dr hab. MAREK J. GRENIEWSKI
Zakład Doświadczalny Oprogramowania przy IMM

WYTWARZANIE OPROGRAMOWANIA METODAMI PRZEMYSŁOWYMI

1. Wprowadzenie do problematyki produkcji oprogramowania

Prawdopodobnie pierwszymi pytaniami, jakie nasuną się Czytelnikowi, będą pytania:
- Czy tytuł nie jest jakimś nieporozumieniem? Czy istnieje coś takiego, jak "produkt programowy"? Programowanie to chyba usługa? W niniejszym artykule autor będzie starał się przekonać Czytelnika, że można mówić o produkcji oprogramowania oraz że oprogramowanie tak wytworzone ma cechy produktu przemysłowego.

Rozwinięcie zastosowań komputerów wiąże się ściśle z koniecznością stworzenia odpowiednio elastycznego i bogatego oprogramowania. Stworzenie tego rodzaju oprogramowania wymaga przejścia na przemysłowe metody wytwarzania oprogramowania.

Przemysłowe metody wytwarzania oprogramowania opierają się o:

- technologię wytwarzania oprogramowania,
- technologiczną konstrukcję oprogramowania,
- oprzyrządowanie programowe,
- sprzęt,
- programistów profesjonalnych.

Technologia wytwarzania oprogramowania dotyczy oprogramowania przeznaczonego do wielokrotnego wykorzystania przez różnych użytkowników. Ma ona na celu zapewnienie:

- uniwersalności oprogramowania,
- odpowiedniej jakości i efektywności oprogramowania,
- możliwości wprowadzenia modyfikacji i dalszej rozbudowy,
- zwiększenia wydajności pracy programistów profesjonalnych,
- szczegółowego planowania prac i na tej drodze - zmniejszenie niepewności związanej z terminami i kosztami wytworzenia oprogramowania,
- standaryzacji form dokumentacji.

Należy podkreślić, że znajomość technologii wytwarzania oprogramowania w Polsce jest niemal zerowa. Pewne doświadczenia w tym zakresie wynikające z ponad rocznej działalności Zakładu Doświadczalnego Oprogramowania przy Instytucie Maszyn Matematycznych ma Zjednoczenie Przemysłu Automatyki i Aparatury Pomiarowej "MERA".

Technologiczne konstrukcje oprogramowania są to konstrukcje, składające się z typowych elementów łączonych w standardowy sposób i charakteryzujące się tym, że dają się podzielić na komponenty /zespoły główne/, które mogą być niezależnie projektowane, uruchamiane, integrowane i testowane.

Programowe oprzyrządowanie procesu wytwarzania oprogramowania składa się między innymi z następujących języków i systemów programowania:

- języków i kompilatorów przenaszalnego oprogramowania,
- systemu bazy danych i pakietów manipulacyjno-dokumentacyjnych typu PPL /Program Production Library/ - niezbędnych do stosowania metod programowania strukturalnego,
- systemów modelowania struktury produktu,
- biblioteki typowych modułów działającej pod nadzorem systemu manipulacyjno-dokumentacyjnego,
- systemów testujących współdziałających z systemem manipulacyjno-dokumentacyjnym,

- systemów wspomaganego projektowania i budowania systemów zastosowaniowych /tzw. Meta-Systemy/.

Należy podkreślić, że obecny stan programowego oprzyrządowania procesu wytwarzania oprogramowania jest ciągle jeszcze bliski zera. Potrzeba jeszcze kilku lat intensywnych prac, aby stworzyć odpowiednio bogate oprzyrządowanie programowe.

Niezbędny sprzęt do wytwarzania oprogramowania, to przede wszystkim odpowiednie komputery wyposażone w terminale programowane, pamięci masowe wielkiej pojemności, urządzenia do automatyzacji składu drukarskiego, graficznego zobrazowania itd. Ponadto urządzenia powielające, drukarskie, introligatorskie itp. Ocenia się, że dla zorganizowania procesu wytwarzania oprogramowania metodami przemysłowymi, dla jednego programisty zatrudnionego przy wytwarzaniu oprogramowania niezbędne jest zainstalowanie sprzętu wartości rzędu 2 mln zł. Tak więc, jednostka produkcji oprogramowania zatrudniająca 250 programistów winna posiadać sprzęt o wartości rzędu 450 - 500 mln zł. Należy podkreślić, że sprzęt taki powinien ulegać odnowie w okresach 4-5 lat.

Sprzęt i oprzyrządowanie programowe tworzą łącznie instalację zwaną Linią Produkcji Oprogramowania.

