

realizacji; sprawa ta jest o tyle ważna, że rozpoczynanie następnych etapów jest często uzależnione od wielu decyzji cząstkowych związanych z uporządkowaniem dotychczasowych odcinków pracy.

c) wybraniu spośród swojego grona osoby bezpośrednio odpowiedzialnej za bieżące nadzorowanie przebiegu projektowania, prac organizacyjnych, szkolenia i doboru odpowiedniego personelu; pracownik ten powinien cieszyć się pełnym zaufaniem i mieć prawo podejmowania samodzielnych decyzji w imieniu kierownictwa.

Zaangażowanie kadry kierowniczej średniego szczebla w projektowaniu ma również poważne znaczenie. Od opinii tych pracowników bardzo wiele zależy. Projektanci systemu powinni ich sobie pozyskać, m.in. przez włączenie ich do współprojektowania, wspólnego dyskusowania poszczególnych rozwiązań i wariantów, rozważania trudności itp. Często błędem projektantów systemu jest niesłuszne lekceważenie tej kategorii pracowników, którzy pomimo pewnych, chwilowych braków z dziedziny ETO mają dużą wiedzę fachową w podstawowych zagadnieniach SPD. Może to powodować nawet z góry niechęć średniej kadry kierowniczej do projektowanego systemu, co może mieć wpływ na powodzenie oraz przyszłą pracę całego projektu. Często pomijany czynnik zaangażowania kadry kierowniczej utrudnia wdrożenie.

5.4. Organizacja procesu projektowania systemów automatycznego przetwarzania danych

5.4.1. Zakres i struktura procesu projektowania

W zależności od wybranych modeli i odmian SPD, a także celów, które użytkownicy chcieliby dzięki nim osiągnąć, istnieje różnorodność metod projektowania SPD. Poważny wpływ mają tu tradycje ośrodków projektowania, a także przyzwyczajenia i wykształcenie kadry kierowniczej zarówno użytkowników, jak i projektantów systemu. Z tego względu trudno jest mówić o pewnej zunifikowanej metodzie projektowania. Natomiast można wskazać pewne istotne wspólne tendencje w tym zakresie, nie podważające słuszności wypracowanych w odmiennych warunkach innych metod, a gwarantujących również osiągnięcie celu. Wprawdzie procesy projektowania inżynieryjno-ekonomicznego znane są od dawna i doczekały się wiele wartościowych uogólnień, jednakże dotyczą one przede wszystkim projektowania obiektów uprzedmiotowionych. Natomiast projektowanie SPD dotyczy zarówno obiektów uprzedmiotowionych, jak i żywych, a także — co jest może najważniejsze — środków wzajemnego komunikowania się w od dawna utrwalonych rozwiązaniach.

Przez dziesiątki lat powstawały różne metodyki projektowania. Jedną z pierwszych była (wypracowana w bankach) metoda „Laboratorium

przedsiębiorstwa" (*Labs in Business*), która polegała na laboratoryjnym przeanalizowaniu typowego obiektu i uogólnieniu praw jego działania. Grupa projektująca uczestniczyła w zarządzaniu danym obiektem (jako obserwatorzy na kolegiach i ważniejszych spotkaniach, na których zapadały decyzje) oraz w typowych jego pracach. W USA w koncernie IBM wypracowano metodykę „planowania badań organizacyjnych” o nazwie SOP (*Study Organization Plan*). Była ona dość rozbudowana i zbyt uniwersalna. Opierano się na badaniach modelu przedsiębiorstwa (*Business Model*) oraz na analizowaniu dysponowanych środków przez poszczególne komórki organizacyjne (*Resource Utilization Sheet*). Po uproszczeniu tej metodyki zaczęto lansować bardziej elastyczną jej odmianę o nazwie „selekcji badań organizacyjnych” (*Selective SOP*). W metodyce tej wymaga się mniej nakładów oraz uwzględniane jest automatyczne realizowanie poszczególnych zakresów projektu.

W 1968 r. firma NCR ogłosiła metodykę uproszczonego projektowania polegającą na dokładnym zdefiniowaniu systemu — o nazwie ADS (*Accurately Defined Systems*), wprowadzając na miejsce materiałów opisowych — konwencję uproszczonego zapisu dla definiowania:

- 1) raportów,
- 2) danych wejściowych,
- 3) algorytmów obliczeń,
- 4) statystyki dotychczasowego przetwarzania danych,
- 5) zależności logicznych.

Wiele stosowanych metodyk charakteryzuje się zbyt dużą szczegółowością i pracochłonnością. Najpoważniejszym ich mankamentem jest położenie największego nacisku na fazę analizowania istniejącego stanu SPD, którego istniejące rozwiązania szczegółowe mogą narzucić kierunek modernizacji SPD.

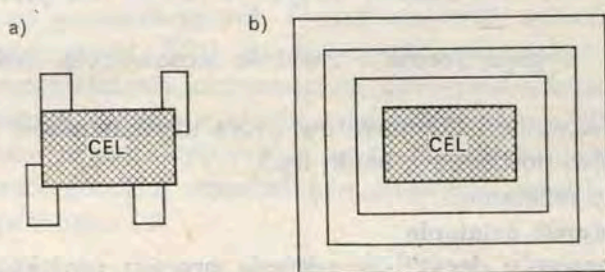
Nowoczesny pogląd na tę sprawę pomniejsza znaczenie fazy analizowania; uważa się, że wystarczy zbadać wymagania podstawowych organów wykonawczych (np. stanowisk roboczych w fabryce) i najwyższego organu zarządzania (np. dyrekcji fabryki), aby móc zaprojektować optymalne rozwiązanie dla całego obiektu.

Przez rozwój modeli SPD udowodniono, że trudno jest mówić o pewnym docelowym stanie systemu. Zaobserwowano, że w SPD tkwi nowe źródło usprawniania procesów produkcyjnych. W związku z tym zmieniły się również metody projektowania. Dotychczasowe metody usprawniały krok za krokiem sam SPD na zasadzie „rozciągania” usprawnień na coraz to nowe jego elementy (ryc. 5.3a). Obecnie natomiast wprowadzane metody projektowania charakteryzują się rozszerzaniem podstawowego celu na zasadzie „pączkowania” (ryc. 5.3b) i stałego przekraczania progów celu raz ustalonego.

