

## Wytwarzanie, integracja i testowanie systemów informacyjnych



K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 12, Folia 1

### Zadania kierownictwa przedsięwzięcia programistycznego

Niezbędnym warunkiem sukcesu jest właściwe zarządzanie przedsięwzięciem.

**Podstawowe zadania kierownictwa przedsięwzięcia programistycznego:**

- ✦ Opracowanie propozycji dotyczących sposobu prowadzenia przedsięwzięcia
- ✦ Kosztorysowanie przedsięwzięcia
- ✦ Planowanie i harmonogramowanie przedsięwzięcia
- ✦ Monitorowanie i kontrolowanie realizacji przedsięwzięcia
- ✦ Dobór i ocena personelu
- ✦ Opracowanie i prezentowanie sprawozdań dla kierownictwa wyższego szczebla

Sposoby zarządzania przedsięwzięciem programistycznym nie różnią się od zarządzania innymi przedsięwzięciami, chociaż posiadają swoją specyfikę, np. nieprzejrzystość procesu budowy oprogramowania.

K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 12, Folia 2

## Czynniki psychologiczne w inżynierii oprogramowania

Czynniki te wynikają z faktu, że oprogramowanie jest używane i tworzone przez ludzi.

**Użytkowanie** - implikuje zasady tworzenia interfejsu użytkownika i dokumentacji użytkowej,

**Tworzenie** - zagadnienia psychologiczne odgrywające rolę w tworzeniu oprogramowania.

Sposób pracy sztucznych systemów (opartych o tzw. "sztuczną inteligencję") różni się zasadniczo od sposobu pracy twórczej ludzkiego umysłu. Przenoszenie zasad sztucznej inteligencji (takich jak ścisłe dedukcyjne rozumowanie, rozbijanie problemu na podproblemy) na grunt inżynierii oprogramowania okazało się fałszywą nadzieją.

### Elementy ludzkiej inteligencji wg. Dreyfusa & Dreyfusa:

- ✦ Umiejętność całościowego (syntetycznego) spojrzenia na problem.
- ✦ Posługiwanie się wiedzą płynącą z doświadczenia, a więc stosowania nieścisłych zasad wnioskowania na bazie wcześniejszych doświadczeń.

Nie zostały odkryte zasady umożliwiające naśladowanie inteligencji przez sztuczne systemy. Tzw. "sztuczna inteligencja" nie istnieje: pierwszy lepszy pies jest znacznie bardziej inteligentny od najbardziej "inteligentnych" systemów komputerowych (jeżeli pominać mechaniczne naśladowanie przez system inteligencji jego twórców.) Stąd wynikają ogromne trudności sformalizowania procesów wytwarzania oprogramowania.

## Osobowość twórców oprogramowania

Istnieją ogromne różnice w predyspozycjach osób dotyczące ich efektywności w produkcji oprogramowania. Wydajność może się różnić o rząd lub więcej.

### Testy osobowości:

metody określenia, czy dana osoba posiada cechy przydatne na danym stanowisku.

### Stosowanie testów osobowości wiąże się z następującymi trudnościami:

- ✦ Osobowość ludzka ma charakter dynamiczny (zmienia się). Wieleletnia praktyka zawodowa nie pozostaje bez wpływu na osobowość. Część cech osobowości może być nabyta i nie da się odkryć wstępnymi testami.
- ✦ Różne zadania mogą wymagać różnych cech osobowości. Inne powinien posiadać analityk (kontakt z klientem), inne zaś programista lub osoba testująca oprogramowanie. Ponadto, metody inżynierii oprogramowania ulegają zmianie, co pociąga za sobą inny stosunek pożądaných cech osobowości do aktualnych zadań.
- ✦ Osoby poddane testom będą starały się raczej odgadnąć pożądaną przez testujących odpowiedź niż odpowiadać zgodnie ze stanem faktycznym. Test nie będzie więc odzwierciedlał cech osobowości osoby, lecz raczej to, jak ta osoba wyobraża sobie cele i kryteria testowania oraz cechy pożądane przez pracodawcę.