Przemysłowe metody wytwarzania oprogramowania wymagają wykształcenia nowego typu programisty, zwanego dalej programistą profesjonalnym. W odróżnieniu od dominującego obecnie typu programisty amatora, programista profesjonalny musi umieć: pracować zespołowo przy użyciu Linii Produkcji Oprogramowania i programować metodami przenaszalnymi z przestrzeganiem zasad określonych technologią wytwarzania oprogramowania. Programista profesjonalny musi także umieć biegle programować w kilku różnych językach programowania oraz posługiwać się językiem dokumentacyjnym.

Przedstawione w niniejszym artykule problemy organizacji produkcji oprogramowania dotyczą oprogramowania przeznaczonego do wielokrotnego wykorzystania.

2. Produkt programowy

W dalszym ciągu niniejszego artykułu każdy system programowania lub pakiet programów wytwarzany metodami przemysłowymi będziemy nazywali produktem programowym. Produkt składa się z następujących części:
- wymagań technicznych na produkt,
- kodu źródłowego produktu / x /,
- kodu wynikowego produktu w postaci dystrybucyjnej,
- kodu źródłowego specjalnego oprzyrządowania programowego /nie zawsze występuje/ /x /,
- kodu wynikowego specjalnego oprzyrządowania programowego /nie zawsze występuje/,
- specyfikacji produktu,
- krótkiego opisu wprowadzającego,
- przewodnika użytkowania,
- przewodnika konserwacji / x /,
- instrukcji dla programisty systemowego,
- instrukcji dla operatora,
- instrukcji przygotowania danych,
- dokumentacji konstrukcyjnej produktu / x /,
- programu szkolenia użytkownika,
- pomocy audiowizualnych do szkolenia,
- przykładów do demonstracji,
- zadań kontrolnych.

Przedmiotem sprzedaży mogą być wszystkie części produktu, w przypadku gdy producent nie prowadzi konserwacji produktu we własnym zakresie /x/ lub - części nie oznaczone gwiazdką - w przypadku gdy producent prowadzi konserwację. Udział kosztów w poszczególnych etapach kształtuje się dla systemów operacyjnych i systemów przetwarzania danych jak w tablicy 1.

Tablica 1

Nr etapu	Nazwa etapu wytwarzania produktu	Procentowy udział w kosztach
1	Rozpoznanie i opracowanie wymagań	10%
2	Projektowanie koncepcyjne	13%
3	Test alfa	6%
4	Projektowanie szczegółowe, kodowanie, uruchamianie, integracja i dokumentowanie	65% (20%)
5	Test beta	26%
6	Przygotowanie do rozpowszechniania	5%
7	Konserwacja produktu	20%
	Razem	100%

Przez cykl wytwarzania produktu rozumie się czas realizacji etapów od 1 do 6 włącznie.

Przykładowo, wg dotychczasowych doświadczeń cykl wytwarzania translatorów i programów sterujących systemu operacyjnego wynosi średnio 24 miesiące, przy czym w zależności od stopnia złożoności zadania może się wahać od 10 do 44 miesięcy; zaś cykl wytwarzania systemów zastosowaniowych i pakietów programów wynosi średnio 18 miesięcy, przy czym w zależności od

stopnia złożoności zadania może się wahać od 8 do 27 miesięcy.

Każdy produkt składa się z pewnej ilości komponentów /zespołów głównych/, które ze względu na swoją strukturę logiczną oraz ilość i rodzaj urządzeń peryferyjnych, mogą być zaliczone do jednej z pięciu klas złożoności. W tablicy 2 przedstawione są przykłady komponentów zaliczonych do poszczególnych klas.

Tablica 2

Lp.	Typ komponentu	Przykład
1.	Prosty	Program drukowania sprawozdań o stanach i obrotach magazynowych
2.	Średnio złożony	Program aktualizacji zbioru danych wieloma typami rekordów
3.	Złożony	Translator
4.	Bardzo złożony	Supervizor
5.	Specjalny	Program obsługi terminali w reżimie interakcyjnym

Cena produktu jest wyznaczona przez sumowanie cen komponentów składających się na dany produkt. Z kolei cena komponentu jest określona w oparciu o rozmiar komponentu sprowadzony np. do ilości rozkazów języka typu Assembler przy użyciu tablic przeliczeniowych i cenę jednostkową jednego rozkazu dla komponentu danej klasy złożoności. Tablica 3 zawiera przykładowo współczynniki zamiany - rozmiarów komponentów programów pisanych w różnych językach.

Tablica 3

Współczynniki zamiany rozmiarów komponentów		Rozmiar komponentu pisanego			
		PLAN	ASSEMBLER	FORTRAN	COBOL
Ocena rozmiarów	PLAN	1,0	0,7	0,25	0,3
	ASSEMBLER	1,4	1,0	0,3	0,4
	FORTRAN	4,0	3,33	1,0	-
	COBOL	3,0	2,5	-	1,0

3. Proces wytwarzania produktu programowego

Zgodnie z tym co przedstawia tablica 1, proces wytwarzania produktu programowego składa się z siedmiu etapów, które zostaną poniżej kolejno omówione.