Dla większości metodyk projektowania wspólne są trzy fazy projektowania:

- I — analiza działania istniejącego SPD badanego obiektu,
- II — określenie wymagań wobec modernizacji SPD (celów użytkownika),
- III — projektowanie nowego lub zmodernizowanego SPD.

Dla fazy I typowe jest nawiązanie ścisłej współpracy projektantów z kierownictwem użytkownika, zdobycie podstaw zrozumienia praw działania obiektu i wypracowanie niezbędnej ewidencji oraz dokumentacji, zapewniających efektywne współdziałanie projektantów z użytkownikiem.



Ryc. 5.3. Metodyka projektowania na zasadzie a) „rozcigania” usprawnień na coraz to nowe elementy, b) rozwijanie celu uprawnień przez „pączkowanie”

Celem fazy II jest określenie roli, jaką SPD musi spełnić, oraz określenie sprawności działania projektowanego SPD, opracowanie koncepcji obiegu oraz struktury danych i informacji, a także dostarczenie dokumentów niezbędnych w fazie III. W fazie II trzeba przeanalizować różne alternatywne rozwiązania i odpowiedzieć na podstawowe, m.in. następujące pytania:

- a) jak często są aktualizowane dane i jaką wartość mają dane aktualizowane na bieżąco?
- b) jak często występują zmiany danych i jakie są tego przyczyny?
- c) czy wymagania użytkownika odnośnie rodzajów sprawozdań są zmienne i czy te zmiany są częste?
- d) jakie są wymagania użytkownika co do okresu, w którym SPD ma odpowiedzieć na postawione przez użytkownika pytanie?
- e) jak często użytkownik ma do czynienia z kartotekami, lub jak często nie ma z nimi do czynienia z braku dostępu lub ich nieaktualności?
- f) jaką praktykę i jakie wykształcenie ma personel, który będzie korzystał z modernizowanego SPD?
- g) jakie jest przygotowanie użytkownika do sprawnego eksploataowania projektowanego SPD?

Niezbędne dla procesu projektowania jest posiadanie pewnych typowych parametrów SPD, których zmiany można by ocenić w miarę realizowania poszczególnych faz. Oto przykłady tego typu parametrów i czynników:

- koszt eksploataowania SPD,
- czas (istnieją różne odmiany obliczenia cyklu przetwarzania),
- dokładność — możliwości, rodzaje i częstotliwości pojawiania się błędów,
- niezawodność — słabe lub mocne miejsca systemu,
- elastyczność — typy i liczba różnych wariantów procedur,
- bezpieczeństwo — legalność danych, informacji i użytkowników oraz ograniczenie w udostępnianiu (tajność),
- przepustowość — średnie i szczytowe możliwości przetwarzania na komputerze,
- jakość — wygląd, forma i trwałość stosowanych nośników informacji,
- sposoby akceptacji dokumentów przez użytkowników i ich kontrahentów (klientów, dostawców, banki itp.),
- sprawność działania,
- ekonomiczność działania.

Przy podejmowaniu decyzji w zakresie procesu projektowania i jego struktury należy bezwzględnie wziąć pod uwagę środki przeznaczone na ten cel. Często użytkownicy, nie posiadając ich, podejmują prace, które technicznie i organizacyjnie są niewykonalne. Tego rodzaju decyzje mogą z góry przekreślić powodzenie projektu, który mógłby być rozpoczęty później po uzyskaniu niezbędnych środków.

W przeciwieństwie do projektowania skomplikowanych obiektów uprzedmiotowionych, duża liczba projektantów SPD nie jest wymagana. Na podstawie praktyki, m.in. zakładu ZOWAR³ można przyjąć kilka postulatów, które mogą być pomocne w organizowaniu projektowania:

- a) należy ograniczać liczbę projektantów SPD, aby zapewnić między nimi a użytkownikiem odpowiednie współdziałanie,
- b) w początkowym okresie procesu projektowania powinno się angażować od 1 do 3 projektantów, a dopiero z chwilą wypracowania koncepcji systemu (faza II) ich liczbę można stopniowo zwiększać,
- c) nowych pracowników powinno się wprowadzać stopniowo, poczynając od prac dokumentacyjnych,
- d) równolegle z rozwojem procesu projektowania należy opracowywać dokumentację, której nie można potem odtworzyć (por. pkt. 7),
- e) proces projektowania powinien być realizowany etapami, aby widoczne były rezultaty pracy, które zachęcałyby zarówno projektantów, jak i użytkownika do dalszego kontynuowania projektowania (por. prawo efektu Thorendike'a),
- f) nie powinno się przenosić zbyt często projektantów prac nad jednym projektem do drugiego; jednemu pracownikowi nie powinno się przydzielać więcej jak 1 do 3 różnych prac projektowych,

³ Zakład Obliczeniowy — Warszawa wyposażony w maszyny IBM 1440, ZAM 41 i korzystający z ICT 1300, dla których opracowano ponad 1500 programów.

g) projektowanie organizacyjnych rozwiązań systemowych, programowanie i wdrażanie powinno być realizowane przez ten sam zespół osób.

W rozdziale 7 omówimy typową strukturę form projektów i dokumentacji. Obecnie ograniczymy się do podania ogólnych proporcji poszczególnych faz i elementów procesu projektowania. W dalszych rozważaniach pominiemy fazę I. Weźmiemy pod uwagę dwie następne fazy i na ich podstawie rozpatrzemy poszczególne etapy projektowania. W fazie II powstają etapy opracowania koncepcji i wstępnego planowania realizacji SPD, natomiast w fazie III występują etapy projektowania szczegółowych rozwiązań organizacyjnych i programowych systemu oraz etapy wdrażania i eksploatacji SPD. W poszczególnych etapach rozważane są zagadnienia współdziałania kierownictwa, oprogramowania, środków technicznych, szkolenia, lokali, przejścia z dotychczasowego SPD na zmodernizowany, zmian organizacyjnych u użytkownika. Skojarzenie tych zagadnień z poszczególnymi etapami projektowania oraz ich pracochłonnością ilustruje tablica 5.3.

