KanBan i jego podstawy (JIT&Lean)

KanBan po raz pierwszy pojawił się w Japonii po II wojnie światowej jako bardzo ważny element “Systemu Produkcyjnego Toyoty”, przełomowej zmiany w podejściu do **zarządzania procesem produkcyjnym** w **rozwoju samochodów.** System ten, znany również jako JIT (just in time), został wynaleziony i wdrożony w Toyocie przez grupę inżynierów pod kierownictwem Taiichi Ohno.

>

W roku 90’ zarządzanie procesem produkcyjnym JIT zostało skonfrontowane z innymi systemami produkcyjnymi w książce “The Machine That Changed the World” (autorstwa Jamesa P. Wormacka, Daniela T.Jonesa & Daniela Roosa). Konfrontacja ta doprowadziła do zdefiniowania tego, co dziś rozumiemy jako **LEAN** zdefiniowane w jego zasadach:

1. *Określenie wartości z punktu widzenia klienta końcowego według rodziny produktów.*
2. *Zidentyfikuj wszystkie kroki w strumieniu wartości dla każdej rodziny produktów, eliminując w miarę możliwości te kroki, które nie tworzą wartości.*
3. *Spraw, aby kroki tworzące wartość występowały w ścisłej kolejności, tak aby produkt płynnie płynął w kierunku klienta.*
4. *W miarę jak przepływ jest wprowadzany, pozwól klientom czerpać wartość z następnego działania wyższego szczebla.*
5. *Po określeniu wartości, zidentyfikowaniu strumieni wartości, usunięciu zmarnowanych kroków oraz wprowadzeniu przepływu i przyciągania, rozpocznij proces ponownie i kontynuuj go, aż do osiągnięcia stanu doskonałości, w którym tworzona jest idealna wartość bez marnotrawstwa.*



Na pierwszy rzut oka trudno się połapać, jak dokładnie usprawnienia w procesie produkcji samochodów można zaadoptować w inżynierii oprogramowania, ale jak już pewnie wiesz, zasady LEAN były jedną z inspiracji dla obu **wynalazców SCRUM & sygnatariuszy Agile Manifesto.**

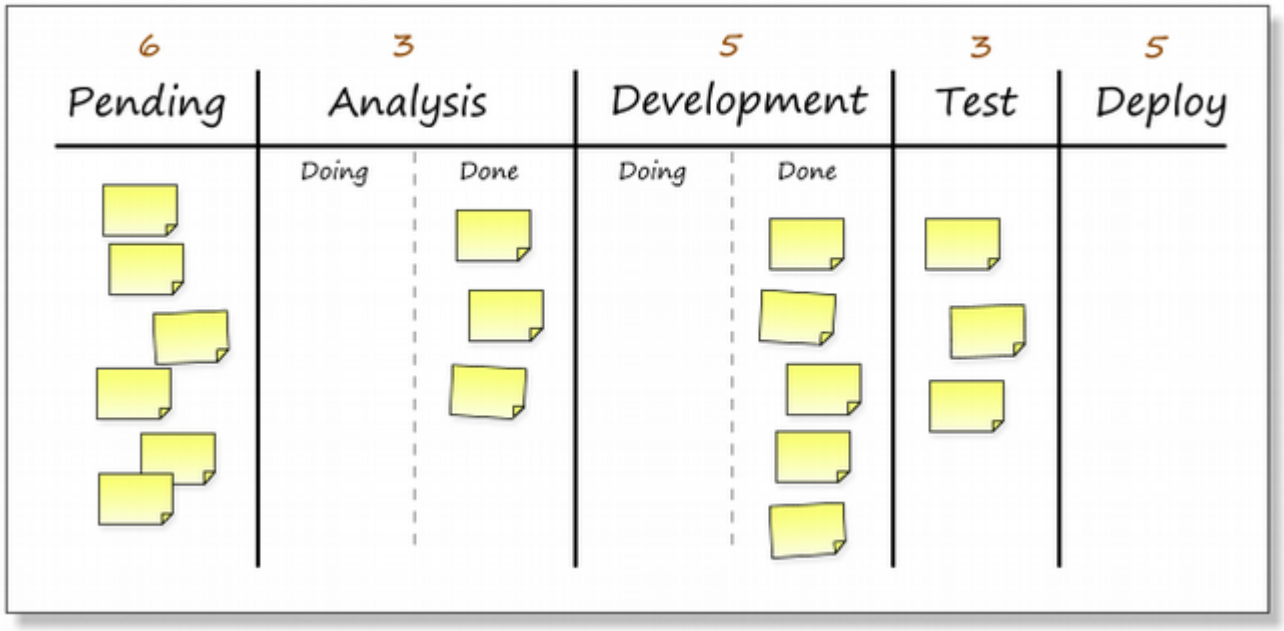
>

Na pierwszy rzut oka trudno się połapać, jak dokładnie usprawnienia w procesie produkcji samochodów można zaadaptować w inżynierii oprogramowania.

