sesji mentoringowych rozmawiałem z nim o postępach i wyzwaniach, to początkowo słyszałem wyrażenia typu: "*...ale tym obszarem już się ktoś inny zajmuje. To będzie zrobione i nie muszę się tym przejmować.*"

Wtedy zapalała mi się lampka ostrzegawcza i pytałem:
"*Czy masz potwierdzone przez tego, kto Ci zlecił ten projekt, za co dokładnie odpowiadasz i czy faktycznie ten obszar nie wchodzi w Twój zakres obowiązków?*"

Odpowiedź była zwykle "*Nie...*" (bo przecież zwinność itp.)

Z czasem okazywało się, że faktycznie pewne tematy były oficjalnie zaopiekowane przez kogoś innego.

Z drugiej strony, po konsultacjach z przełożonym część nowych zadań i odpowiedzialności jednak pojawiła się na liście Marka. To, że ktoś pomagał, nie oznaczało, że bierze odpowiedzialność za całe zadanie na siebie. Nawet w samoorganizującym się zespole ważne są przypisania do poszczególnych ról i określenie odpowiedzialności.

Dokumenty takie jak Karta Projektu, Plany czy Harmonogramy mogą kojarzyć się z tradycyjnym podejściem do projektów. Są to jednak Twoi sprzymierzeńcy także w podejściu zwinnym. Jakby nie było, to mówimy o **zwinnym Zarządzaniu Projektami**, czyli dobrze znane praktyki z tradycyjnego zarządzania projektami mają też zastosowanie w agile.

Najprostsze z tych narzędzi, czyli Karta Projektu, to jest to coś, co oszczędzi Ci sporo nerwów także w projekcie zwinnym. Ustalenie z wysokiej perspektywy co jest do zrobienia (mierzalne cele!), na kiedy, przez kogo itd. jest kluczowe w każdym projekcie. Dzięki temu będzie jasne, co to oznacza, że oprogramowanie działa i po czym poznasz, że wdrożyłeś je z sukcesem. Aby zwinnie znaleźć odpowiednią drogę prowadzącą do tego sukcesu i na końcu móc świętować kolejny domknięty projekt, to trzeba **zrozumieć, doprecyzować i opisać podstawowe założenia projektu.**

Zobacz krótkie wideo, na którym m.in. pokazuję jak zebrać podstawowe informacje na temat projektu w Confluence.

Lekcja z tej historii jest taka, że "zwinnie" oznacza również "zgodnie z planem". Plany mogą mieć różną formę, może będą mniej formalne i częściej zmieniane, ale powinny istnieć.

Powiedzenie "jesteśmy zwinni" nie powinno być wymówką do pozbawienia się wszelkiej kontroli nad projektem i pójścia na niekontrolowany żywioł.

Rozdział 4

Zwinne wymagania, czyli czy User Stories są wystarczające?

W różnych miejscach Internetu możesz trafić na informacje o “zwinnych wymaganiach” czy “zwinnej analizie wymagań”.
Co to oznacza?
Jak pracować z wymaganiami w projektach zwinnych?

W poprzednim rozdziale pisałem, że zwinny projekt pozostaje dalej projektem. Dlatego w agile warto stosować metody, które sprawdzają się w klasycznych projektach od dziesięcioleci.
Dotyczy to również pracy z wymaganiami.

Reguły dotyczące modelowania systemów informatycznych, notacje i standardy są takie same dla projektów zwinnych i waterfallowych.
Zmienia się głównie proces.

Jak i dlaczego?
O tym za chwilę...

Im dłużej to odkładasz, tym będzie trudniej

Paul, jako Product Owner, przygotowywał zadania dla programistów w projekcie zwinnym.

Zespół wytwarzający oprogramowanie przyjął, że skoro pracują zwinnie, to wszystkie informacje o zadaniach będą trzymać wyłącznie w Jirze. W trakcie prac pominięto tworzenie jakiejkolwiek innej dokumentacji.