Tablica 5.3

Wzór dokumentu planistyczno-ewidencyjnego, który ułatwia organizowanie procesu projektowania SPD według jednolitego modelu

Etapy Zagadnienia	Faza II		Faza III		Eksploatacja SPD 5
	koncepcja SPD 1	planowanie SPD 2	projektowanie SPD 3	wdrażanie SPD 4	
Współpraca z kierownictwem					
Oprogramowanie					
Środki techniczne					
Szkolenie					
Lokal					
Konwersja SPD					
Zmiany organizacyjne					
	0%	30%	45%	65%	100%

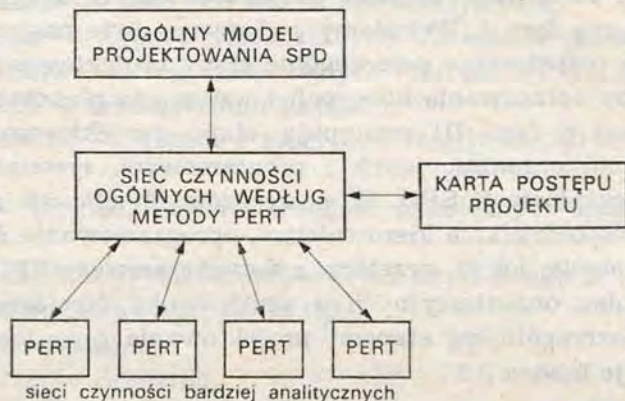
Tablica 5.4

Pracochłonność etapów projektowania SPD
we włoskiej firmie metalurgicznej DALMINE

Etapy	Wskaźniki procentowe ilości osobodni
Studia wstępne	5%
Analiza problemu i projektowanie systemu	25%
Szczegółowe projektowanie	30%
Programowanie	20%
Uruchamianie	20%
Razem:	100%
Okres działania systemu	20 lat

Źródło: Por. L. Kazalski, *Europejski program badawczy DIEBOLD*, „Maszyny matematyczne” 1968, nr 3.

Inny podział na etapy przedstawiamy w tablicy 5.4, według danych włoskiej metalurgicznej firmy Dalmine, projektującej system przetwarzania danych według modelu ZSIK. Na podstawie ogólnego modelu procesu projektowania SPD można ułożyć sieć ogólnych czynności według



Ryc. 5.4. Schemat powiązania dokumentów planistyczno-ewidencyjnych towarzyszących projektowaniu SPD

metody PERT dla tego modelu. Następnie dla poszczególnych etapów i rodzajów zagadnień można ułożyć sieć dla bardziej szczegółowych czynności, również według metody PERT. Według sieci ogólnych czynności PERT można uaktualnić kartę postępu projektu (por. ryc. 5.4).

5.4.2. Pracochłonność projektowania

Złożoność i niepowtarzalność rozwiązań SPD, dodatkowo zróżnicowanych odmiennością sprzętu ETO, a także niejednakowe kwalifikacje projektantów spowodowały powstanie licznych metod obliczania pracochłonności projektowania. Niektóre z tych metod mają charakter raczej przykładowy z tego względu, że nadal największą rolę spełnia w tym zakresie doświadczenie i wyczucie kierownictwa projektu.

Przytoczymy pewne przykłady głównych tendencji, które występują przy ustalaniu normatywów pracochłonności projektowania systemu. Przynajmniej dwie tendencje można wyraźnie rozróżnić. Pierwsza polega na szczegółowej ocenie stopnia trudności poszczególnych programów i dobieraniu odpowiednich wag. Wadą tej metody jest to, że na wstępnym etapie określania pracochłonności projektowania nie ma jeszcze dobrego rozeznania struktury i stopnia skomplikowania systemu programów. Druga metoda określania pracochłonności polega na orientacyjnym szacowaniu zużycia czasu, często na podstawie statystyki z dotychczasowej własnej praktyki lub innych podobnych doświadczeń. Wadą tej metody jest to, że pomniejszając rolę stopnia skomplikowania systemu programów, można czasem popełnić znaczne błędy.

Przedstawimy pewne normatywy pracochłonności projektowania opierając się na:

- szczegółowej ocenie pracochłonności poszczególnych programów,
- orientacyjnej ocenie pracochłonności poszczególnych programów,
- orientacyjnej ocenie pracochłonności projektowania określonych rodzajów SPD, na podstawie doświadczeń.

Ad a). Omówimy tu dwie metody: w jednej posługujemy się tzw. złożonym wskaźnikiem pracy (*composit work index*), druga została opracowana przez firmę IBM. Druga metoda różni się tym od pierwszej, że oprócz oszacowania czasu niezbędnego do ułożenia samego programu — uwzględnia się w obliczeniach pracochłonności niezbędny czas na czynności pomocnicze towarzyszące programowaniu. Przechodząc do omówienia pierwszej metody — złożony wskaźnik pracy (*C*) wyraża się wzorem:

$$C = \frac{a \cdot b \cdot c \cdot d}{e},$$

gdzie:

- liczba rozkazów w języku maszyny (w tys.) wówczas, gdy programy nie są zbyt długie,
- współczynnik złożoności,
- współczynnik zmian,
- współczynnik oryginalności programu,
- współczynnik doświadczenia programistów.