## Cechy dobrego inżyniera oprogramowania

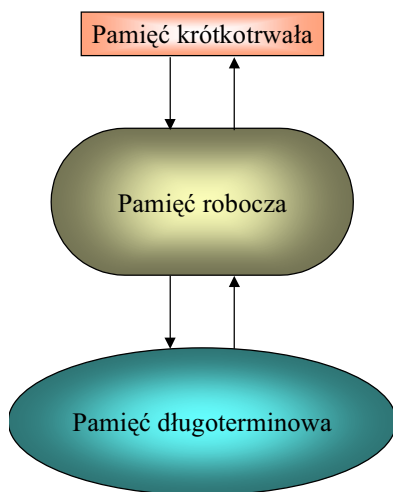
- ✦ **Umiejętność pracy w stresie.** W pracy często zdarzają się okresy wymagające szybkiego wykonania złożonych zadań. Dla większości osób niewielki stres działa mobilizująco. Po przekroczeniu jednak pewnego progu następuje spadek możliwości danej osoby. Próg ten jest różny dla różnych osób.
- ✦ **Zdolności adaptacyjne.** Informatyka jest jedną z najszybciej zmieniających się dziedzin. Ocenia się, że 7-9 miesięcy przynosi w informatyce zmiany, które w innych dziedzinach zajmują 5-7 lat. Oznacza to konieczność stałego kształcenia dla wszystkich inżynierów oprogramowania - stałe poznawanie nowych narzędzi, sprzętu, oprogramowania, technologii, metod, sposobów pracy. Niestety, nie wszyscy to tempo wytrzymują. (Uśpienie, zajmowanie się jednym problemem w jednym środowisku przez lata jest w informatyce bardzo groźne!)



K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 12, Folia 5

## Organizacja pamięci ludzkiej

Na jakość pracy inżyniera oprogramowania ma wielki wpływ organizacja pamięci ludzkiej.



Przechowuje ostatnio otrzymane informacje, np. ostatnio przeczytane zdanie. Są tam także informacje, o których aktualnie myślimy. Pojemność jej jest niewielka, stąd np. reguła  $7 \pm 2$ . Wielkość tej pamięci ma duży wpływ na nasze możliwości pracy nad złożonym problemem.

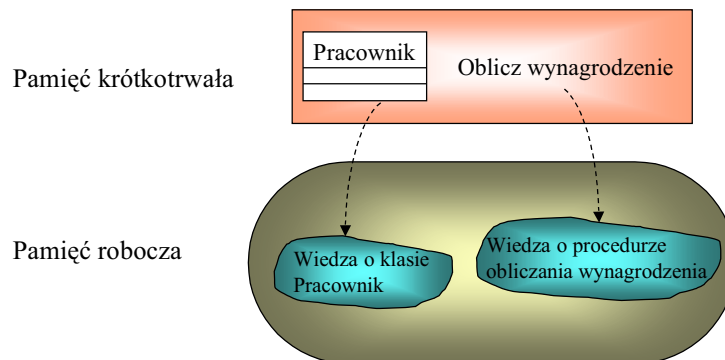
Pamięć o znacznie większej pojemności. Przechowuje informacje, które są łatwo dostępne. Przechowywane są np. fakty, nad którymi dana osoba aktualnie pracuje.

Pamięć długoterminowa ma ogromną pojemność. Informacje w niej zawarte nie są jednak dostępne w każdej chwili, niekiedy przypomnienie faktu wymaga czasu i wysiłku. Zapomniane fakty zwykle nie są wymazywane z tej pamięci, lecz stają się trudniej dostępne. Konieczne jest przypominanie, powtarzanie faktów, aby je utrwalić w pamięci długoterminowej.

K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 12, Folia 6

## Rola skojarzeń w pracy nad złożonymi problemami

Możliwość budowania skojarzeń jest cechą ludzkiego umysłu i znacznie wzmacnia zarówno objętość zapamiętywanej informacji jak i szybkość dostępu.