Wymagania we wstępnej formie trafiały do Jiry. Następnie, po analizie i przeglądach z zespołem Paul zamieniał je w precyzyjnie opisane zadania. Był to proces, który występuje dość powszechnie w IT. Dlatego też prace szły sprawnie do przodu.

Do pewnego momentu.

W pewnym momencie Paul potrzebował zmienić funkcjonalność dotyczącą wyliczeń kosztów transportu. Moduł ten powstał ponad rok wcześniej. Wprowadzenie prostej zmiany wymagało teraz **przekopania się przez stertę User Stories, które nie zawsze były aktualne.** Ostatecznie Paul i tak musiał zebrać wszystkie reguły dotyczące obliczeń w jednym miejscu w formie “standardowej” specyfikacji. Trwało to jednak dłużej, niż gdyby robił to na bieżąco.

Historia Paula przypomina mi o tym, że **nie istnieje coś takiego jak "zwinne wymagania"**. Poleganie wyłącznie na User Stories działa tylko w określonych sytuacjach. Sprawdzi się, gdy projekt jest naprawdę prosty, a dostarczenie Działającego Oprogramowania nie wymaga wdrożenia rozbudowanej logiki oraz integracji pomiędzy systemami.

Po przeciwnej stronie są złożone, skomplikowane projekty, których prawidłowe działanie ma krytyczny wpływ na organizację bądź zdrowie lub życie użytkowników. Tego typu systemy wymagają formalnej analizy i zwykle powstają w bardziej tradycyjny sposób. A pośrodku jest cała masa odcieni szarości, czyli systemów posiadających prostsze i trudniejsze moduły.

Warto podchodzić do każdego projektu indywidualnie i zwinnie zbudować taki proces analizy wymagań, który przyniesie najwięcej korzyści przy najmniejszym nakładzie pracy.

Na przykład, w wielu projektach wystarczy, że tylko część wymagań, czyli zakres najbliższego wydania, będzie precyzyjnie opisana i zamodelowana. Pozostałe funkcjonalności mogą być początkowo określone na wysokim poziomie ogólności i zaprezentowane na przykład w formie roadmapy. Nie jest to jednak reguła.

Ważne, żeby proces pracy z wymaganiami oraz zakresem był zrozumiały dla wszystkich. Tak samo ważny jest spójny format specyfikacji i podział odpowiedzialności, czyli kto, kiedy będzie aktualizował takie dokumenty.

Projektując taki proces, uwzględnij między innymi:

- sposób wytwarzania oprogramowania (zwinny lub tradycyjny),
- rodzaj oprogramowania (na zamówienie, czy publicznie dostępne),
- relację zamawiający-wykonawca (czy są z tej samej firmy, czy nie?),
- dostępność interesariuszy w relacji do zrozumienia dziedziny,
- krytyczność systemu,
- wszelkie ograniczenia, jak czas, budżet i inne.

Jeżeli dopiero wchodzisz w temat pracy z wymaganiami w IT, to mam coś, co **pomoże Ci zbudować taki proces i podnieść jakość dostarczanych wymagań**. Nawet jeżeli nie masz doświadczenia technicznego.

Informacje i **darmowy fragment książki są** pod tym linkiem.

Podsumowując, agile czy nie, wymagania muszą być opisane precyzyjnie, bez dziur i sprzeczności. Tutaj nie ma podziału na zwinne i klasyczne. Bez precyzyjnych, testowalnych wymagań, droga do wytworzenia Działającego Oprogramowania stanie się niekończącą serią poprawek i zmian, czasami interpretowanych jako “zwinność”.

Sam proces zbierania, definiowania i aktualizowania wymagań może być jak najbardziej zwinny. Rozwiązania mogą być projektowane przyrostowo w kolejnych iteracjach. Dokumentacja może być nieco “lżejsza”, zwłaszcza na początkowych etapach projektu.

Zwinność to też częste **prototypowanie i testowanie**. “Fail Fast, Learn Faster” to popularne powiedzenie w IT. Skoro nie wiadomo, która z dróg doprowadzi do sukcesu, to czasami warto sprawdzić kilka z nich. Dzięki narzędziom no-codowym zbudowanie prototypu bywa szybsze niż przygotowanie specyfikacji. Prototyp może początkowo stać się właśnie taką specyfikacją.