W zestawieniu tym przedstawiamy elementy oceny składające się na:

Współczynnik złożoności — b	Wagi
— dużo wydruków i złożona redakcja,	0,3
— wydruki z koniecznością redagowania poszczególnych znaków,	0,1
— nowe algorytmy,	0,3
— aktualizowanie lub modyfikowanie zapisów,	0,1
— wrywkowy dostęp,	0,2
— tworzenie nowej bazy danych (reorganizacja),	0,9
— wyszukiwanie informacji,	0,3
— wydawanie wyników na monitorze.	0,3
Współczynnik zmian — c	Wagi
— brak zmian,	0,1
— niewiele zmian,	1,3
— kilkanaście zmian,	1,7
— dużo znaczących zmian,	2,0
Współczynnik oryginalności — d	Wagi
— opracowanie po raz pierwszy programu tego typu,	2,0
— istnieje program, który można naśladować,	1,4
— jest kilka programów, które można naśladować.	1,4
Współczynnik doświadczenia programistów — e	Wagi
— wysoko wykwalifikowany programista,	3,0
— dobry programista,	2,5
— średnio wykwalifikowany programista,	2,0
— dostatecznie wykwalifikowany programista,	1,5
— mało wykwalifikowany programista.	1,0

Przykład 1. Określmy pracochłonność ułożenia programu raczej złożonego, który ma aktualizować kartotekę ($b_1 = 0,1$) w sposób wyrywkowy ($b_2 = 0,2$). W toku układania programu wystąpiło niewiele zmian ($c = 1,3$), przy czym można było już korzystać z innego podobnego programu jako wzoru dla układanego programu ($d = 1,4$) przez średnio wykwalifikowanego programistę ($e = 2,0$). Po podstawieniu wybranych współczynników do podanego na wstępie wzoru otrzymamy $C = 1,5$ (półtora osobomiesiaca).

W metodzie opracowanej przez IBM obliczanie pracochłonności projektowania podzielono na 8 etapów dotyczących określenia:

- 1) złożoności programu,
- 2) kwalifikacji wykonawców (*know-how*),
- 3) znajomości danej pracy przez wykonawców,
- 4) pracochłonności ułożenia programu,
- 5) pracochłonności pomocniczych prac,
- 6) wielkości czasu nieużytecznego,
- 7) czasu pozaprojektowego,
- 8) łącznego zapotrzebowania czasu pracy.

Pracochłonność wykonania programu (etap 4) określa się w osobodniach, według podanej zależności:

$$\begin{array}{ccc} \text{etap 1} & & \text{etap 2} \qquad \qquad \text{etap 3} \\ \text{ZŁOŻONOŚĆ} & \times & \left\{ \begin{array}{cc} \text{KWALIFIKACJE} & \text{ZNAJOMOŚĆ} \\ \text{PROGRAMU} & \text{WYKONAWCÓW} + \text{DAJĄCEJ PRACY} \end{array} \right\} \end{array}$$

Ocena złożoności programu polega na określeniu:

— złożoności procedur wejściowo-wyjściowych i przyporządkowaniu im 1 do 5 wag,

— procedur przetwarzania w języku COBOL i przyporządkowania im 4 do 27 wag oraz procedur w autokodzie i przyporządkowaniu im 12 do 37 wag.

Ocena kwalifikacji wykonawców opiera się natomiast na następujących założeniach:

Stanowisko	Waga	Praktyka w zakresie programowania
Starszy programista	od 0,5 do 0,75	ułożył wiele programów dla różnych komputerów.
Programista	od 1 do 1,5	ułożył programy o różnym stopniu skomplikowania dla jednego i tego samego komputera
Młodszy programista	od 2 do 3	układał programy, ale dla innego komputera
Stażysta	od 3,5 do 4,0	nie programował

Ocena znajomości danej pracy przez wykonawców opiera się na następujących danych:

Wymagana znajomość	Ocena	
	bardzo dobra	dostateczna
Szczegółowa wiedza o danej pracy	0,25	0,75
Ogólna dobra wiedza o danej pracy i znajomość pewnych szczegółów	0,50	1,25
Zadowalająca ogólna wiedza o danej pracy, pewna wiedza (lub jej brak) dotyczy szczegółów	0,75	1,50
Nie jest wymagana znajomość danego zagadnienia, natomiast pożądana jest wiedza o zbliżonych zagadnieniach	1,0	1,75
Brak znajomości tej lub podobnej pracy	1,25	2,0

Pracochłonność wykonania prac pomocniczych (etap 5) określa się jako narzut od 70 do 110% w stosunku do etapu 4. Do tego typu prac zalicza się: instruowanie wykonawców, koordynowanie prac itp. Natomiast określenie czasu nieużytecznego (etap 6) otrzymuje się jako narzut 5 do 10% do etapu 4. Zużycie tego czasu wynika z współpracy z instytucjami z zewnątrz (często jednostkami nadrzędnymi), trudności w dostępie do komputerów, zweryfikowania oprogramowania dostarczonego przez producenta, konieczności występowania przerw w pracy (np. śniadaniowych). Określenie czasu pozaprojektowego (etap 7) wynika z potrzeby planowania urlopów, przeszkolenia wojskowego, konferencji, współudziału w innych projektach itp.

Obliczenia od etapu 1 do 4 wykonuje się dla każdego programu oddzielnie. Dalsze zużycie czasu określa się jako narzuty w stosunku do etapu 4.

Przykład 2. Określmy pracochłonność ułożenia programu zdefiniowanego w przykładzie 1.

Etap 1: ze względu na wyrywkowy sposób aktualizowania kartoteki wybieramy dla określenia złożoności procedury: wejściowo-wyjściowej — wagę 4, a aktualizacji redagowanej w autokoderze wagę 24.

Etap 2: program układany jest przez średnio wykwalifikowanego programistę, w związku z tym wybieramy wagę 1,0.

Etap 3: odnośnie znajomości danej pracy, ze względu na wyrywkowy sposób aktualizowania kartoteki, wymagana jest szczegółowa wiedza o danej pracy według bardzo dobrej oceny, dlatego wybieramy wagę — 0,25.

Etap 4: pracochłonność ułożenia programu wyniesie:

$$(4 + 24) \cdot (1,0 + 0,25) = 35 \text{ osobodni,}$$

tzn. mniej więcej tyle co w przykładzie 1 (1,5 osobomiesięca odpowiada średnio 38 osobodniom pracochłonności projektowania).

Etap 5: czas na wykonanie prac pomocniczych określimy przez narzut 70% do czasu ułożenia programu (etap 4), co wyniesie 24,5 osobodni.

Etap 6: nieużyteczny czas pracy określimy przez 5% narzutu do czasu ułożenia programu (etap 4), co wyniesie 1,75 osobodni.

Etap 7: czas pozaprojektowy określimy przez 3% narzut w stosunku do czasu ułożenia programu (etap 4), co wyniesie 1 osobodzień.