3.1. Rozpoznanie i opracowanie wymagań dotyczących produktu. Realizując ten etap, należy znaleźć odpowiedź na następujące pytania:

- Co mogłoby być produkowane?
- Do jakich produktów ma to być podobne?
- Jacy są potencjalni użytkownicy przyszłego produktu?
- Jak ocenia się wstępnie koszt produktu?

W wyniku realizacji tego etapu powstają wymagania techniczne na produkt.

Po zakończeniu etapu należy podjąć decyzję będącą odpowiedzią na pytanie: Czy można to wyprodukować? W przypadku odpowiedzi pozytywnej, rozpoczyna się realizowanie drugiego etapu.

3.2. Projektowanie koncepcyjne produktu. Realizując ten etap, należy znaleźć odpowiedź na następujące pytania:

- Jak produkt powinien być zbudowany, jaki zawiera nukleus i jakie kolejne warstwy nadbudowane?
- Jakie istniejące komponenty lub moduły biblioteczne mogą być wykorzystane?
- W jakich językach programowania produkt powinien być pisany?
- Jak produkt powinien być testowany?
- Jakie są rozmiary i skala trudności poszczególnych komponentów produktu?

W wyniku realizacji tego etapu powstają pierwsze wersje: specyfikacji produktu, dokumentacji konstrukcyjnej produktu i wymagań na specjalne oprzyrządowanie programowe produktu.

3.3. Test alfa. Realizując ten etap, należy znaleźć odpowiedź na następujące pytania:

- Czy projektowany produkt będzie spełniał wymagania techniczne?
- Czy zaprojektowana struktura produktu jest właściwa ze względu na realizację i konserwację?
- Czy proponowana metoda testowania produktu jest adekwatna do jego struktury?
- Jak ocenia się koszty produktu?

W toku realizacji tego etapu wprowadza się szereg zmian do wyników poprzedniego etapu. W wyniku powstają kolejne wersje: kosztorysu produktu, specyfikacji produktu,

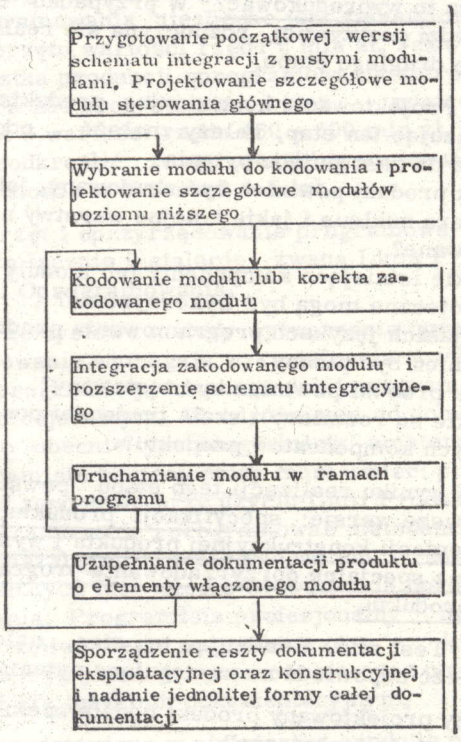
Uwaga: w przypadku, gdy dla wytworzenia produktu konieczne jest opracowanie specjalnego oprzyrządowania, cena zbytu jest obliczana łącznie, na podstawie stopnia złożoności i rozmiarów komponentów produktu i specjalnego oprzyrządowania.

dokumentacji konstrukcyjnej produktu i wymagań na specjalne oprzyrządowanie programowe produktu. Ponadto powstają wymagania na zadania kontrolne.

Po zakończeniu etapu należy podjąć decyzję będącą odpowiedzią na pytanie: Czy należy projekt realizować? W przypadku odpowiedzi pozytywnej, rozpoczyna się realizacja czwartego etapu.

3.4. Projektowanie szczegółowe, kodowanie, uruchamianie, integracja i dokumentowanie. Etap ten jest realizowany według schematu 1. W wyniku wykonania tego etapu powstają następujące części produktu:

- kod źródłowy produktu,
- kod wynikowy produktu w postaci dystrybucyjnej,
- kod źródłowy specjalnego oprzyrządowania programowego /nie zawsze występuje/,



Schemat 1 Praca przy stosowaniu metody programowania strukturalnego CPT /Chief Programmer Team/

- kod wynikowy specjalnego oprzyrządowania programowego /nie zawsze występuje/,
- przewodnik użytkowania produktu,
- przewodnik konserwacji produktu,
- instrukcja dla programisty systemowego,
- instrukcja dla operatora,
- instrukcja przygotowania danych,
- dokumentacja konstrukcji produktu

3.5. Test beta. Realizując ten etap, należy znaleźć odpowiedź na następujące pytania:

- Czy produkt spełnia wymagania techniczne?
- Czy dokumentacja użytkowa wystarcza dla posługiwania się produktem?
- Czy dokumentacja konstrukcyjna wystarcza dla konserwacji produktu?
- Ile wyniosły dotychczasowe koszty wytworzenia produktu?