W pewnym momencie przychodzi jednak czas na projektowanie systemu. Wtedy do akcji wkraczają dobrze ugruntowane narzędzia i standardy. Im bardziej złożony system, tym dokładniej trzeba go zamodelować. Cechy takie jak bezpieczeństwo, niezawodność czy dostępność wymagają solidnej architektury rozwiązania.

Rozdział 5

Jak zarządzać zwinnym projektem bez przesadnych formalności?

Kolejnym popularnym internetowym wątkiem jest pytanie, czy **szacować czasochłonność w Story Point, czy za pomocą godzin** (dni)? Czyli pytania o to, które z podejść jest lepsze.
Jaka jest odpowiedź?

Story Pointy czy w jednostkach czasu?

Wróćmy jeszcze do Kasi.

Kasia w tym projekcie po raz pierwszy zetknęła się z szacowaniem czasochłonności za pomocą tzw. Story Points. Proces ten wyglądał tak, że przed spotkaniem planującym, zespół przeglądał wszystkie User Stories i w trakcie sesji tzw. Planning Pokera przypisywał do każdej User Story wartości ze skali 1, 2, 3, 5 czy 8. Tych wartości nie można było przekładać na jednostki czasu. Jednak w trakcie planowania zespół decydował, że nie weźmie do sprintu więcej niż X, bo tyle wynosi ich średnia z ostatnich 3 miesięcy.

Kasia czuła, że planowanie można nieco usprawnić. Jak bowiem zaplanować pracę na okres urlopowy czy świąteczny, kiedy wydajność spada względem średniej? Zastanawiało ją też, dlaczego nie można wycenić zadań precyzyjniej, czyli np. na 6 czy 7 punktów, skoro zadania były szczegółowo omówione przed sprintem i każdy wiedział, co jest do zrobienia.

Po czasie Kasia znalazła odpowiedzi na te pytania. Okazało się, że proces planowania można było przenieść na jeszcze wyższy poziom!

Kasia zaczęła stosować Story Pointy do szacowania w dłuższej perspektywy i pokazywania biznesowi wpływu różnych decyzji na plan projektu. Planując sprinty, uwzględniała jednak rzeczywistą dostępność poszczególnych osób, obłożenie innymi zadaniami itd.

Czy sytuacja Kasi brzmi może znajomo?

Planowanie projektów zwinnych wydaje się być raczej prostym zadaniem. Wystarczy oszacować złożoność zadań korzystając z relatywnych estymacji (Story Points) i dobrać do sprintu tyle zadań, ile wynosi średnia prędkość.

Takie podejście jednak ogranicza prawdziwy potencjał zwinnego planowania. Jeżeli wejdziesz w ten temat nieco głębiej to szybko okaże się, że tak jak Kasia, będziesz w stanie planować dużo dokładniej, przy niewielkim wysiłku wkładanym w ten proces.

Zarządzanie projektem zwinnym to bowiem znacznie więcej niż dobieranie zadań z długiej listy zwanej backlogiem. Warto, aby projekt realizowany zgodnie z agile był regularnie planowany w dłuższej i krótszej perspektywie, na podstawie zbieranych danych i wskaźników.

Warto też, aby wskaźniki te odnosiły się do założeń projektu, jego celów zdefiniowanych na kolejne przyrosty. Czyli tego, o czym pisałem dużo w Rozdziale 3.

Proces planowania powinien więc dostarczać interesariuszom informacji o spodziewanych terminach osiągnięcia założonych celów, kosztach, zagrożeniach czy możliwych przesunięciach.