Tablica KanBan, limit WiP,

system pull

Najprostszym sposobem wyrażenia KanBan jest wyraźne pokazanie pełnej mapy strumienia wartości na Tablica KanBan (później zdefiniowana jako “tablica”). Tablica powinna być widoczna i łatwo dostępna dla wszystkich programistów, którzy są odpowiedzialni za wszystkie kroki występujące w strumieniu wartości.



Kierunek strumienia wartości – tylko od lewej do prawej!

>—>—->—>—->—>—>->—>—>—>—>—>—>—>—>—>—

>

Teraz mamy mapę rozwoju naszego produktu. W naszym przypadku wartością produktu jest działające oprogramowanie, więc możemy sobie wyobrazić, że każdy sticky na pokładzie jest małą “partią” potencjalnie działającego oprogramowania (później zdefiniowanego jako “feature”). Każda funkcja, która jest planowana do rozwoju, musi “podróżować” przez tablicę od lewej do prawej przez wszystkie etapy (oczekiwanie, analizy, rozwój, testowanie, wdrażanie), zanim będzie mogła zostać dostarczona jako wartość dla klienta końcowego. W tym momencie możesz zapytać, gdzie jest haczyk? Wdrożenie tego rodzaju przepływu pracy nie wydaje się trudne. Istnieją (przynajmniej :D) dwie pamięci podręczne. Jednym z nich jest zaimplementowanie limitu pracy w toku (WiP Limit), drugim jest zaimplementowanie systemu Pull.

Work in Progres Limit (WiP limit):
Nad każdą nazwą kroku na tablicy znajduje się liczba, któraokreśla maksymalną ilość “stickies” (cech), które mogą zostać opracowane na danym

etapie strumienia wartości. Może to być sprzeczne z intuicją, aby uznać, że ograniczenie pracy w toku może faktycznie pomóc w szybszym dostarczaniu wartości klientom końcowym. Prawda jest taka, że to naprawdę pomaga, a dzieje się tak dzięki zrozumieniu prawa Little'a, które mówi:

> **Prawo Little'a ujawnia, że ogólnie rzecz biorąc, dla danego procesu o danej przepustowości, im więcej rzeczy, nad którymi pracujesz w danym czasie (średnio), tym dłużej zajmie ukończenie tych rzeczy (średnio).**

.

Pull System:

W "normalnym" systemie push osoba odpowiedzialna za etap analizy po prostu PUSH funkcję, która właśnie została przez nią przeanalizowana do rozwoju. W tej sytuacji deweloper nie ma kontroli nad limitem własnej

pracy w toku. Albo czeka, aż funkcje “wskoczą” do rozwoju (co jest stratą czasu), albo jest pokryty ogromną ilością funkcji do opracowania na raz, przez co etap rozwoju staje się wąskim gardłem (rozwiązywanie wąskich gardeł również powoduje stratę czasu).

>

Jak widać na planszy w dwóch istniejących krokach (analiza & rozwój) istnieją dodatkowe podkroki – robienie & zrobione. W systemie pull developer WYCIĄGA funkcję do rozwinięcia z wykonanego podetapu kroku analizy. Daje mu to (deweloperowi) kontrolę nad własnym limitem pracy w toku. Może on rozpocząć pracę nad nową funkcją tylko wtedy, gdy posiada pojemność, którą tworzy kończąc pracę nad poprzednią funkcją.

Tworzenie systemu pull i ciągłe sprawdzanie & dostosowywanie go *teoretycznie*:

- **wyeliminowanie możliwości straty czasu z powodu oczekiwania na pracę oraz rozwiązywania wąskich gardeł;**

- **daje możliwość opracowania i dostarczenia *bezbłędnej wartości dla klienta końcowego.***

Oczywiście w rzeczywistości nie jesteśmy w stanie całkowicie wyeliminować marnotrawstwa, ale opanowanie przepływu pracy za pomocą tablicy KanBan jest doskonałym narzędziem do utrzymania marnotrawstwa na minimalnym możliwym poziomie.