Umysł ludzki sprawnie kojarzy wiedzę o klasie pracownik i wiedzę o procedurze obliczającej wynagrodzenie pracownika. Posługiwanie się rysunkami i czytelnymi nazwami zdecydowanie ułatwia tworzenie tego typu skojarzeń.

K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 12, Folia 7

## Rodzaje wiedzy

- ✦ **Wiedza składniowa.** Polega na mechanicznym zapamiętaniu pewnych faktów, bez ich istotnego przetworzenia. Jest słabo zintegrowana z wcześniej zdobytą wiedzą. Np. do takiej wiedzy zaliczamy reguły składniowe danego języka programowania.
- ✦ **Wiedza semantyczna (znaczeniowa).** Fakty są zapamiętane nie w postaci ich formy, lecz w postaci znaczenia. Np. znajomość zasady instrukcji *while*, znajomość koncepcji pojęcia klasy i dziedziczenia, itd. Nowa wiedza jest zintegrowana z wcześniej zdobytą wiedzą.

**Istnienie tych dwóch rodzajów wiedzy może mieć wpływ na politykę kadrową.**

Np. pracownik rozumiejący zasady obiektowości (wiedza semantyczna) może lepiej sobie poradzić niż pracownik dobrze znający składnię i reguły użycia poszczególnych konstrukcji C++ (wiedza składniowa).

**Firmy przywiązują zbyt wielką wagę do wiedzy składniowej**, np. znajomości konkretnych języków i systemów. W istocie, ta wiedza może być stosunkowo szybko nabyta (kilka tygodni przeciętnie na opanowanie nowego języka). Natomiast wiedza semantyczna może być przedmiotem lat studiów i doświadczeń.

K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 12, Folia 8

## Nastawienie osób do pracy w zespole

Czynniki psychologiczne mają zasadniczy wpływ na efektywność pracy zespołu. Wyróżnia się następujące typy psychologiczne:

- ✦ **1. Zorientowani na zadania** (*task-oriented*). Osoby samowystarczalne, zdolne, zamknięte, agresywne, lubiące współzawodnictwo, niezależne.
- ✦ **2. Zorientowani na siebie** (*self-oriented*). Osoby niezgodne, dogmatyczne, agresywne, zamknięte, lubiące współzawodnictwo, zazdrosne.
- ✦ **3. Zorientowani na interakcję** (*interaction-oriented*). Osoby nieagresywne, o niewielkiej potrzebie autonomii i indywidualnych osiągnięć, pomocne, przyjazne.

Osoby typu 1 są efektywne, o ile pracują w pojedynkę. Zespół złożony z takich osób może być jednak nieefektywny. Lepsze wyniki dają zespoły złożone z typów 3. Typ 1 i 2 może być także efektywny w zespole, o ile jest odpowiednio motywowany przez kierownictwo. Typy 3 są konieczne w fazie wstępnej wymagającej intensywnej interakcji z klientem.

K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 12, Folia 9

## Lojalność grupowa

Terminem tym określa się silny, osobisty związek pomiędzy poszczególnymi członkami zespołu, grupą i wynikami pracy grupy. Niekorzystne efekty:

- ✦ **Trudność zmiany lidera**. Silnie związana grupa nie akceptuje nowego lidera narzuconego z zewnątrz. Często jednak formalny lider źle kieruje pracą.
- ✦ **Myślenie grupowe** (*groupthink*). Brak krytycyzmu w stosunku do efektów pracy grupy, nie rozważanie jakichkolwiek pomysłów i rozwiązań nie pochodzących z wnętrza grupy, wzajemne podtrzymywanie się w poglądach, często niesłusznych lub tendencyjnych. Rezultatem jest znaczny spadek jakości wyników pracy.

### Walka z myśleniem grupowym:

- ✦ **Sesje krytyki**, podczas których dozwolona jest jedynie krytyka przyjętych rozwiązań, natomiast zabroniona jest jakakolwiek obrona osiągnięć grupy.
- ✦ **Włączanie do zespołu krytycznych osobowości** - osób o szczególnych zdolnościach do wyszukiwania błędów i kwestionowania przyjętych rozwiązań (tzw. "czepialskich"). Osoby te są zwykle nie lubiane, co często nie stanowi dla nich problemu.

