| Modele sekwencyjne | Modele zwinne - Agile (Scrum, Kanban) |
| --- | --- |
| **Model Waterfall “Krok po kroku”:**  Składa się z liniowo uporządkowanych, następujących po sobie faz: planowanie —> analiza wymagań —> projektowanie —> implementacja —> testowanie —> pielęgnacja.  Model nie uwzględnia bezpośrednio możliwości wykorzystania faz wcześniejszych do planowania i projektowania testów. Błędy będą znajdowane dopiero po fazie implementacji, co zwiększa koszt ich usunięcia.  **Model V “Tak szybko, jak to możliwe”:**  Wytwarzanie i testowanie są równie ważnymi czynnościami - V.  Jego lewe ramię pokazuje następstwo faz wytwórczych, a prawe – następstwo faz testowych.  Podczas zbierania wymagań mogą powstawać plany testów akceptacyjnych, w fazie specyfikacji powstają plany testów systemowych, w fazie projektu technicznego powstają plany testów integracyjnych, a w fazie kodowania - plany testów modułowych.  Mają ściśle określony work flow. | **Model przyrostowy (inkrementacyjny):**  Do istniejącej części produktu dodawana jest jakaś mała część (inkrement). Inkrementem może być zarówno małe ulepszenie istniejącej funkcjonalności, jak i dodanie nowej.  Zdefiniowane są ścisłe wymagania, które dzielone są na zbiory, do których dodaje się funkcjonalności, aż całe oprogramowanie będzie gotowe.  **Model iteracyjny:**  W iteracyjnym modelu wytwarzania oprogramowania grupy funkcji są definiowane, projektowane, tworzone i testowane razem w serii cykli, w określonym okresie czasu. Każda iteracja dostarcza działające oprogramowanie.  Nie ma pełnych wymagań. Dodaje się funkcjonalność krok po kroku tak, aby oprogramowanie w każdej iteracji działało w całości.  Nie mają ściśle określonego workflow. |
| Zalety:  Świetnie sprawdza się w długoterminowych projektach trwających rok lub dłużej, najczęściej związanych z powstaniem namacalnego efektu, z obszerną specyfikacją wymagań  Wady:   * długi czas wytwarzania produktu * wysoki koszt produkcji * nieelastyczny podział na kolejne fazy * konieczny nadzór kierownictwa * zakres, który w podejściu tradycyjnym musi być bardzo dobrze znany już na samym początku projektu. * moment weryfikacji wyników prac przez klienta (zewnętrznego lub biznes) dopiero po zakończeniu etapu lub całego projektu; wprowadzanie zmian na takim etapie jest trudne, ze względu na pierwotnie ustalony plan, a w skrajnych przypadkach może prowadzić do konieczności anulowania i ponownego przygotowania projektu; * wysoki stopień sformalizowania; obszerna dokumentacja projektowa; im obszerniejsza dokumentacja w planowaniu z wyprzedzeniem tym trudniej reagować na późniejsze zmiany wymagań; * nie znajduje zastosowania w projektach obarczonych ryzykiem i niepewnością, w których na etapie planowania projektu nie zna się pełnych wymagań i oczekiwanych wyników. | Zalety:   * Praca podzielona jest na krótkie cykle (sprinty), w trakcie których zespół skupia się na dostarczeniu konkretnej wymaganej przez klienta wartości, która stanowi część finalnego rozwiązania; * samoorganizujące się zespoły i stąd większa autonomia testerów odpowiedzialność całej grupy * lepsze zarządzanie, kontrola projektu oraz monitorowanie i adresowanie ryzyka niż w modelu sekwencyjnym; * wysoka produktywność wytwarzania * szybka reakcja na zmiany * dostępność wsparcia interesariuszy biznesowych dla testerów w rozwiązywaniu pytań dotyczących oczekiwanego zachowania systemu; * mniej formalizmów niż w modelu sekwencyjnym * ciągłe testowanie funkcjonalne * szeroka perspektywa testowania systemowego * duża interakcja osobista * testerzy mogą działać przeważnie jako trenerzy w zakresie testowania zarówno dla interesariuszy, jak i programistów * częste regresje * automatyzacja bardziej opłacalna * w każdym sprincie dostarcza się funkcjonalne przyrosty (np. pojedyncza działająca funkcja aplikacji), które mogą być oceniane i weryfikowane przez klienta; * łatwiejsza identyfikacja błędów w oprogramowaniu * prosta konstrukcja projektu, która rozwija się wraz z jego postępem   Wady:   * jest łatwe w teorii i w wymogach formalnych, ale bardzo trudne do wdrażania w praktyce; * wymaga wysokich kompetencji i zaangażowania zespołu * konieczne wsparcie i zaufanie menedżerów wyższego szczebla aby możliwe było wprowadzenie niektórych zasad i funkcjonowanie w sposób zwinny * konieczność kompleksowej transformacji całej firmy w kierunku Agile * konieczność ciągłego doskonalenia się zespołów * wymaga zbierania wymagań i informacji zwrotnej w każdej iteracji * podejście Agile nie dotyka takich aspektów jak sprzedaż projektu, analiza wymagań biznesowych, rekrutacja i szkolenia zespołu, dokumentowanie pracy, zarządzanie kosztami i konfiguracją oraz relacja z udziałowcami i stosunki prawno-formalne. |