W toku realizacji tego etapu opracowuje się, według wymagań, zadania kontrolne, sprawdza się poprawność i przejrzystość dokumentacji, wprowadza się konieczne zmiany, uzupełnienia itp.

Po zakończeniu etapu należy odpowiedzieć na pytanie: Jak i kiedy produkt ma być rozpowszechniany?

3.6. Przygotowanie do rozpowszechniania. Realizując ten etap, należy znaleźć odpowiedź na następujące pytania:

- Jak demonstrować produkt?
- Jak szkolić użytkowników?
- Kto i gdzie będzie szkolił użytkowników?
- Kto będzie konserwował produkt?

W wyniku realizacji etapu powstają następujące części produktu:

- krótki opis wprowadzający,
- przykłady do pokazu,
- program szkolenia użytkownika
- pomoce audiowizualne do szkolenia

Ponadto, w ramach tego etapu następuje wydanie dokumentacji w odpowiednich nakładach.

3.7. Konserwacja produktu. Jest to etap realizowany przez okres kilku lub więcej lat. W toku realizacji trzeba możliwie szybko reagować na sygnały pochodzące od użytkowników i odpowiadać na jedno z dwu pytań: Jak należy zlokalizować i usunąć wykryty błąd? Jak zwiększyć efektywność produktu?

4. Baza elementowa

Elementem konstrukcyjnym programu jest tzw. makros. Makrosy wchodzące w skład bibliotek i przeznaczone do wielokrotnego wykorzystania nazywamy modułami. Makrosy lub moduły pisane w języku wysokiego rzędu dzielą się na:

- opisujące strukturę danych,
- funkcyjne.

W dalszych rozważaniach ograniczymy się do przedstawienia jedynie tych ostatnich, nazywając je w skrócie elementami funkcyjnymi. Każdy element /makros lub moduł/ funkcyjny posiada jedno wejście i jedno wyjście. Przez wejście i wyjście rozumiemy trwałe przekazanie sterowania. Każdy element funkcyjny, ma swoją nazwę jednoznaczną w obrębie komponentu. Rozmiar elementu funkcyjnego nie przekracza 50 - 100 linii kodowych w języku wysokiego rzędu. Stero-

wanie do elementu funkcyjnego może być przekazane:

a/ Z ustawieniem śladu, według którego nastąpi powrót do elementu wołającego po wykonaniu czynności realizowanych przez dany element funkcyjny.

b/ Z ustawieniem śladu, według którego nastąpi powrót do elementu wołającego dopiero po wykonaniu czynności realizowanych przez wskazany element funkcyjny, z przekazaniem sterowania do kolejnego elementu funkcyjnego umieszczonego za danym elementem po jego realizacji.

c/ Bez ustawiania śladu, z przekazaniem sterowania do kolejnego elementu funkcyjnego umieszczonego za danym elementem po jego realizacji.

d/ Bez ustawienia śladu, z przekazaniem sterowania według śladu wcześniej ustawionego do realizacji danego elementu.

Elementy funkcyjne ustawione w programie w kolejnych obszarach pamięci mogą przekazać sobie wzajemnie sterowanie według zasady b/, c/ lub d/ - tworzą tzw. sekwencję elementów funkcyjnych. Dowolny element funkcyjny może zawierać instrukcję /jedną lub więcej/ wołania dowolnego elementu funkcyjnego lub sekwencji elementów funkcyjnych. Przez wołanie rozumiemy chwilowe przekazanie sterowania do danego elementu lub sekwencji elementów z powrotem przez ślad.

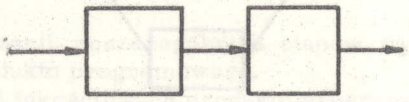
Każdy element funkcyjny jest dokumentowany według zasady przedstawionej na schemacie 2

NAZWA ELEMENTU FUNKCYJNEGO
LISTA nazw elementów funkcyjnych współpracujących bezpośrednio z danym elementem
OBSZAR DANYCH WEJŚCIOWYCH - lista nazw danych argumentów dla danego elementu funkcyjnego
OBSZAR DANYCH WYJŚCIOWYCH - lista nazw danych - wyników dla danego elementu funkcyjnego .
OBSZAR DANYCH ROBOCZYCH - lista nazw danych efemerydalnych
OBSZAR DANYCH PAMIĘTANYCH - lista nazw danych przechowywanych między kolejnymi realizacjami elementu
AKCJE - lista akcji powodujących przekazanie sterowania do elementu
FUNKCJE - opis czynności realizowanych przez dany element

