

4.1.1.

Historia powstania

Wskutek wydarzeń historycznych nastąpił pewien zbieg okoliczności, który, jak to zwykle bywa, dla jednych osób stanowił źródło frustracji, a dla innych okazję do wykorzystania. W trakcie drugiej wojny światowej Japonia walczyła ze Stanami Zjednoczonymi Ameryki i w końcu tę wojnę przegrała. Warunki niedoboru surowców spowodowane zniszczeniami wojennymi stworzyły środowisko, w którym stosowanie popularnej w tym czasie produkcji masowej było niemożliwe. Jak zatem zbudować niewielką liczbę samochodów, zachowując równocześnie ich niską cenę? Przed takim dylematem stanęła w 1940 roku mała firma Toyota. Odpowiedź znalazł Taiichi Ohno, uznawany dzisiaj za ojca Toyota Production System. Jeżeli firmy nie stać na produkowanie samochodów na zapas i przechowywanie części w magazynie, należy zbudować system, który zakłada, że produkt zostanie dostarczony do klienta najszybciej jak to jest możliwe od momentu złożenia zamówienia. Głównym napędem tego podejścia jest eliminacja straty, czyli wszystkich elementów procesu, które nie przyczyniają się do wytworzenia produktu, a tym samym nie dostarczają wartości.

W latach dziewięćdziesiątych ubiegłego wieku praktyki Toyoty zostały uznane za standard wiodący i wdrożone w wielu fabrykach. TPS wpłynął także na powstanie takich dziedzin, jak Lean Software Development i Lean Management.

4.1.2.

Zasady Lean

Podstawą Lean są zasady (ang. *Lean Principles*), które Mary Poppendieck odniosła do środowiska wytwarzania oprogramowania. Przyjrzyjmy się im po kolei.

Eliminuj stratę (ang. *Eliminate Waste*)

Proces wytwarzania produktu powinien jak najszybciej dostarczyć klientowi wartość. Wszystko, co z punktu widzenia klienta przyczynia się do dostarczenia wartości, jest również wartościowe z punktu widzenia Lean. I odwrotnie – wszystko, co nie przynosi wartości, jest stratą (ang. *waste*), czy – jak to trafniej można określić w języku polskim – marnotrawstwem. Eliminując elementy przynoszące straty, podnosimy efektywność naszych praktyk.

W nomenklaturze Toyota Production System strata jest określona japońskim słowem ***muda*** i stanowi jeden z trzech typów zbędnej wariacji w systemie. Dwa pozostałe to ***mura***, czyli nieregularność, i ***muri***, czyli nadmierne obciążenie pracą. Szukanie i eliminowanie straty jest zadaniem ciągłym. Podstawowym narzędziem służącym do określania efektywności procesu i szukania w nim strat jest **Mapowanie Strumienia Wartości** (ang. **Value Stream Mapping**).

Tabela 1. Straty w produkcji i odpowiedniki tych strat w wytwarzaniu oprogramowania

Strata w produkcji	Strata w wytwarzaniu oprogramowania
Zapasy (ang. <i>Inventory</i>)	Częściowo wykonana praca (ang. <i>Partially Done Work</i>)
Zbędne przetwarzanie (ang. <i>Extra Processing</i>)	Dodatkowe procesy (ang. <i>Extra Processes</i>)
Nadprodukcja (ang. <i>Overproduction</i>)	Dodatkowe funkcjonalności (ang. <i>Extra Features</i>)
Transport (ang. <i>Transportation</i>)	Przełączanie zadań (ang. <i>Task Switching</i>)
Czekanie (ang. <i>Waiting</i>)	Czekanie (ang. <i>Waiting</i>)
Ruch (ang. <i>Motion</i>)	Ruch (ang. <i>Motion</i>)
Błędy (ang. <i>Defects</i>)	Błędy (ang. <i>Defects</i>)

Suchanie strat należy zacząć od sprawdzenia listy siedmiu tradycyjnie wymienianych strat procesu produkcji oraz straty dodanej po wprowadzeniu Lean Management do sektora usług.

1. Częściowo Wykonana Praca

Częściowo wykonana praca deweloperska zwykle jest zbędna, a ponadto stwarza niepożądane ryzyko. Jeżeli kod nie został zintegrowany i przetestowany pod kątem zgodności ze standardami, to może być on źródłem błędów i komplikować dalszy rozwój produktu. Pamiętajmy, że systemy informatyczne to ciągle rozwijane, skomplikowane środowiska pełne zależności, dlatego sytuacje, w których złe praktyki określonego modułu spowodują problemy tylko w nim, w praktyce nie występują.

Jeżeli nie zdążyliśmy wykonać zaplanowanych zadań, a praca nie zostanie dokończona w kolejnej iteracji, co zresztą nadal jest złą praktyką, to ktoś wyrzucił pieniądze w błoto.

Żeby uniknąć takich sytuacji, należy zwrócić uwagę na praktyki planowania i ustalania priorytetów. Należy również pilnować dyscypliny pracy w iteracjach, ponieważ lepiej nie zacząć pracy wcale, niż zostać z częściowo wykonaną pracą na koniec iteracji.

2. Dodatkowe procesy

Klasycznym przykładem w tej kategorii jest tworzenie niezliczonych prezentacji i arkuszy kalkulacyjnych z zestawieniami i raportami, których nikt nie czyta. Przygotowanie dokumentacji tylko po to, aby klient zaakceptował system, czy tworzenie i przypisywanie zadań w komputerowym systemie zarządzania projektem, żeby oficjalnie zatwierdzić kolejny etap, to strata.

Jeśli to tylko możliwe, należy redukować dokumentację do rozmiarów strony A4 i postaci tabeli lub mapy myśli. Zamiast dokumentacji wymagań i przypadków użycia można stworzyć automatyczne testy akceptacyjne, które staną się zżyjącą dokumentacją