Pomocne będą takie wykresy i pomiary jak na przykład:

- burndown chart na poziomie sprintu oraz całej wersji,
- przewidywany termin realizacji na podstawie średnich prędkości,
- czas trwania cyklu (cycle time),
- karty kontrolne do oceny powtarzalności procesu,
- tabela szacowania wpływu (zaawansowane narzędzie, przykład)

To tylko przykłady. W twojej sytuacji bardziej wartościowe mogą być inne metryki. Ja na przykład, w jednym z projektów analizowałem relację estymacji relatywnych (w Story Points) do czasu realizacji zadań, żeby zweryfikować jakość procesu estymowania. Trzeba to było robić z wyczuciem, żeby za chwilę obie metody nie połączyły się w jedną. Ostatecznie dało mi to sporo wartościowych informacji, dzięki czemu poprawiłem proces planowania.

Ważne więc, żeby zbierać i wykorzystywać informacje w jakimś celu. Jak zwykle nie chodzi o to, żeby dokładać sobie pracy, ale żeby ułatwiać sobie i innym życie dzięki posiadaniu wiarygodnych wskaźników.

Żeby przetestować, jak działa u Ciebie proces planowania, możesz wykonać prosty test.
Załóżmy, że w Twoim projekcie jest do wdrożenia zmiana, którą biznes potrzebuje dość pilnie. Jak szybko będziesz w stanie pokazać symulację wpływu tej zmiany na dostarczanie pozostałych funkcjonalności?
Jeżeli taka analiza zajmuje kilka godzin, zamiast kilku kliknięć, to znaczy, że zwinne planowanie można usprawnić.

Rozdział 6

Tajemnica samoorganizacji, czyli esencja zwinności.

Mamy Jirę i daily, czyli jesteśmy zwinni...

Agnieszka obserwowała ciekawą sytuację w projekcie, którym miała się zająć pod nieobecność głównego menedżera projektu.

Teoretycznie proces wyglądał na zwinny. Zespół korzystał z Jiry, miał daily i inne regularne spotkania. Na tych spotkaniach coś jednak nie grało. Lider zespołu odpytywał poszczególne osoby z postępu ich zadań. W zasadzie ciągnął każdego za język, bo **poziom inicjatywy ze strony poszczególnych osób był bardzo niski**. Osoby wywołane do tablicy mówiły o statusie realizacji zadania, typu: “Zrobiłem taska 342, teraz zaczynam 531 i będę go robił. To tyle”.

Agnieszka zastanawiała się, czy tak faktycznie ma wyglądać zwinność. Kiedy chciała porozmawiać o szczegółach współpracy z poszczególnymi osobami, to większość odsyłała ją do lidera zespołu. “On tu rządzi, my tylko się dostosowujemy. Porozmawiaj z nim”.

Czy masz może podobne doświadczenia?

W tej krótkiej historii wybrzmiewa brak większego zaangażowania w wykonywane zadania. Domyślam się, że na spotkaniach online ten zespół nie włączał kamer, a większość osób zajmowała się w trakcie spotkania statusowego innymi rzeczami. Byle tylko zdać raport i wrócić do swoich obowiązków.

Co poszło nie tak?

Przecież zespół pracował zwinnie, więc zaangażowanie i samoorganizacja powinny się pojawić. Na tym polega agile. Ludzie muszą być zaangażowani.

Wbrew pozorom, zarządzanie w projektach zwinnych jest **dużo bardziej wymagające** niż tradycyjne, polegające na autorytarnym wydawaniu poleceń.

Lider zespołu zwinnego powinien równie **zwinnie zmieniać swój sposób pracy**. Na początku pomaga klasyczne podejście, czyli częstsze pokazywanie co i jak jest do zrobienia. Jest to naturalne, bo w początkowych miesiącach pracy nowy zespół nie ma szans się samoorganizować.

Z czasem, dobry zwinny lider, oddaje coraz więcej odpowiedzialności poszczególnym osobom. Tak! Nie tylko deleguje zadania, ale i odpowiedzialność. Słucha i daje regularny feedback. Po jakimś czasie zwinna samoorganizacja staje się możliwa do wprowadzenia w kontrolowany sposób.

Ten proces nie zadzieje się sam z siebie ani przez przypadek. Jeżeli pojawia się prawdziwa samoorganizacja i zaangażowanie to jest to znak, że lider, który już jest mniej zauważalny, wykonał świetną robotę!