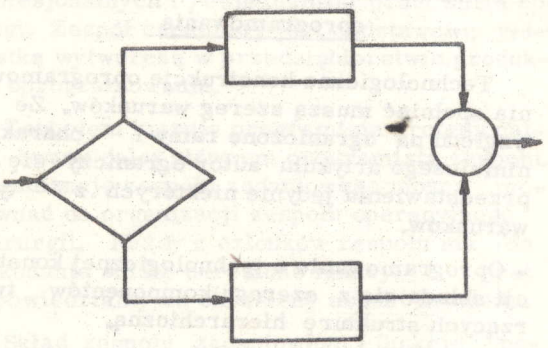
Schemat 2

Uwaga: element funkcyjny nie może zmieniać się w wyniku kolejnych realizacji.

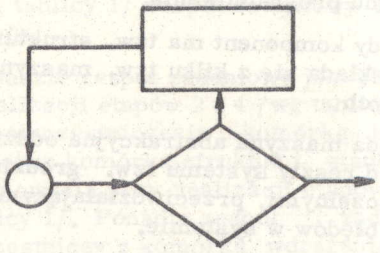
Każdy element funkcyjny może mieć jedną z pięciu struktur logicznych:
- Sekwencja operacji /schemat 3/,
- IF - THEN-ELSE /schemat 4/,
- DO - While /schemat 5/,
- DO - UNTIL /schemat 6/,
- CASE /schemat 7/



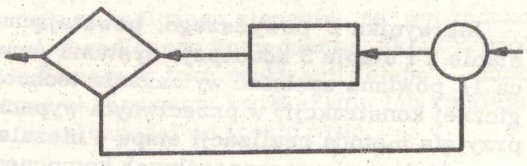
Schemat 3



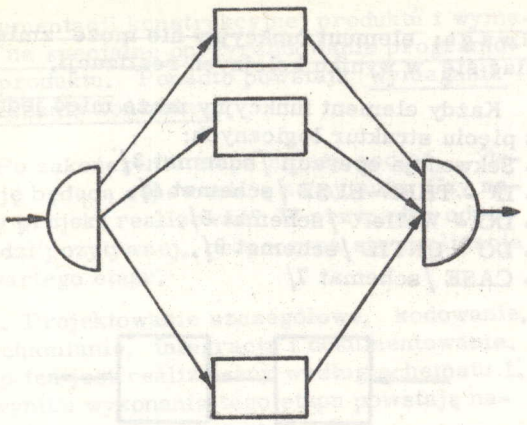
Schemat 4



Schemat 5



Schemat 6



Schemat 7

5. Technologiczne konstrukcje oprogramowania

Technologiczne konstrukcje oprogramowania spełniać muszą szereg warunków. Ze względu na ograniczone ramy i charakter niniejszego artykułu autor ograniczy się do przedstawienia jedynie niektórych z tych warunków.

- Oprogramowanie o technologicznej konstrukcji składa się z szeregu komponentów tworzących strukturę hierarchiczną.

- Komponent każdej warstwy może kontaktować się bezpośrednio co najwyżej z dwoma komponentami nadrzędnym i podrzędnym /jeśli ten ostatni istnieje/. Komponent najniższej warstwy nazywamy nukleusem danego systemu programowania.

- Każdy komponent ma tzw. strukturę ziarnistą i składa się z kilku tzw. maszyn abstrakcyjnych.

- Każda maszyna abstrakcyjna oddzielona jest od reszty systemu tzw. grodziami błędo-szczelnymi, przeciwdziałającym propagacji błędów w systemie.

- Każda maszyna abstrakcyjna realizuje jedno z typowych działań na strukturach danych

- Każda maszyna abstrakcyjna składa się z elementów funkcyjnych /omawianych w poprzednim rozdziale/.

- Podział oprogramowania na komponenty jest optymalny ze względu na założoną funkcję celu.

Jak wynika z powyższego, powstająca na etapie 2 i etapie 3 koncepcja systemu /tablica 1/ powinna spełniać wymagania technologicznej konstrukcji; w przeciwnym wypadku przyjęta metoda realizacji etapu 4 niezależnego budowania poszczególnych komponentów nie da się zastosować.

6. Standardy produkcji oprogramowania

Wprowadzenie standardów produkcji oprogramowania jest niezbędne dla zapewnienia efektywnego działania przedsiębiorstwa produkcji oprogramowania. Cele standardów są następujące:

- Zabezpieczenie użytkownika przed wytwarzaniem złej jakości produktu programowego.