Możesz ponownie pomyśleć, że wdrożenie systemu WiP & Pull to nic wielkiego, ale skonfrontuj swoje oczekiwania z informacją, że zajęło to 10 LAT (50’-60’) dla Toyoty, aby w pełni wdrożyć metodę JiT do produkcji. Zmagali się z niezliczonymi większymi problemami (4-miesięczny strajk WSZYSTKICH pracowników produkcyjnych z powodu NIEAKCEPTOWANIA że muszą teraz nauczyć się obsługi wszystkich maszyn na danej linii produkcyjnej) lub mniejszych (jak wymieniać informacje między stacjami roboczymi, które są głośniejsze niż silniki samolotowe) problemów podczas przejścia ze “standardowego” sposobu na ten nowy sposób produkcji JiT.

Podsumowanie: Plusy, minusy Kan Ban

Podsumowanie: Plusy, minusy Kan Ban
Nawet jeśli coś jest bardzo trudne do osiągnięcia, nie oznacza to, że nie należy do tego dążyć. Wdrażanie KanBan w organizacjach/zespołach jest teraz o wiele łatwiejsze ze względu na długą historię nakazów i zakazów opisanych w wielu książkach o LEAN w praktyce.

KanBan jest bardzo elastyczny ale trudnym do opanowania narzędziem do:

- **rozwój i utrzymanie zarówno złożonych produktów (na przykład rozwój oprogramowania) & “nie tak złożonych” produktów jak samochody itp;**
- **szybkie i & precyzyjne rozpoznawanie gdzie & kiedy występuje wąskie gardło (dając tym samym możliwość skupienia się TYLKO na tych problemach, które przeszkadzają w dostarczaniu wartości, skupianie się na innych problemach jest marnotrawstwem);**
- **minimalizacja ilości marnotrawstwa, które**

występuje w określonym przepływie pracy/produkcji;

- **płynne dostarczanie wartości do klienta końcowego;**

Zalety Kanban:

- **KanBan jest radykalnie nienakazowy i łatwy do zastosowania do każdego danego Możesz poświęcić 10 minut na narysowanie pierwszego szkicu tablicy KanBan dla swojego przepływu pracy i już możesz zacząć go używać. Nie ma ról, artefaktów, zestawu obowiązków, tajnych spotkań itp. KanBan dostosowuje się do Ciebie. Nie Ty do KanBan.**
- **Po ustanowieniu przepływu pracy Kanban pewien przepływ zaczyna dyktować tempo, w jakim elementy pracy przemieszczają się przez strumień wartości. Pozwala to przewidzieć, jak szybko określona ilość wartości może zostać dostarczona do klienta końcowego.**

Wady KanBan:

- **KanBan jest radykalnie nienakazowy i łatwy do zastosowania w dowolnym przepływie**

pracy ale nie jest w żadnym wypadku łatwym procesem do utrzymania! Z definicji KanBan to’ciągłe poszukiwanie ulepszeń i cykl redukcji marnotrawstwa. Nie ma “końca” procesu adaptacji KanBan i jeśli organizacja nie chce kontynuować tego procesu, to nie znajdzie prawdziwych przyczyn występujących marnotrawstw & wąskich gardeł. Brak ściśle określonej struktury spotkań, ról i odpowiedzialności może być również frustrujący, ponieważ KanBan wymaga absolutnej uwagi szczegółów i wzajemnego zrozumienia zasad lean od każdego etapu strumienia wartości członka zespołu. Wymuszanie KanBan “w niewłaściwy sposób” (bez absolutnego zrozumienia jego celu przez wszystkich uczestników) może stać się przypadkiem, w którym wzajemna odpowiedzialność za dążenie do doskonałości jest źle rozumiana przez indywidualną odpowiedzialność za wskazywanie palcem każdego, kto popełnia błędy.

.

- **Po ustanowieniu przepływu pracy KanBan**

> **określony przepływ zaczyna dyktować tempo przemieszczania się elementów pracy przez strumień wartości. Pozwala to przewidzieć, jak szybko określona ilość wartości może zostać dostarczona do klienta końcowego.**

To jest dokładnie to, czym jest – prognozą. Wyobraźmy sobie teraz, że firma X zostaje poproszona przez klienta o dostarczenie złożonego, w pełni rozwiniętego produktu w ściśle określonym czasie. Jeśli X zaakceptuje takie żądanie podczas adaptacji KanBan, naraża się na ryzyko nieznajomości rzeczywistej przepustowości swojego zespołu. Będą pod ogromną presją dostarczania na czas, co z pewnością dramatycznie zwiększy koszty ORAZ obniży jakość produktu końcowego. (jako że 2-krotny wzrost liczby deweloperów nie sprawi , że w magiczny sposób dostarczą produkt 2x szybciej, jakość produktu końcowego musi ucierpieć)