Podsumowując, Jira, daily i inne rytuały pomagają tworzyć zgrany, samoorganizujący się zespół. Do pełni sukcesu potrzeba jednak czegoś więcej. Tym czymś jest zbudowanie bezpieczeństwa psychologicznego, wzajemnego szacunku i poczucia wspólnego celu. Do tego potrzebne jest świadome przywództwo, najlepiej zgodne z duchem Servant Leadership (przywództwo wspierające).

Temat zwinnego przywództwa jest niezwykle obszernym zagadnieniem. Poświęciłem mu całą książkę, sporo artykułów, ale przede wszystkim lata praktyki.

Po tych wszystkich latach wiem jedno. Jeżeli potrafisz współpracować z ludźmi, to możesz mieć braki w tematach technicznych, a i tak dasz sobie radę. Odwrotnie, jeżeli jesteś świetnym specjalistą technicznym, ale bez rozwiniętych umiejętności miękkich, to jest bez porównania trudniej.

Rozdział 7

Podsumowanie - najważniejsze zasady

Wszystkie przytoczone historie mają swoje źródło w moich doświadczeniach zawodowych. Zmieniłem tylko nieco imiona i gdzieniegdzie zmodyfikowałem kontekst.
Sens pozostaje niezmieniony.
Podobne historie usłyszysz zresztą w firmach na całym świecie.

Dlatego tak ważne bywa uczenie się z przeszłych doświadczeń sukcesów i porażek. Jak powiedział Eleanor Rossvelt *“Learn from the mistakes of others. You can’t live long enough to make them all yourself.”*

Najważniejsze elementy na **drodze do zwinnego dostarczania Działającego Oprogramowania**, to według mnie:

1. Stosowanie zwinnych technik tylko tam, gdzie faktycznie się sprawdzą, a nie tam, gdzie jest to modne.
2. Określanie celu projektu i innych parametrów analogicznie do tradycyjnych projektów. Projekt zwinny też, jakby nie było, jest projektem.
3. Dostosowanie procesu zbierania wymagań do specyfiki danego projektu. Korzystanie z szerokiego zakresu technik i narzędzi, nie tylko User Stories, jeżeli sytuacja tego wymaga.
4. Planowanie projektu w różnych perspektywach czasowych, korzystając z estymacji relatywnych (np. Story Points) i klasycznych wycen czasowych. Jednocześnie monitorowanie postępu i bieżące reagowanie na wszelkie odchylenia z wykorzystaniem metryk.
5. Świadome zarządzanie w duchu przywództwa wspierającego.

Zdaję sobie sprawę, że na tych kilkudziesięciu stronach nie wyczerpałem całego tematu zwinnego zarządzania projektami.

Pominąłem na przykład temat narzędzi, które współcześnie potrafią zdziałać cuda. Jira jest dalej numerem jeden w zarządzaniu zwinnymi projektami, przynajmniej na polskim rynku. Inne aplikacje jednak ostro depczą jej po piętach, jak choćby niesamowicie elastyczny monday.com (zobacz go w akcji na kilku krótkich nagraniach).

Zachęcam Cię więc do dalszego zgłębiania tematyki zwinności. Wbrew pozorom wiele z podstawowych zasad znajdziesz w czasach oddalonych nawet o 50 lat!

Na przykład, genialny menedżer, Peter Drucker napisał w 1973 roku:

"Throughout management science - in the literature as well as in the work in progress - the emphasis is on

- *techniques rather than principles,*
- *on mechanics rather than decisions,*
- *on tools rather than on results, and, above all,*
- *on efficiency of the part rather than on performance of the whole*"*

Czy to nie brzmi znajomo?

Peter Drucker 50 lat temu podkreślał konieczność stosowania elastycznych reguł, zamiast kurczowego trzymania się tych samych technik. Uważał, że wynik jest dużo ważniejszy niż stosowane narzędzia, czyli rozwiązania.
I co najważniejsze, podkreślał konieczność patrzenia na rozwiązanie jako na całość, a nie zlepek różnych modułów.

