



Manifest zwinności : każdy i zespół planują razem

Sprawne (zwinne) metody przysły z początkiem lat 90. do planowania oprogramowania uzyskując około roku 1999 po raz pierwszy popularność przez programowanie ekstremalne (Kent Beck). Znaczenie zwinnych metod niezwykle wzrosło ostatnimi laty. Dotyczy to zwłaszcza - ale nie tylko – sektora IT: 62% projektów rozwoju oprogramowania pracuje w Niemczech metodą zwinności, 16 % zgoła wyłącznie, podczas gdy zaledwie 22% ogółu projektów pracuje konwencjonalnymi metodami.

Filozofia „zwinnej pracy“ podkreśla zdolność pojedynczych osób i samoorganizacji zespołu rozwojowego. Za pomocą „sprawnych metod“ zabiega się o usunięcie słabych stron modeli rozwoju programów, uchodzących za zbyt sztywne. Ponieważ sztywne i przenikające planowanie nie jest praktyczne, to od samego początku nie mamy ustalonych wymagań wobec oprogramowania. Rozwija się ono mianowicie wraz z rosnącym zrozumieniem systemu oprogramowania, co ma zostać osiągnięte przede wszystkim na drodze intensywnej komunikacji z klientem. Dla ułatwienia komunikacji ekstremalne programowanie wykorzystuje dla zrozumienia tak zwane „User Stories“, zatem opowiadania użytkowników w celu opisania właściwości systemowych względnie istotnych funkcji. Planowanie należy przeprowadzić we współpracy i nie jest to wyłączna decyzja managera projektu. Współpracę grupy na rzecz planowania zakresu kolejnego programu czy jego udostępnienia określa się jako „Planning Game“. Gry planowe i szacunek nakładów na projekt powtarza się cyklicznie (proces iteracyjny).

Podczas spotkania w Utah w 2001 roku wybrano na określenie takiej metody rozwoju oprogramowania pojęcie „zwinność“ i opracowano „Manifest zwinności“ (sprawności) (www.agilemanifesto.org/iso/de/manifesto.html): **„wykorzystujemy nowe drogi opracowywania software, zajmując się tym samemu i pomagając w tym innym. Przez tę pracę nauczyliśmy się doceniać te wartości:**

- **Osoby i interakcje** stoją ponad procesami i narzędziami
- **Funkcjonalny software** stoi powyżej najszerzej dokumentacji
- **Współpraca z klientem** stoi powyżej negocjacji nt. umowy
- **Reagowanie na zmiany** stoi powyżej realizacji planu.”

Sprawne (zwinne) zasady to dyrektywy zwinnej pracy. Czasem zwinne zasady określa się także mianem metody. **Manifest zwinności obejmuje 12 zasad.**¹

- Zadowolenie klienta poprzez wczesną i permanentną dostawę cennego software
- Zwinne procesy wykorzystują zmiany (nawet późno rozwojowe) jako przewagę konkurencyjną dla klienta.
- Dostawy działającego software iw regularnych, zwłaszcza krótkich, odstępach czasu (kilka tygodni lub miesięcy)
- Niemalże codzienna współpraca ekspertów i programistów w ramach projektu (np.: Collective Code Ownership)
- Udostępnienie środowiska i udzielenie wsparcia niezbędnych dla wypełnienia zadań przez użytkownika
- Przekaz informacji o ile to możliwe na wysokości oczu w indywidualnej rozmowie
- Za wyżną miarę postępu przyjmuje się funkcjonalność software

- Utrzymanie równego tempa pracy przez zleceniodawców, programistów i użytkowników w celu zrównoważonego rozwoju
- Stały akcent na techniczną perfekcję i dobry design
- Istotą jest prostota
- Samoorganizacja zespołów w fazie planowania i realizacji
- Autorefleksja zespołów na temat własnej postawy w zakresie dostosowania do wzrostu wydajności

Zasada zrównoważonego tempa: bez godzin nadliczbowych

) MAINTAIN A SUSTAINABLE PACE



Jedną z dwunastu zasad zwinności jest zrównoważone tempo (sustainable pace). W Beck istnieje „40-hour week” (40-godzinny tydzień pracy) jako jedna z 12 praktyk ekstremalnego planowania. Godziny nadliczbowe to wskazówka na poważne problemy w

projekcie, jakie należy zlokalizować i rozwiązać. Na skutek permanentnych godzin nadliczbowych programiści nie są raczej w stanie kodować jasno i wyraźnie. Stają się oni nieproduktywni i programują więcej błędów. Roman Pichler, jako wybitny przedstawiciel, pisze: w Scrum – najbardziej rozpowszechnionej zwinnej metodzie – nie wolno akceptować pracy w godzinach nadliczbowych.

»zwinny rozwój software jest zbiorczym pojęciem wielu metod i praktyk, bazujących na wartościach i zasadach Manifestu zwinnej produkcji software.« (Agile Alliance, 2018)

W **Scrum**, najbardziej rozpowszechnionej zwinnej metodzie postępowania, istnieją role Product Owner, Team i Scrum Master. Product Owner przejmuje punkt widzenia klienta i opisuje w tak zwanym Product Backlog wymagania niezbędne dla osiągnięcia celu projektu. Dokument uzupełnia on systematycznie ustalając priorytety wymagań w zależności od ogólnego postępu.

Wymagania realizuje się w tak zwanych sprintach o zawsze identycznej długości (2-4 tygodni).. W Sprint Planning zespół programistów uzgadnia wymagania z Product Owner. Zespół wybiera zadania do opracowania w kolejnym sprincie. Szacuje on powiązane z tym nakłady by zapewnić wykonalność w wybranym okresie sprintu.

Wyniki sprintu prezentowane są w Sprint Review Product Owner (odpowiednio klientom/użytkownikom). Za załatwione uważa się funkcjonalności wyłącznie sprawdzone i nadające się do użytku

Team organizuje się w danym procesie całkowicie samodzielnie i musi dysponować należytyymi zasobami i zdolnościami. W retrospektywie po sprincie zespół ocenia proces pracy i kooperacji celem wypracowania środków na rzecz poprawy. Zespół wspiera tu w charakterze eksperta ds. metody Scrum Master w celu dotrzymania ustalonych zasad i usuwania przeszkód na drodze do realizacji.