- Zabezpieczenie programisty profesjonalnego przed postawieniem mu niejednoznacznie sformułowanego zadania.

- Jednoczesne określenie podziału zadań między pracownikami zatrudnionymi bezpośrednio lub pośrednio przy produkcji, wydawnictwach, szkoleniu i obsłudze użytkowników.

- Umożliwienie planowania poszczególnych etapów wytwarzania oprogramowania i kontrolę przebiegu realizacji.

- Zmniejszenie niekorzystnych dla przedsiębiorstwa produkcji oprogramowania efektów płynności kadr, przez zapewnienie możliwości przejęcia i kontynuowania pracy przez innych pracowników.

- Umożliwienie szybkiej modyfikacji istniejących produktów i wykorzystanie w nowych produktach komponentów lub modułów produktów wcześniej wytwarzanych.

- Stworzenie podstaw elastycznego organizowania zespołów zadaniowych.

- Ułatwienie szkolenia nowej kadry przedsiębiorstwa produkcji oprogramowania.

Standardy produkcji oprogramowania dzielą się na pięć grup:

a/ Standardy etapów i czynności wytwarzania produktu programowego,

b/ Standardy kompletacji i zawartości poszczególnych części produktu programowego.

c/ Standardy formułowania warunków technicznych na produkt programowy.

d/ Standardy konstrukcyjne produktu programowego.

e/ Standardy oprzyrządowania programowego.

Poniżej omówione zostaną poszczególne grupy standardów.

6.1. Standardy etapów i czynności wytwarzania produktu programowego. Przykładem standardów tej grupy są przedstawione w rozdziale 3 etapy wytwarzania produktu programowego. Oczywiste jest, że szczegółowość standardów jest znacznie większa od zakresu przedstawionego w rozdziale 3.

Do tej grupy standardów zaliczamy również schematy metodyczne pracy, takie jak schemat metody programowania strukturalnego CPT /rozdział 3/. Do tej samej grupy standardów zaliczamy tablice wydajności programisty profesjonalnego, uzupełnione kryteriami zaliczenia komponentu do danej grupy oraz tablice przeliczania rozmiarów komponentów pisanych w różnych językach programowania i tablice przeliczania wydajności programistów profesjonalnych różnych kategorii.

6.2. Standardy kompletacji i zawartości poszczególnych części produktu programowego. Przykładem standardu na skład produktu programowego jest lista kompletacyjna produktu programowego przedstawiona w rozdziale 2. Dla każdej części produktu programowego istnieje standard zawartości, określający co ma zawierać określona część produktu i w jakim układzie. Szczególnym przypadkiem standardu zawartości jest standard na komentarze kodu źródłowego produktu i kodu źródłowego oprzyrządowania specjalnego.

6.3. Standardy formułowania warunków technicznych na produkt programowy. Standardy te w jakimś sensie są rozwinięciem fragmentu standardów omawianych w pkt 6.2., a mianowicie zawartości wymagań technicznych na produkt programowy. Mają one na celu scharakteryzowanie ilościowo-jakościowe produktu programowego.

6.4. Standardy konstrukcyjne produktu programowego. Standardy konstrukcyjne mają na celu określenie struktury produktu programowego. Do tej grupy standardów zaliczamy:

- Zasadę konstruowania produktu w formie "nucleus" i nadbudowanych nad nim kolejnych warstw.
- Zasadę "ziarnistej" budowy nucleusa i warstw.
- Zasadę realizowania nucleusa i kolejnych warstw produktu jako komponentów produktu.
- Zasadę modularnego budowania komponentów zgodnie z metodą programowania strukturalnego CPT oraz bazą elementów funkcjonalnych.
- Zasady budowy bibliotek modułów i komponentów oraz ich użytkowania.
- Standardy na struktury danych.

Ponadto, do grupy tej zaliczamy standardy na łączenie modułów i komponentów produktu programowego.

6.5. Standardy oprzyrządowania programowego. Do tej grupy standardów zaliczamy standardy na reprezentacje języków dokumentacyjnych i języków programowania oraz ich wzorcowych implementacji np. COBOL ANS, FORTRAN ANS, APL, BASIC, DETAB, FORTAB, DBMS i inne/, a w szczególności standardy na makrogeneratory i ich wzorcowe implementacje /np. STAGE II, PML i inne/. Do tej grupy również zaliczamy standardy na języki manipulacji bibliotekami typu PPL i inne elementy programowego oprzyrządowania Linii Produkcji Oprogramowania.

Standardy są podstawą opracowywania metodyk, takich jak:

- Oceny rozmiarów i złożoności planowanych do wytworzenia produktów.
- Planowania wytworzenia produktu programowego.
- Realizacji poszczególnych etapów wytwarzania produktu programowego.
- Badań jakościowych produktu programowego.