Łączne zapotrzebowanie czasu na ułożenie programu, po podsumowaniu poszczególnych czasów określonych na etapie 4, 5, 6, 7 wyniesie 62,25 osobodni, tzn. że do czasu ułożenia programu (netto) należy doliczyć jeszcze około 80% czasu niezbędnego na załatwienie spraw pomocniczych.

Pod tym względem metoda ta ma więcej zalet w stosunku do metody referowanej na wstępie. Jednakże pod warunkiem, że do obliczonego czasu na ułożenie programu według złożoności wskaźnika pracy — dodamy 70 do 80% czasu na sprawy związane z układaniem programu.

Ad b). Omówimy tu normatywy opracowane przez pracowników IBM — M. Bernsteina⁴ i Dicka H. Brandona⁵. Korzystając z tych normatywów nie musimy wdawać się w drobiazgową ocenę złożoności programu, której jeszcze dostatecznie nie znamy w momencie planowania prac projektowo-programowych. Według pierwszego autora, pracochłonność projektowania i programowania zależy od trójstopniowej złożoności programu i różni się dla trzech faz projektowania (tablica 5.5). Konfigurację programów prostych, złożonych i bardzo skomplikowanych przedstawiamy na ryc. 5.5.

Tablica 5.5

*Pracochłonność projektowania i programowania
w zależności od trójstopniowej złożoności programu*

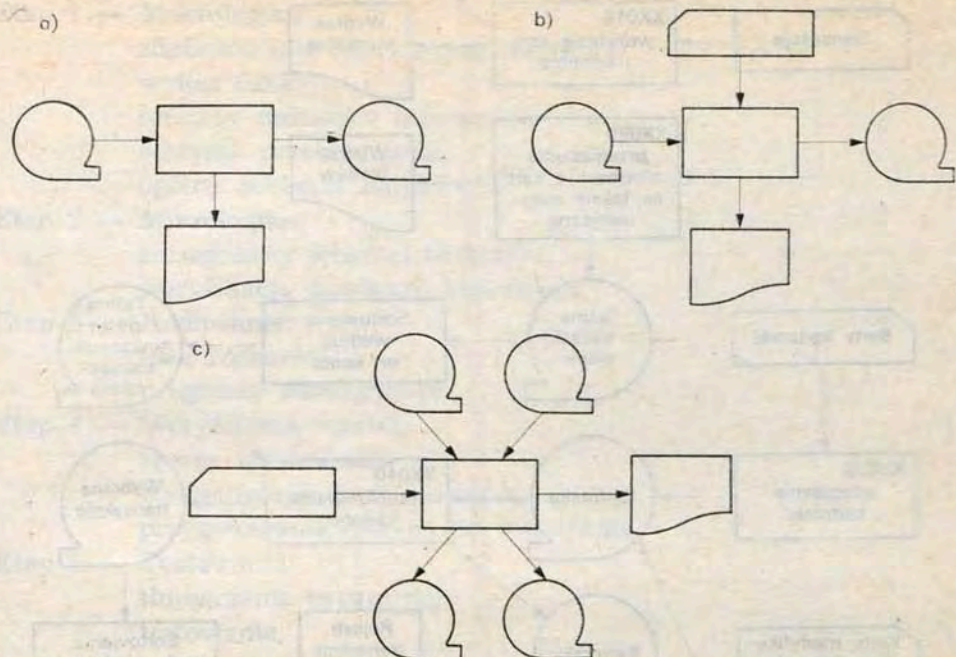
Stopień złożoności programu	Osobodni		
	projektowanie systemu APD	programowanie	uruchamianie
Prosty	3	2	3
Złożony	8	5	8
Bardzo skomplikowany	16	10	20

Z r ó d ł o: Por. M. Bernstein, IBM Corporation, Honolulu 1963.

P r z y k ł a d 3. Określimy pracochłonność ułożenia 7 programów o różnym stopniu złożoności, przedstawionych na ryc. 5.6. Określenie stopnia złożoności poszczególnych programów oraz pracochłonności ich wykonania przedstawiamy w tablicy 5.6. Z danych tej tablicy wynika, że pracochłonność wykonania jednego programu złożonego z kilkuset rozkazów autokodowych (typowego dla podanej tu próbki złożonej z 7 różnych programów) wynosi 21 dni kalendarzowych (150 : 7).

⁴ Por. M. Bernstein, IBM Corporation, Honolulu 1963.

⁵ Por. Dick H. Brandon, *Management Standards for Data Processing*, D. van Nostrand Company, Inc. New York 1963.