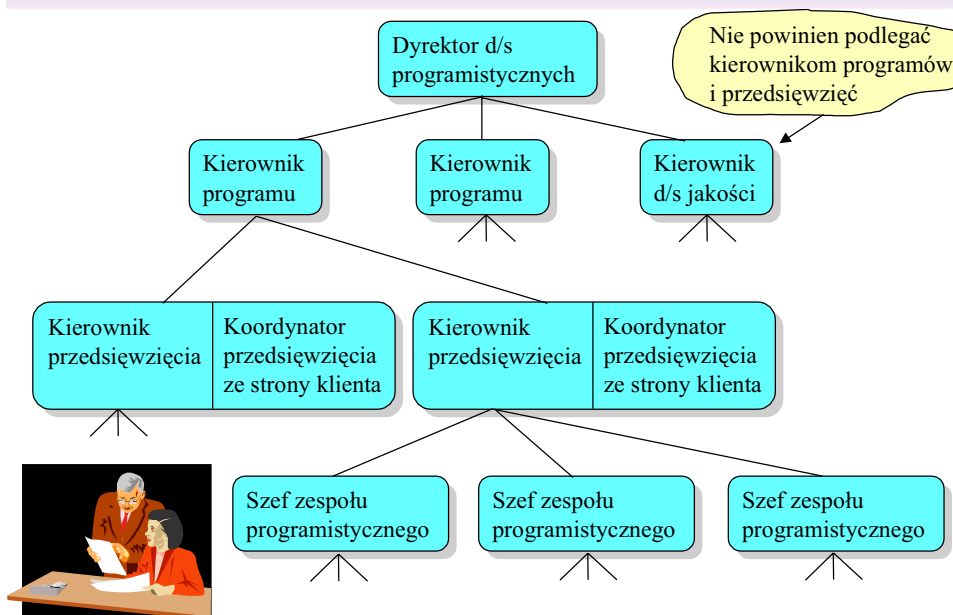
K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 12, Folia 10

## Ergonomia pracy

- ✦ Zamiast dużej hali, lepsze wyniki daje umieszczenie dwóch-trzech stanowisk pracy w wielu mniejszych pomieszczeniach.
- ✦ Personalizacja stanowiska pracy.
- ✦ Pokój zebrań dla organizowania formalnych spotkań pracowników.
- ✦ Miejsce dla spotkań nieformalnych (np. omówienie spraw przy kawie).
- ✦ Poczucie pracy na nowoczesnym sprzęcie. Wydajność i chęć ludzi do pracy gwałtownie spada, jeżeli odczuwają oni, że pracują na przestarzałym sprzęcie - nawet wtedy, gdy wymiana sprzętu jest merytorycznie nieuzasadniona.
- ✦ Komfort psychiczny, właściwa atmosfera w pracy, eliminacja napięć i zadrażeń, nie dopuszczanie do rozmycia odpowiedzialności, sprawiedliwa ocena wyników pracy poszczególnych członków zespołu, równomierny rozkład zadań.

K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 12, Folia 11

## Struktura zarządzania firmą programistyczną



K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 12, Folia 12

## Funkcje osób pracujących nad oprogramowaniem

- ✦ **Kierownik programu/przedsięwzięcia**
- ✦ **Analityk** - osoba bezpośrednio kontaktująca się z klientem, której celem jest określenie wymagań i budowa modelu systemu
- ✦ **Projektant** - osoba odpowiedzialna za realizację oprogramowania. Może posiadać bardziej wyspecjalizowane funkcje:
  - Projektant interfejsu użytkownika
  - Projektant bazy danych
- ✦ **Programista** - osoba implementująca oprogramowanie
- ✦ **Osoba wykonująca testy**
- ✦ **Osoba odpowiedzialna za konserwację oprogramowania**
- ✦ **Ekspert metodyczny** - osoba szczególnie dobrze znająca stosowaną metodykę
- ✦ **Ekspert techniczny** - osoba szczególnie dobrze znająca sprzęt i narzędzia