7. Zespół zadaniowy

Zespół zadaniowy składa się z programistów profesjonalnych i /ewentualnie/ pracownika obsługi. Zespół zadaniowy jest podstawową jednostką wytwórczą w przedsiębiorstwie produkcji oprogramowania.

Zespołem kieruje programista profesjonalny, zwany dalej głównym programistą zespołu. Organizację zespołu zadaniowego można porównać do organizacji zespołu operacyjnego chirurgii. Każdy z członków zespołu ma do wykonania ściśle określone funkcje i jest odpowiedzialny za określony aspekt produktu.

Skład zespołu zadaniowego i funkcje poszczególnych członków są różne na różnych etapach powstawania produktu. Dlatego zespół zadaniowy może liczyć od 3 do 12 osób. Na etapach 4 i 5 /wg tablicy 1/ zespół zadaniowy liczy od 4 do 12 osób. Natomiast na etapach 2 i 3 /wg tablicy 1/ zespół zadaniowy liczy od 3 do 6 osób.

W zasadzie zespół zadaniowy jest powoływany do realizacji etapów 2 i 4 /wg tablicy 1/ oraz współuczestniczenia z komórką kontroli jakości oraz komórką struktur i standardów oprogramowania przy realizacji etapów 3 i 5 /wg tablicy 1/. Ponadto zespół zadaniowy współuczestniczy z komórką wdrażania produktów i komórką wydawniczą przy realizacji etapu 6 /wg tablicy 1/.

W uzasadnionych przypadkach, zespół zadaniowy może zostać powołany na etapie 1 /wg tablicy 1/ i współuczestniczyć z komórką badania rynku i komórką informacji techniczno-ekonomicznej.

Zespół zadaniowy jest powoływany dla wykonania określonego zlecenia, na wyprodukowanie określonego komponentu. Skład zespołu zadaniowego w toku realizacji etapu może ulegać okresowemu zwiększeniu lub zmniejszeniu w zależności od frontu prac.

Przedstawimy przykładowo zadania poszczególnych członków zespołu zadaniowego

przy realizacji etapu 4 /wg tablicy 1/. Etap 4 jest realizowany zgodnie ze schematem 1.

W skład minimalnego zespołu zadaniowego wchodzą:
- główny programista zespołu,
- programista asystent,
- programista analityk,
- programista operator PRL /Program Production library/.

W miarę potrzeb zespołu, zespół zadaniowy zostaje rozszerzony o dalszych programistów i ewentualnie programistów analityków.

Do zadań głównego programisty przy realizacji etapu 4 oprócz kierowania całością prac zespołu zadaniowego, należy projektowanie szczegółowej struktury logicznej produktu oraz schematu integracji komponentu.

Do zadań programisty asystenta należy kodowanie prostszych modułów i uzupełnienie dokumentacji produktu.

Do zadań programisty analityka należy projektowanie szczegółowe całej sygnalizacji komponentu i kodowanie modułów realizujących tę sygnalizację.

Do zadań programisty operatora PPL należy operowanie biblioteką /PPL/ przy użyciu terminala dla potrzeb zespołu zadaniowego, w celu integracji i dokumentowania modułów produktu.

Do zadań pozostałych programistów należy kodowanie i uzupełnianie bardziej złożonych modułów lub przygotowywanie danych testujących dla uruchomienia modułów.

8. Linia Produkcji Oprogramowania

Dotychczasowe rozważania pozwalają sprecyzować wymagania na linię Produkcji Oprogramowania /w skrócie LPO/. Na przykładzie poprzedniego rozdziału pokazano, w jaki sposób użytkuje LPO zespół zadaniowy, realizując etap 4 /tablica 1/. Z rozważań rozdziału 5 wynika, jakie wymagania stawia przed LPO realizacja etapu 2 i etapu 3.

W świetle dotychczasowych rozważań widać jasno, że LPO musi być:

- Węzłem informacyjnym zapewniającym koordynację całej działalności wytwórczej przedsiębiorstwa produkcji oprogramowania.

- Urządzeniem umożliwiającym interakcyjne manipulowanie kodami elementów funkcyjnych i dokumentacją konstrukcyjną komponentów i produktów w zakresie niezbędnym dla stosowania np. metody CPT.

- Urządzeniem umożliwiającym badanie przez modelowanie optymalności konstrukcji produktu.
- Urządzeniem umożliwiającym interakcyjne kompilowanie i generowanie w oparciu o metody przenaszalnego oprogramowania.

