| **Różnice pomiędzy sekwencyjnymi a zwinnymi metodykami rozwoju** | |
| --- | --- |
| **ZWINNE**  **(Scrum, Kanban)** | **SEKWENCYJNE**  **(Waterfall, Model V)** |
| **RODZAJE PROJEKTÓW** | |
| Długoterminowe i bezterminowe. | Krótkoterminowe. |
| **FAZY ROZWOJU PROJEKTU** | |
| Tworzenie oprogramowania podzielone na mniejsze projekty. Poszczególne fazy tworzenia oprogramowania mogą wystąpić więcej niż raz podczas tworzenia projektu. | Tworzenie oprogramowania kończymy pełnym projektem. Wszystkie fazy rozwoju projektu, takie jak projektowanie, rozwój, testowanie itp., są zakończone jednorazowo. |
| **PROCES TWORZENIA PRODUKTU** | |
| Budujemy przyrostowo dostarczając wartość w kolejnych iteracjach (Scrum-sprintach). | Proces tworzenia oprogramowania jest podzielony na poszczególne fazy, gdzie kolejna rozpoczyna się po zakończeniu poprzedniej. |
| **PODEJŚCIE** | |
| Produktowe - zapewnienie, że opracowany produkt spełnia wymagania użytkowników i jest dostosowywany, jeśli zmieniają się potrzeby użytkowników. | Projektowe - koncentruje się na zakończeniu projektu  i końcowej jakości. |
| **ZMIANA** | |
| Podejście elastyczne które umożliwia wprowadzanie zmian w wymaganiach nawet po zakończeniu wstępnego planowania. | Z góry określone wymagania. Nie ma miejsca na zmianę wymagań po rozpoczęciu opracowywania projektu. |
| **ODPOWIEDZIALNOŚĆ** | |
| Cały zespół odpowiada za finalny rezultat. | Pracę rozdziela się pomiędzy różne osoby i to im przydziela się odpowiedzialność za dany wycinek pracy. |
| **KOSZTY NAPRAWY BŁĘDÓW** | |
| Testowanie na każdym etapie projektu. Potencjalne bugi wykrywane wcześniej co obniża koszt ich naprawy. | Testowanie rozpoczyna się dopiero po kodowaniu co podwyższa koszt naprawy wykrytych bagów. |

| **MODEL WODOSPADOWY** | |
| --- | --- |
| **ZALETY** | **WADY** |
| Łatwy do zarządzania, zrozumienia i funkcjonalny. Każda faza ma określone rezultaty i proces oceny. | Spełnia tylko ściśle określone wymagania. |
| Dość prosty dzięki stałej strukturze. | Nie może być stosowany do projektów w fazie monitorowania. |
| Oszczędza dużo czasu. Szybka dostawa projektu. | Nie ma możliwości poznania możliwego wyniku projektu. |
| Umożliwia łatwe testowanie i analizę. Łatwo adaptowalna metoda dla różnych zespołów na daną fazę. | Nie nadaje się do projektów długoterminowych  i bezterminowych. |
| Działa dobrze w przypadku mniejszych projektów, w których wymagania są łatwe do zrozumienia. | Możliwe wysokie koszty naprawy błędów w związku z późnym rozpoczęciem testowania. |
| **MODEL ZWINNY** | |
| **ZALETY** | **WADY** |
| Proces skoncentrowany na klienta. Ciągłe zaangażowanie użytkownika na każdym etapie. | Nie jest to przydatna metoda dla małych projektów. |
| Włączenie testowania jako części i punktu wyjścia tworzenia oprogramowania (test-driven development). | Projekt może łatwo wykoleić się, jeśli właściciel produktu nie jest pewien, jakiego rezultatu oczekuje.. |
| Dostępność wsparcia interesariuszy biznesowych dla testerów w rozwiązywaniu pytań dotyczących oczekiwanego zachowania systemu. | Koszty wytworzenia oprogramowania mogą być wysokie. |
| Samoorganizujące się zespoły, w których cała grupa jest odpowiedzialna za jakość i daje testerom większą autonomię w ich pracy. | Uboga dokumentacja, która nie powinna przekraczać kilku stron a4. |
| Skupienie się na tworzeniu działającego oprogramowania o wysokiej jakości. | Istnieje ryzyko, że samozarządzające się zespoły nie będą odpowiednio skoncentrowane na pracy. |