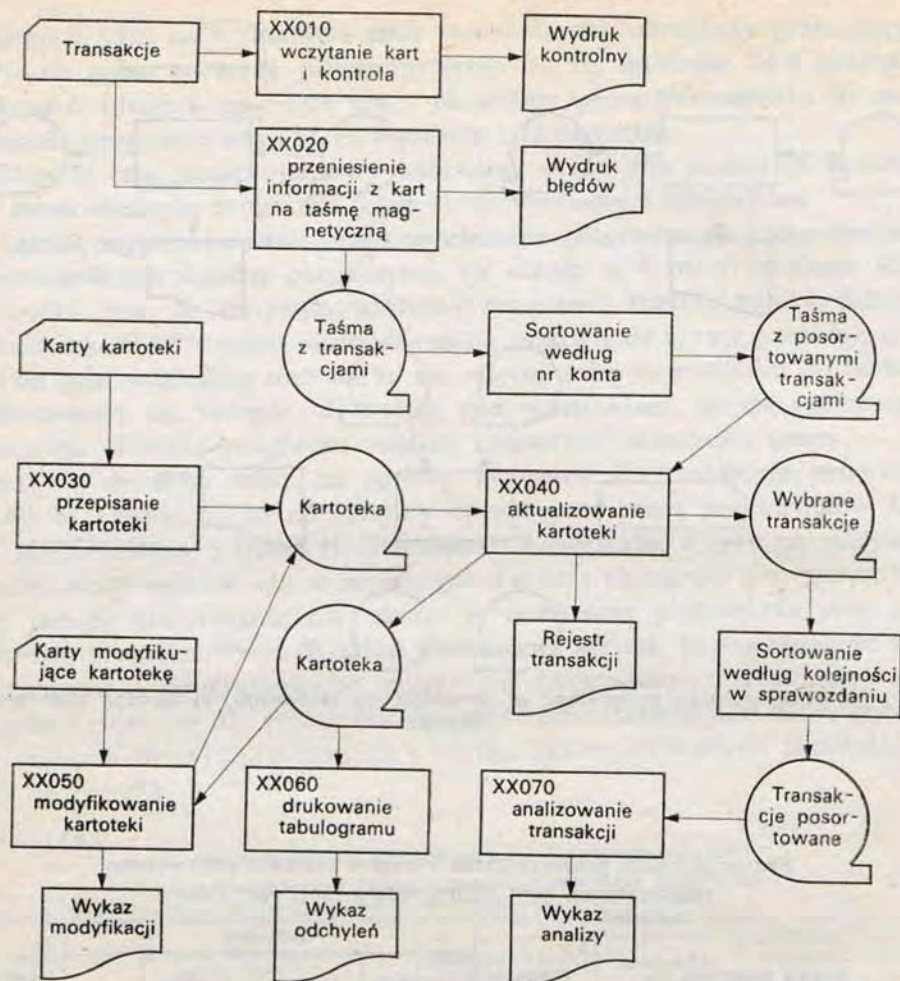


Ryc. 5.5. Konfiguracja programu: a) prostego, b) złożonego, c) bardzo skomplikowanego

Pracochłonność projektowania i programowania programów realizujących cykl obliczeniowy (por. ryc. 4-5)

Tablica 5.6

Nazwa programu	Program	Osobodni				Dni kalendarzowych
		projektowanie	programowanie	uruchamianie	razem	
XX010 wczytywanie kart i kontrola	prosty	3	2	3	8	11
XX020 karta-taśma	prosty	3	2	3	8	11
XX030 przepisanie kartoteki	prosty	3	2	3	8	11
XX040 aktualizowanie kartoteki	bardzo skomplikowany	16	10	20	46	66
XX050 modyfikowanie kartoteki	złożony	8	5	8	21	29
XX060 drukowanie tabulogramu	prosty	3	2	3	8	11
XX070 analizowanie transakcji	prosty	3	2	3	8	11
Łącznie		39	25	43	107	150



Ryc. 5.6. Plan operacyjny przetwarzania danych realizowany 7 programami, których pracochłonność ułożenia obliczona jest w przykładzie 3

Ocena ta chociaż tylko szacunkowa zbliżona jest do oceny szczegółowej metody opracowanej przez firmę IBM, według której ułożenie programu aktualizacji kartoteki wymagało 61,25 osobodni, podczas gdy w tej metodzie dla ułożenia tego programu (por. program XX040) potrzeba jest tyleż (66) dni kalendarzowych. Niemniej ocenę tę należy traktować dość ostrożnie i tylko przykładowo, bowiem przedstawiony dla ilustracji podział programów według stopnia złożoności jest umowny i w konkretnych wypadkach program uznany tu za prosty w rzeczywistości okazać się może bardzo skomplikowany i odwrotnie.

Dick H. Brandon (przytoczymy dane sumaryczne) dzieli proces programowania na 6 etapów, a te z kolei na czynności, dla których w zależności od stopnia złożoności programu (A, B, C, D, E) podaje pracochłonność w dniach.

- Etap 1 — Makrologika
 zdefiniowanie ogniwa przetwarzania,
 wykaz funkcji,
 formaty nośników informacji,
 schemat przetwarzania,
 ogólny schemat blokowy.
- Etap 2 — Mikrologika:
 szczegółowy schemat blokowy,
 weryfikacja powiązań logicznych.
- Etap 3 — Kodowanie:
 lista rozkazów,
 programy standardowe.
- Etap 4 — Weryfikacja ręczna:
 ręczne sprawdzanie,
 wykaz miejsc kontrolowania,
 przygotowanie danych do testowania.
- Etap 5 — Testowanie:
 tłumaczenie programu,
 testowanie,
 wdrażanie.
- Etap 6 — Dokumentacja.

Srednia pracochłonność typowego programu, który spotykamy w codziennej praktyce usługowego ośrodka obliczeniowego wynosi:

<i>Etap</i>	<i>Czas w dniach</i>
1	1
2	6,5
3	7
4	3,5
5	6 (tłumaczenia, 18 testów danych)
6	3,5 (18 stron)

Razem 27,5 dni, czyli 4,5 tygodnia pracy.

Dane te pokrywają się mniej więcej z wynikami obliczeń M. Bernsteina. Ogólnie można stwierdzić, że porównywanie różnych programów jest ryzykowne, bowiem może być obarczone dużym błędem.

Jak wynika z dotychczasowego przeglądu metod obliczania pracochłonności ułożenia programu, pracochłonność ta wynosi dla typowego programu niezintegrowanego SPD modelu PT od 1 do 3 miesięcy w zależności od stopnia złożoności programu oraz wprawy programisty.

Ad c). Przytoczymy tu nieco zmodyfikowaną propozycję pracownika IBM I. Herminjarda opracowaną w Szwajcarii w 1967 r. do projektowania bardziej złożonego systemu przetwarzania wyrywkowo-bieżącego. Autor wyróżnia cztery zadania projektowe, których pracochłonność określa na podstawie doświadczenia:

1) układanie programów wraz z testowaniem do przetwarzania bezpośredniego transakcji przesłanych siecią transmisji danych — odbywało się z prędkością 250 rozkazów na miesiąc,

2) układanie programów do przetwarzania pośredniego transakcji bez korzystania z sieci transmisji danych odbywało się z prędkością od 300 do 600 rozkazów na miesiąc,

3) układanie programu projektującego harmonogram bieżącego przetwarzania przez komputer — odbywało się z prędkością 100 rozkazów na miesiąc,

4) testowanie systemu programów wyniosło od 50 do 70% pracochłonności pierwszego i drugiego zadania. Warto tu dodać, że testowanie bardzo trudnego programu (jakim jest program wymieniony w zadaniu 2) może czasami przekroczyć sam okres układania programu. W ten sposób liczba rozkazów do ułożenia programów wyniesie dla:

- | | |
|---|-------|
| — programów do przetwarzania bezpośredniego | — X1 |
| — programów testowania przetwarzania bezpośredniego (od 20 do 40% X1) | — X2 |
| — programu projektującego harmonogram | — X3 |
| — programów dla przetwarzania pośredniego | — X4. |

Autor proponuje, aby pracochłonność programowania wyrażać wskaźnikiem osobomiesięcy w następujący sposób dla:

— programów do przetwarzania bezpośredniego	$X1 : 250 = Y1$	osobomiesięcy
— programów testowania przetwarzania bezpośredniego	$X2 : 250 = Y2$	„
— programu projektującego harmonogram	$X3 : 100 = Y3$	„
— programów dla przetwarzania pośredniego	$X4 : 500 = Y4$	„
— testowania wymienionych programów	$60\%Y1 = Y5$	„
	Razem Y6	„

Na podstawie doświadczenia różnych ośrodków obliczeniowych wykorzystanie czasu pracy komputera na uruchamianie programów szacuje się na około 2 do 3 godzin pracy średniej wielkości maszyny przez 1 programistę pracującego przez 1 miesiąc. Oznacza to, że 30 programistów zatrudnionych w ośrodku obliczeniowym wymaga miesięcznie od 60 do 90 godzin pracy stosunkowo szybkiej maszyny do testowania swoich programów. Wielkość tego zapotrzebowania wzrasta w miarę nabierania wprawy przez programistów i przeciętnie podwaja się co 2 lata. Przyjmuje się, że na pełne wytestowanie programu zawierającego 1 000 rozkazów trzeba 12 do 18 godzin maszyny, przy stosunkowo małej wprawie programisty. Dla biegłych programistów trzeba zwykle o połowę mniej czasu. Przeciętnie w wyniku jednego miesiąca programowania systemów przetwarzania danych otrzymuje się programy, które dają 1 godzinę ob-

ciążenia maszyny w cyklicznym eksploatacyjnym przetwarzaniu. Z tego względu wspomniany 30-osobowy zespół projektowo-programowy w wyniku rocznej pracy (po okresie szkoleniowym) może obciążyć maszynę w zakresie 2 zmian pracy (same przetwarzanie użyteczne oraz inne prace pomocnicze na maszynie, jak szkolenie, konserwacja itp.). Z praktyki wynika, że przeważnie po 2-letnim okresie szkolenia i projektowania zespół może obciążyć 1 komputer oraz może przystąpić do prac związanych z wykorzystaniem następnej maszyny wspomnianej już wielkości.

W przedstawionych metodach określenia pracochłonności projektowania bierze się za podstawę pracochłonność ułożenia 1 programu. Metody te mają z tego względu ograniczone zastosowanie i mogą dotyczyć przede wszystkim bardzo doświadczonych grup projektowych lub ośrodków obliczeniowych i to raczej w zakresie projektowania cząstkowych SPD. W wypadku projektowania bardziej złożonych SPD bardziej przydatna może okazać się metoda posługiwania się danymi zawartymi w tablicy 5.2 i tablicy 5.4. Ogólnie można stwierdzić, że tylko w niektórych wypadkach udaje się dobrym projektantom — zaprojektować i zaprogramować w terminie krótszym niż 3 miesiące 1 ogniwo przetwarzania złożone z kilku programów. Przeciętnie 1 ogniwo przetwarzania dla SPD projektowanego według modelu PT wymaga 3 do 6 osobomiesięcy pracochłonności.

Obliczenia te nie uwzględniają jednego faktu, mianowicie niezależnie od wymaganej pracochłonności projektu użytkownik lub grupa projektowa dysponuje stałą obsadą pracowników, którą trudno zmieniać w zależności od okresowych potrzeb. Według firmy DIEBOLD⁶, dla złożonego

Tablica 5.7

Struktura i liczba zatrudnionych pracowników przy projektowaniu SPD według modelu ZSIK, według faz projektowania

Specjalność	Faza		
	projektowania	wdrażania	eksploatacji
Analitycy systemu	3	1	
Projektanci systemu tradycyjnego	2	1	
Projektanci APD	2	2	0,25
Programiści systemów programowania	2	1	
Programiści zastosowań	1	3	0,5
Programiści zmian			0,5
Operatorzy		2	2
Razem	10	10	3,25

⁶ Por. Diebold European Research Program, IMIS — The Common Data Base, April 1967, Dokument nr E29.

SPD projektowanego według modelu ZSIK (IMIS) potrzebna jest stała obsada, licząca w okresie projektowania i wdrażania 10 osób, natomiast, w okresie eksploatacji liczba osób może być zmniejszona od 3 do 4 osób (por. tablica 5.7). Należy dodać, że bierze się pod uwagę tylko osoby wysoko wykwalifikowane. W tym zakresie istnieją jednak duże rozbieżności poglądów i praktycznych rozwiązań. Na przykład według normatywów wspomnianej już metodyki — SOP, do złożonego SPD liczba zatrudnionych spada z 24 w początkowej fazie projektowania do 13 podczas eksploatacji systemu. Jeśliby podzielić czas projektowania i eksploataowania na 5 okresów, to średnie zatrudnienie w tych okresach przedstawia się jak w tablicy 5.8.

Tablica 5.3

Średnie zatrudnienie pracowników w fazach: projektowania, wdrażania i eksploatacji SPD

Okresy		Projektowanie i wdrażanie				Eksploatacja
		1	2	3	4	5
Koordynator	D	+1	0	-0	0	0
		1	1	1	1	1
Projektanci	D	+4	-2	-1	-1	0
	P	4	2	1	0	
Programiści	$\frac{1}{2}$ D	+10	-4	-3	-2	0
	$\frac{1}{2}$ P	10	6	3	1	1
Koderzy	P	+2	0	0	-2	
		2	2	2	0	0
Operatorzy maszyn	$\frac{1}{2}$ D	+4	+2	+2	+1	0
	$\frac{1}{2}$ P	4	6	8	9	9
Operatorzy systemów APD	D	+4	0	0	0	0
		2	2	2	2	2
Razem		23	19	17	13	13

D — doświadczeni, P — praktykanci, górna liczba oznacza przyrosty lub zwolnienia, dolna liczba wskazuje stałe zatrudnienie w danym okresie.