*Źródło: Drucker, P. F., Management: Tasks, Responsibilities, Practices, Harper & Roe, New York, 1973.

Peter Drucker nie zajmował się jednak stricte oprogramowaniem.

Zobaczmy więc inny przykład.

Harlan Mills otrzymał w latach 70-tych poważne zadanie.
Projekty realizowane przez IBM dla instytucji rządowych (przetargi publiczne) przynosiły straty ze względu na przekroczenia budżetu i czasu trwania. Harlan Mills miał to zmienić.

Dziesięć lat później tak opisywał rezultaty, które osiągnął:

"Ten years ago general management expected the worst from software projects – cost overruns, late deliveries, unreliable and incomplete software

Today management has learned to expect on-time, within budget, deliveries of high-quality software.

A Navy helicopter ship system, called LAMPS, provides a recent example. LAMPS software was a four-year project of over 200 person-years of effort, developing over three million, and integrating over seven million words of program and data for eight different processors distributed between a helicopter and a ship in ***45 incremental deliveries.****"

Czy to czasem nie jest klasyczny agile?
Czteroletni projekt o budżecie 200 osobo-lat zrealizowany dzięki systematycznemu dostarczaniu kolejnych kawałków oprogramowania.
Krok po kroku.
Dając wartość z każdym przyrostem.

*Źródło: IBM Systems Journal 4/1980, "The management of software engineering Part 1: Principles of software engineering", H.D.Mills

Jeżeli miałbym na koniec przekazać Ci jedną tylko myśl, to chciałbym, żebyś w poszukiwaniach drogi do optymalnej zwinności **korzystał z prostych, sprawdzonych przez dziesięciolecia zasad i wdrażał je wspierając się wszelkimi nowinkami technologicznymi.**

Wiele z współczesnych “nowatorskich” metod to tak naprawdę stare, dobre techniki ubrane w modne i chwytliwe nazwy. Nic więcej.

A jeżeli treść tego **ebooka rezonuje z Twoim wyobrażeniem zwinności** i przynajmniej kilka razy kiwałeś głową czytając te treści, to mam w ramach Project Makers **coś jeszcze**, co może Cię zainteresować.

Jeżeli jesteś na **wczesnym etapie wdrażania podejścia zwinnego** i szukasz sposobów na usystematyzowanie wiedzy, to szkolenia z tego zakresu będą odpowiednim miejscem, żeby zacząć. Zobacz aktualną ofertę 👩‍🏫 szkoleń otwartych, zamkniętych i kursów online.

Jeżeli **wchodzisz w rolę lidera** to książka “Fascynujący Świat Przywództwa” będzie strzałem w dziesiątkę ➡ Zobacz szczegóły ⬅

Jeżeli już **działasz zwinnie** i zaczynasz dostrzegać wyzwania, na które nie ma odpowiedzi w książkach czy na szkoleniach, to jest duża szansa, że znajdziemy rozwiązania w trakcie indywidualnych sesji. Umów się na bezpłatne spotkanie organizacyjne. Im szybciej tym lepiej, bo miejsca rozchodzą się dość szybko: wybierz termin 🗓️

Nazywam się **Artur Guła.**

Przez ponad dekadę pomagam **dostarczać wysoką jakość w projektach IT**.
Pracuję głównie z małymi zespołami, które dostarczają niebanalne rozwiązania software'owe.

Byłem ekspertem Polskiej Agencji Rozwoju Przedsiębiorczości z obszaru zarządzania projektami.

Jestem autorem książek o zarządzaniu i wielu artykułów na ten temat.

artur@projectmakers.pl

https://linkedin.com/in/arturgula/

Dzielę się doświadczeniami na konferencjach branżowych, podcastach i spotkaniach online.

Od 2019 roku prowadzę szkolenia i warsztaty, które pomagają uczestnikom jeszcze lepiej zrozumieć zwinne zarzadzanie projektami.

Preferuję podejście skupione na osiąganiu jasno określonych celów. Sprint po sprincie.
Ze wsparciem odpowiednich narzędzi.
W duchu zwinnego przywództwa Servant Leadership.