## Testy, które mają sprawdzić czy system przy danym obciążeniu jest w stanie wykonywać usługi z założoną wydajnością, to:

testy wydajnościowe

## W modelu interakcji zdalnej bazującym na przesyłaniu komunikatów:

1. wywołanie zdalnego komponentu imituje lokalne wywołanie procedury  
3. warstwa pośrednia przekazuje wywołania do zdalnego komponentu

## Przy testach jednostkowych zewnętrzne zależności:

2. można pozorować (mocks)  
3. można symulować (stubs)

## Do zalet modelu warstwowego architektury systemu można zaliczyć:

2. ułatwienie przyrostowego procesu tworzenia oprogramowania  
3. prostota podmiany warstwy (przy zachowaniu jej interfejsu)  
4. umożliwienie ukrycia zależności od konkretnej platformy

## Architektura zorientowana usługowo (ang. Service Oriented Architecture) to podejście polegające na:

budowaniu systemu w postaci zbioru usług, które mogą być dostarczane przez różnych dostawców.

## Według modelu wersjonowania semantycznego, zakładając że wersje systemu mają postać sekwencji X.Y.Z, wprowadzenie nowej funkcjonalności (przy zachowaniu wstecznej kompatybilności) wymaga zwiększenia wartości w sekwencji:

Y

## Do pożądanych cech zaawansowanego systemu automatyzującego proces budowy systemu należą:

1. Automatyzacja testów  
2. Minimalizacja rekompilacji  
3. Integracja z systemem zarządzania wersjami  
4. Raportowanie rezultatów

## Testy użytkowników to testy, które są wykonywane przez rzeczywistych użytkowników w środowisku rzeczywistym. Ich typy obejmują testy alfa, beta i akceptacyjne. Są stosowane na końcowych etapach rozwoju oprogramowania. Mogą być testami usterek lub testami zatwierdzającymi.

## Dług techniczny to:

2. Koszt dodatkowej pracy, jaką trzeba włożyć w realizację zadań wynikający z niskiej jakości istniejącego kodu

## Jeżeli w systemie Git zatwierdzamy zmiany w plikach, które znajdują się już pod kontrolą systemu, można wykorzystać następujące polecenia:

1. git commit; git push origin master  
3. git add; git commit  
4. git commit -a

## Wiele kopii repozytorium kontroli wersji może istnieć w systemach kontroli wersji typu:

rozproszonego

## Operacja idempotentna to operacja:

której wielokrotne wykonanie daje taki sam rezultat

## Do czynności procesu ciągłej integracji (ang. continuous integration) należą:

3. Budowa oprogramowania  
4. Automatyczne uruchamianie testów

## Minimalna liczba klas równoważności dla parametru wejściowego, który jest listą liczb całkowitych, gdzie poprawna liczba danych w liście mieści się w przedziale (2, 10), a poprawne wartości liczb znajdują się w przedziale (100, 10000), to:

9

## Do zalet modelu architektury o nazwie repozytorium należą:

1. możliwość efektywnej implementacji przechowywania dużej ilości danych  
2. centralizacja zarządzania kopiami zapasowymi   
3. centralizacja zarządzania bezpieczeństwem danych

## Testowanie metodą 'białej skrzynki' to:

inaczej testowanie strukturalne, w którym testy opracowuje się na podstawie znajomości struktury programu/komponentu

## W modelu Gitflow wdrożenie wersji systemu na środowisko produkcyjne wymaga:

dołączenia zmian z gałęzi release do gałęzi master

## W budowanym systemie istotnym elementem jest proces przetwarzania, który można podzielić na etapy i określić ich kolejność. Dodatkowo musi istnieć możliwość optymalizacji tego przepływu przez zrównoleglenie. Wzorzec architektoniczny, który można zastosować, to:

przetwarzanie potokowe

## Informacja o tym, które wersje komponentów wchodzą w skład danej wersji systemu, nazywa się:

linią bazową

## Inspekcje oprogramowania mogą dotyczyć:

1. projektu systemu  
2. specyfikacji wymagań  
3. kodu źródłowego systemu

## Jeśli krytycznymi wymaganiami są efektywność i zdatność do pielęgnacji, to:

nie można zrealizować obu tych wymagań jednocześnie na maksymalnym poziomie

## Jeśli system rozproszony postrzegany jest przez użytkowników jako system scentralizowany, to znaczy, że posiada on własność zwaną:

przezroczystością

## Komunikacja pomiędzy komponentami systemu rozproszonego może odbywać się z wykorzystaniem modelu:

1. zdalnego wywołania procedury  
2. interakcji bazującej na komunikatach  
3. warstwy pośredniej  
4. klient-serwer