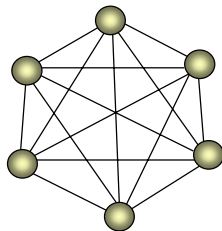


**Model analityk/projektant + programista:** funkcje analizy i projektu w jednych rękach, funkcje programisty dość niskiego poziomu. W warunkach polskich model nie zdaje egzaminu.  
**Model analityk + projektant/programista:** Model bardziej realistyczny.

K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 12, Folia 13

## Organizacja zespołu programistycznego

Struktura sieciowa

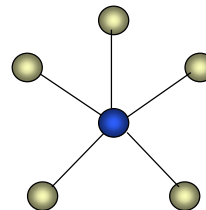


### Zalety:

- ✦ Dzięki ścisłej współpracy członkowie zespołu wzajemnie kontrolują swoją pracę. Szybko osiągane są standardy jakości.
- ✦ Umożliwia realizację idei wspólnego programowania
- ✦ Ponieważ praca członków zespołu jest znana dla innych członków, łatwo mogą oni przejąć obowiązki pracownika, który opuścił zespół.

Struktura sieciowa nie może liczyć więcej niż 8 osób.

Struktura gwiazdzista



Jest przydatna wtedy, gdy w skład zespołu wchodzi wielu niedoświadczonych pracowników. Szef kontroluje i koordynuje pracę.

Wielkość zespołu może być znacznie większa niż w strukturze sieciowej.

Duże problemy w momencie odejścia szefa zespołu.

K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 12, Folia 14

## Zapewnianie jakości

Nie powinno być mylone z testowaniem oprogramowania:

**“Zapewnienie jakości to zestaw procedur, technik i narzędzi mających zapewnić, że tworzony produkt spełnia narzucone standardy. Jeżeli standardy nie są jawnie określone, zapewnienie jakości oznacza zaspokojenie minimalnych, rynkowych wymagań jakości.” (Bersoff, 1984)**

### Kryteria jakości:

- Zgodność z wymaganiami użytkownika
  - Efektywność
  - Łatwość konserwacji
  - Ergonomiczność
- ✦ Na jakość oprogramowania wpływają działania podejmowane we wszystkich fazach jego życia.
  - ✦ Istotne jest określenie kryteriów jakości i ich priorytetu. Kryteria te powinny być zawarte w dokumencie zwanym planem jakości.
  - ✦ Jakość produktu jest silnie związana z jakością procesu, który go wytwarza. Na tym założeniu oparte są normy ISO 900x

## Poziomy rozwoju firmy programistycznej

### Pięć poziomów rozwoju organizacji z punktu widzenia dojrzałości procesu:

- ✦ **Początkowy.** Na tym poziomie nie istnieją żadne standardy procesu. Decyzje są podejmowane *ad hoc*. Ten poziom mogą mieć również firmy o dobrym zaawansowaniu technicznym.
- ✦ **Powtarzalny.** Poszczególne przedsięwzięcia wykonywane są w podobny sposób. W firmie istnieje co do tego zgoda, można więc mówić o pewnym standardzie firmy, które są jednak standardami *de facto*. Standardy te nie są udokumentowane. Nie istnieją ścisłe procedury kontroli.
- ✦ **Zarządzany.** Standardy postępowania są dobrze zdefiniowane i sformalizowane. Istnieje ścisła kontrola przestrzegania standardów. Są osoby odpowiedzialne za opracowanie i uaktualnianie standardów.
- ✦ **Mierzony.** Proces nie tylko podlega kontroli na zgodność ze standardami, ale jest mierzony w sposób ilościowy. Mierzona jest np. wydajność. Wyniki są wykorzystywane do poprawy sposobów realizacji przyszłych przedsięwzięć.
- ✦ **Optymalizowany.** Standardy są w ciągły sposób uaktualniane tak, aby uwzględnić doświadczenia w przyszłych przedsięwzięciach. Standardy zawierają elementy pozwalające na dostosowanie procesu do aktualnych potrzeb.