- Urządzeniem umożliwiającym symulowanie szerokiej klasy komputerów /tzw. zasada First Softwore Computer/.
- Urządzeniem umożliwiającym podłączenie metodą Chanel to Chanel Adapter dowolnego komputera, dla którego wytwarzane jest oprogramowanie do LPO, dla przesyłania testowanych fragmentów produktu z LPO do komputera i odwrotnie.
- Urządzeniem automatyzującym skład drukarski dokumentacji produktu.

Jak wynika z powyższego, LPO powinna składać się z następującego sprzętu:

- Szybkiego procesora wyposażonego w dużą pamięć operacyjną i stosunkowo liczne kanały.

- Pamięci dyskowej o pojemności rzędu 500 Mbajtów.

- Około 40 terminali programowanych wyposażonych w małe pamięci dyskowe, urządzenia drukujące, specjalizowane klawiatury itp.

- Około 10 terminali monitorów - ekrancwych.

- Adaptery kanałów umożliwiające podłączenia innych procesorów do LPO.

- Urządzenia do automatyzacji składu drukarskiego.

- Klasycznych urządzeń peryferyjnych.

Terminale powinny być podzielone na 4 grupy - zgodnie ze sposobem ich użytkowania:

- terminale programowane wykorzystywane przez zespoły zadaniowe programistów produkujących oprogramowanie;
- terminale do prac redakcyjnych nad dokumentacją prowadzone przy współudziale komórki wydawniczej;
- terminale programowe wykorzystywane dla przygotowania, wdrożeń i szkolenia użytkowników przez komórkę wdrażania i komórką szkolenia;
- terminale wykorzystywane dla potrzeb zarządzania produkcją.

Na Linię Produkcji Oprogramowania składać się będzie następujące oprzyrządowanie programowe:
- Specjalizowany system operacyjny dla LPO zapewniający efektywną obsługę terminali, współpracę z bibliotekami dyskowymi i dwustronne przekazywanie zadań i przyjmowanie wyników z podłączonych przez adaptery kanał-kanał pozostałych komputerów;
- System obsługi bazy danych dostosowany do współpracy z PPL dla efektywnego prowadzenia bibliotek dokumentacji i programów;
- Pakiety manipulowania elementami bibliotek /PPL/ realizujące między innymi następujące funkcje: wyszukiwanie, włączanie według zadanych kontekstów, rozrzucanie i łączenie tekstów z bibliotek;

- Zestaw niezależnych kompilatorów dla języków programowania systemowego i dokumentowania;
- Pakiet programów dla sprawdzania poprawności programów napisanych w konwencji programowania strukturalnego /metoda CPT/;
- Pakiety programów dla tworzenia dokumentacji eksploatacyjnej i konstrukcyjnej oraz jej aktualizowania;
- Systemy wspomaganego budowania systemów zastosowaniowych;
- Pakiet programów modelowania działania projektowanego produktu.

9. Uwagi końcowe

Jak wynika z dotychczasowych rozważań zorganizowanie wytwarzania oprogramowania metodami przemysłowymi jest przedsięwzięciem bardzo złożonym. Prowadzone obecnie prace nad stworzeniem w Zjednoczeniu MERA pierwszej Linii Produkcji Oprogramowania zostaną zakończone w roku 1976. Eksperymentalne wykorzystywanie LPO rozpocznie się na początku roku 1975.

W wyniku dotychczas prowadzonych prac zostały ukształtowane poglądy na problematykę produkcji oprogramowania oraz przygotowania została dostatecznie liczna kadra programistów profesjonalnych, niezbędna dla podjęcia produkcji oprogramowania metodami przemysłowymi.

Literatura

[1] A. Detailed Review of Chief Programmer Team Operations, IBM FSC 72 - 5177, Federal System Center 18100 Frederick Pike Gaithersburg, - Maryland 20760.

[2] B. W. Boehm, Software and its Impact A Quantitative Assessment Datamation. May 1973, pp 48-59.

[3] J. Dańda i A. Wiśniewski: Problematyka Projektowania Systemów Oprogramowania - Potrzeby i Dotychczasowe Rezultaty w Projektowanie Maszyn i Systemów Cyfrowych, PWN, Warszawa 1972.

[4] W. M. Turski: Projektowanie Oprogramowania Systemów Liczących w Projektowanie Maszyn i Systemów Cyfrowych, PWN, Warszawa 1972.

Spis treści

Redakcja i Zakład Małej Poligrafii: Dział Wydawnictw Przedsiębiorstwa Automatyki Przemysłowej "Mera-Pnefal", ul. Patriotów 77, 04-950 Warszawa. Tel. 12-41-71 /Red./ i 12-41-60 /ZMP/ zam. 5 /74 R-81. Nakład 1300 egz. + 100 egz.

MERA

AUTOMATYKA PRZEMYSŁOWA

APARATURA POMIAROWA

INFORMATYKA

BIULETYN

12(142)
Rok XII 1973