5.4.3. Kierowanie projektowaniem

Kierowanie pracą zespołu projektującego SPD wymaga odpowiedniego planowania i ewidencjonowania prac i — co może jest najważniejsze — powinno prowadzić do zgrania zespołu i twórczego inspirowania jego pracami. Nie pomogą projektowi najbardziej wymyślne formy kontroli ani metody planistyczno-ewidencyjne, jeżeli realizatorzy projektu, a

w szczególności kierownictwo, nie będą mieli odpowiednich kwalifikacji. Planowaniu i ewidencji przypisuje się często przesadnie dużą rolę, natomiast zapomina się o konieczności uzgadniania z użytkownikami pewnych rozwiązań koncepcyjnych systemu. Niedopracowanie koncepcji może spowodować większe — choć początkowo niedostrzegalne — kłopoty niż brak pozornie niezbędnych precyzyjnych form sprawozdawczości oraz kontroli.

Zadaniem kierownictwa projektu jest także zapewnienie niezbędnych warunków prowadzenia prac projektowych, a więc: dobrych warunków lokalowych, odpowiedniego wynagrodzenia zgodnego z kwalifikacjami i rzeczywistym wkładem pracy, sprawnego wykonywania prac pomocniczych typu maszynopisania, kreślenia, powielania, dostępu do komputera, na której prowadzi się testowanie programów. Kierownictwo powinno dbać o odpowiednie szkolenie specjalistyczne wykonawców projektu, ułatwiać im dostęp do bieżącej literatury fachowej, udział w ciekawych zjazdach i kongresach, korzystanie z wartościowych staży w przodujących ośrodkach krajowych i zagranicznych, publikowanie prac itp.

Jest bardzo istotne, aby kierownictwo projektu interesowało się przebiegiem pracy każdego wykonawcy. Projektant powinien być przeświadczony o tym, że jego odcinek pracy w zespole jest ważny i jego osobiste osiągnięcia przyczyniają się do sukcesu projektu. Każde ofiarne wykonane zadanie powinno być od razu przez kierownictwo zauważone i odpowiednio ocenione. W niektórych wypadkach projektant powinien partycypować w otrzymaniu premii proporcjonalnej do efektów ekonomicznych, które zostały osiągnięte dzięki projektowi; jest to jednak uzależnione od obowiązujących przepisów.

Kierownictwo projektu powinno mieć kredyt zaufania u przełożonych i podwładnych. W wypadku nadużycia zaufania, lepiej zmienić kierownictwo niż podrywając jego niezbędny w projektowaniu autorytet doprowadzić do popsucia atmosfery w zespole, a w konsekwencji do jego rozpadu.

Do mierników oceny postępu prac projektowo-wdrożeniowych można zaliczyć: czas i nakłady finansowe wykorzystane przy wykonaniu poszczególnych rodzajów zadań w stosunku do zaplanowanych. Pewne trudności mogą wystąpić przy określeniu zakresów zadań. Można je ustalać liczbowo jako: ogniwa przetwarzania, programy lub jako projekty (i programy) tabulogramów, kartotek.

Dokumentacja planistyczno-ewidencyjna powinna być maksymalnie uproszczona. Na przykład można ją sprowadzić do:

a) uniwersalnego druku wykresu Gantta, na który nanosi się planowane i faktyczne terminy rozpoczęcia i zakończenia zadań; harmonogram może być ułożony w układzie „wykonawca — poszczególne zadania” lub odwrotnie; pożądane jest oddzielne układanie harmonogramu dla ogniwa przetwarzania i dla programów,

b) miesięcznych kart pracy projektantów wypełnianych raz na miesiąc⁷, w których ewidencjonuje się wykorzystanie czasu na poszczególne zadania projektowe i pozaprojektowe. Na podstawie danych z tej karty można uaktualniać na bieżąco wykorzystywanie środków finansowych. W uzasadnionych wypadkach karta pracy jest podstawą obciążenia finansowego (fakturowania) zleceniodawcy.

c) karty postępu projektu, który wynika z ogólnego modelu projektowania (por. ryc. 5.4). Na karcie tej wpisuje się terminy planowane i faktycznie dotrzymywane.

5.5. Organizacja komórek projektowych

5.5.1. Struktura zatrudnienia

Struktura personelu zatrudnionego w grupach projektowych i ośrodkach obliczeniowych zależy od wielu czynników, np. od rodzaju użytkownika, stopnia samodzielności ośrodka. Wyodrębnienia poszczególnych grup zawodów można dokonać według następujących kryteriów:

- 1) podobieństwa wymagań kwalifikacyjnych (wykształcenia),
- 2) odpowiedzialności wynikającej z obowiązków,
- 3) rodzaju wykorzystywanego sprzętu itp.

Uwzględniając w analizie zatrudnienia te kryteria, można przyjąć następujące grupy pracowników:

- a) projektanci systemów,
- b) programiści,
- c) operatorzy systemów,
- d) operatorzy maszyn,
- e) kierownicy,
- f) urzędnicy.

Do grupy projektantów systemów można zaliczyć pracowników zaangażowanych w studia i badania, projektowaniu formularzy, projektowaniu metod organizacyjnych, projektowaniu systemów APD, projektowaniu zastosowań metod matematycznych, projektowaniu systemów obliczeń numerycznych. Do grupy programistów zaliczyć można programistów: maszyn analitycznych, systemów APD, obliczeń numerycznych, uniwersalnego oprogramowania, operatorów systemów APD.

Strukturę zatrudnienia oraz schemat awansowania pracowników ośrodka obliczeniowego przedstawiamy na ryc. 5.7.

W nowo organizowanych ośrodkach obliczeniowych dość ważnym problemem jest kolejność przyjmowania pracowników. Na samym początku

⁷ W niektórych ośrodkach projektowych można zaobserwować tendencję codziennego wypełniania tej karty. W rzeczywistości jest to fikcja, bowiem faktycznie projektant wypełnia ją jednorazowo po zakończeniu miesiąca.