## Dokumentacja procesu wytwarzania

W trakcie trwania przedsięwzięcia powstają następujące dokumenty:

- Dokumentacja procesu produkcji oprogramowania.
- Dokumentacja techniczna opisująca wytworzony produkt.

### Dokumentacja procesu:

- ✦ **Plany, szacunki, harmonogramy** - dokumenty tworzone przez kierownictwo przedsięwzięcia. Odbiorcami ich są przełożeni wyższego szczebla. Zaakceptowane dokumenty tego typu pełnią rolę poleceń dla wykonawców.
- ✦ **Raporty** - Dokumenty przygotowywane przez kierowników dla przełożonych. Opisują przebieg i rezultaty prac.
- ✦ **Standardy** - dokumenty opisujące pożądany sposób realizacji
- ✦ **Dokumenty robocze** - rozmaite dokumenty zawierające propozycje rozwiązań. Twórcami są członkowie zespołu. Zaakceptowane mogą stać się standardami.
- ✦ **Komunikaty** - rozmaite, z reguły krótkie dokumenty służące do wymiany informacji pomiędzy członkami zespołu.

K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 12, Folia 17

## Dokumentacja techniczna

Składowe dokumentacji technicznej został omówione poprzednio.

Dokumentacja techniczna przed oddaniem oprogramowania do eksploatacji powinna być poddana weryfikacji celem wyeliminowania błędów i nieścisłości.

### Istotne jest wypracowanie w firmie standardów dokumentacji technicznej:

- ✦ Procesów wytwarzania dokumentacji: tworzenia wstępnej wersji dokumentów, wygładzania, drukowania, powielania, oprawiania, wprowadzania zmian w istniejących dokumentach. Konieczne jest ścisłe określenie odpowiedzialnych za to osób.
- ✦ Treści i formy dokumentów: strona tytułowa, spis treści, budowa rozdziałów, podrozdziałów i sekcji, indeks, słownik.
- ✦ Sposobu dostępu do dokumentacji: niezbędne jest stworzenie rodzaju biblioteki dokumentów technicznych, z zapewnieniem sprawnego dostępu do dowolnego dokumentu.



K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 12, Folia 18

## Zarządzanie wersjami

Produkt oddany do eksploatacji musi podlegać zmianom. Każda modyfikacja oznacza powstanie wersji systemu, mniej lub bardziej różnej od wersji poprzedniej. Niektórzy klienci mogą nie chcieć zmiany oprogramowania, co implikuje istnienie wielu wersji produktu. Inną przyczyną powstawania wersji jest zróżnicowanie potrzeb użytkowników. Np. mogą być wersje będące kombinacją modułów oprogramowania. Jeszcze inną przyczyną jest istnienie wielu platform sprzętowych i systemów operacyjnych.

### Konieczne jest opracowanie systemu zarządzania wersjami, zawierającego:

- ✦ Informację o wszystkich wykonanych i oddanych do eksploatacji wersjach
- ✦ Informację o klientach, którzy nabyli daną wersję
- ✦ Wymagania sprzętowe i programowe poszczególnych wersji
- ✦ Informację o składowych (klasach, encjach, modułach) wchodzących w skład danej wersji
- ✦ Informację o zadaniach zmian w stosunku do danej wersji
- ✦ Informację o błędach wykrytych w poszczególnych wersjach.

## Miary produktywności

**Konieczna jest ocena poszczególnych pracowników** ze względu na:

- Konieczność odpowiedniego motywowania najbardziej wydajnych osób
- Możliwość wykorzystania zebranych danych do szacowania przyszłych zadań

### Tradycyjne miary produktywności:

- Liczba linii uruchomionego i przetestowanego kodu źródłowego (bez komentarzy) napisanych np. w ciągu miesiąca.
- Liczba instrukcji kodu wynikowego wyprodukowanego w pewnym okresie czasu
- Liczba stron dokumentacji napisanej w pewnym okresie czasu
- Liczba przykładów testowych opracowanych w pewnym okresie czasu

Stosowanie nowoczesnych narzędzi może utrudnić lub uniemożliwić posługiwanie się tymi miarami:

- (1) Języki wysokiego poziomu => krótki kod, duża wydajność;
- (2) Gotowe biblioteki, generatory => duża liczba instrukcji w krótkim czasie.

Miary te mogą także doprowadzić do tendencyjnego wypaczenia procesów produkcji oprogramowania, o ile będą preferowały długość kodu. Miary te są bardzo trudne do zastosowania dla analityków i projektantów.

## Harmonogramowanie przedsięwzięć

### Polega na:

- Ustaleniu kalendarza prac
  - daty rozpoczęcia przedsięwzięcia
  - dni roboczych i wolnych w przewidywanym okresie realizacji przedsięwzięcia
  - czasu pracy w poszczególnych dniach
- Podziale przedsięwzięcia na poszczególne zadania
- Określenie parametrów zadań
- Określenie zasobów niezbędnych do realizacji poszczególnych zadań
- Ustaleniu dostępności zasobów
- Ustaleniu kolejności i czasów wykonania poszczególnych zadań



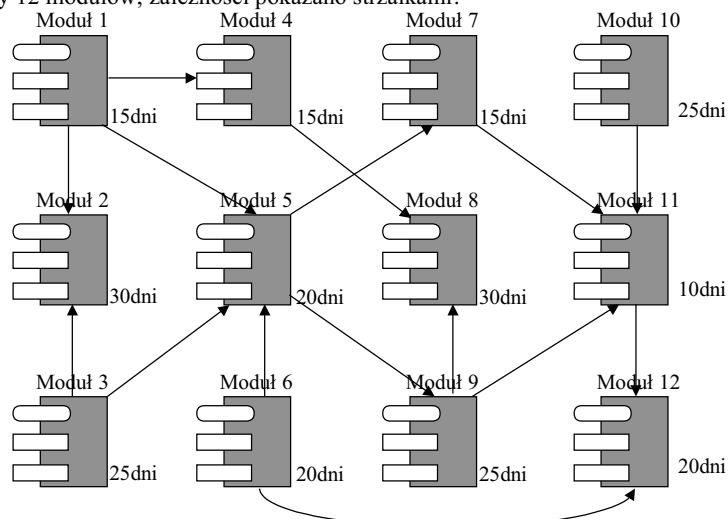
### Po ustaleniu zadań konieczne jest określenie parametrów czasowych:

- czasu wykonania
- najwcześniejszy możliwy termin rozpoczęcia
- pożądany czas zakończenia
- innych ograniczeń, np. zadań których zakończenie jest niezbędne do rozpoczęcia nowych zadań.

K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 12, Folia 21

## Przykładowy diagram zależności

Np. mamy 12 modułów; zależności pokazano strzałkami:



Tego rodzaju diagram podlega analizie ścieżki krytycznej określanej jako PERT.

Diagramy Gantt'a, diagramy ograniczeń zasobów - inne metody ustalenia harmonogramów.

K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 12, Folia 22

## Ekonomiczne aspekty działalności firmy

Jakość produktu jest tylko jednym z czynników wpływających na wynik ekonomiczny firmy. Inne aspekty:

- Reklama i promocja produktu
- Renoma producenta
- Rodzaj i zakres gwarancji oraz innych usług dla klientów
- Przyzwyczajenia klientów
- Sposób wyceny rozmaitych wersji produktu.

Na wielkość zysków wpływają także koszty własne poniesione przy produkcji:

- Inwestycje niezbędne wewnątrz firmy
- Koszty reorganizacji firmy
- Koszty szkoleń
- Koszty zakupu narzędzi CASE
- Nakłady na dokładne testowanie oprogramowania

**Zwrot nakładów następuje zwykle po pewnym czasie i często nie może być traktowany jako pewny.**

