Purdue University
Purdue e-Pubs

Department of Computer Science Technical Reports

Department of Computer Science

1980

Effort Minimization Based on Hierarchical Modularization

S. D. Conte

Report Number:
80-347

Conte, S. D., "Effort Minimization Based on Hierarchical Modularization" (1980). *Department of Computer Science Technical Reports.* Paper 277.
https://docs.lib.purdue.edu/cstech/277

This document has been made available through Purdue e-Pubs, a service of the Purdue University Libraries. Please contact epubs@purdue.edu for additional information.

EFFORT MINIMIZATION BASED ON HIERARCHICAL MODULARIZATION

S. D. CONTE

Computer Sciences Department
Purdue University
West Lafayette, IN 47907

CSD-TR 347

Effort Minimization Based on Hierarchical Modularization

Preliminary Report S.D. Conte

1. Hierarchical Modularization of Large Software Systems

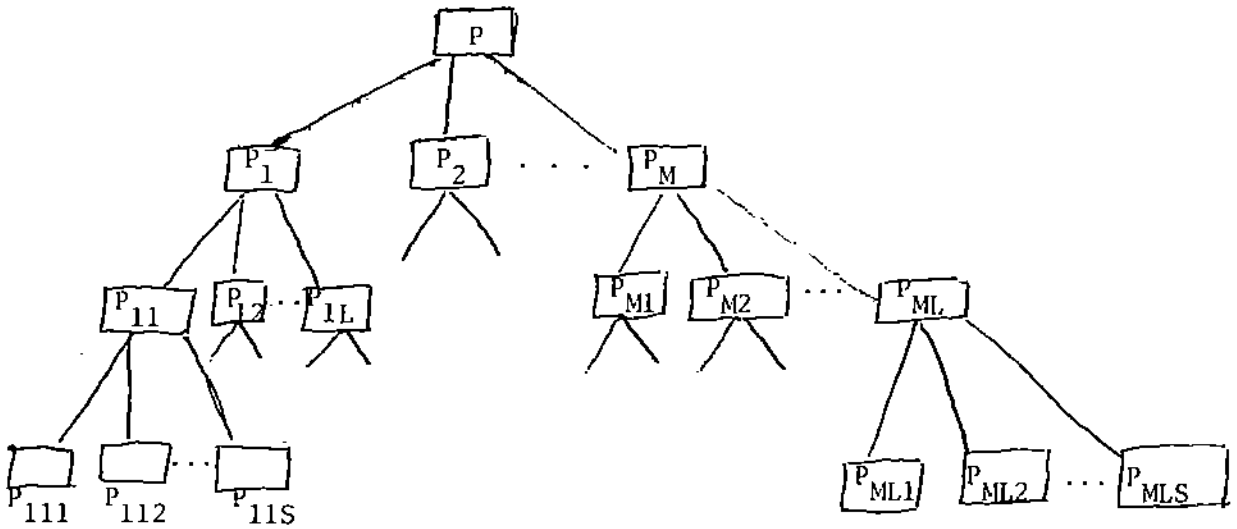
The principle of modularizing large programming projects is now widely accepted by software managers. Experience indicates that proper modularization leads to fewer errors and better understandibility. However, there have been few quantitative estimates of the benefits of modularization. The theory of software science can be used to produce metrics which lead to both effort minimization and to error reduction.

For very large programs Lattanzi [1] and others have suggested that a four level hierarchical modularization seems to be reasonable. Lattanzi has suggested the following levels for programs of the size indicated.

Level	Name	Size (Source Lines)	Software Science Length	#Modules
0	Program	110K - 225K	300 - 600K	1
1	Subprogram	3000 - 5000	8000 - 13000	M
2	Module	300 - 500	800 - 1300	ML
3	Segment	30 - 50	80 - 150	MLS

As the Column #Modules indicates, each of the M subprograms will be divided into L modules and each of the L modules into S segments.

Turner [2] has also emphasized the importance of structured modularization. He points out that if modules are limited to between 50 and 200 lines of executable code, as is commonly advocated, then a large program will contain very many modules and the way in which these modules are structured will have a significant inpact on the reliability of the resulting code as well as on the effort required to write this code. Turner advocates what he calls a pure tree structure hierarchical modularization in which a given lower level module may be called only by a single module on the next higher level. Cross-sharing of modules between trees is not allowed. In this paper we shall essentially be concerned only with pure tree structured modularization pictured graphically below:



Pure Tree Structured Hierarchical Modularization

II½. Basic Assumptions and the Software Science Effort

Software science [3] introduces the basic matrics η_1, the number of unique operators and η_2, the number of unique operands. The length N of a program P is defined to be

$$N = N_1 + N_2$$

where N_1 is the total occurances of operators in P and N_2 is the total occurances of operands in P. In our model we shall assume that we are given or can estimate the length N of the program P. If the estimate of the size of a program is given in lines of code (LOC), then it is easy to convert the size into the software science length by the formula

$$N = k \cdot LOC$$

where the conversion factor k depends on the language being used. For Fortran, for example, $k \sim 6.6$.

In the hierarchically modularized tree of Figure 1, we will have in all MLS segments at level 3. We assume, for simplicity, that all segments are of the same length $\overline{N}$. Hence we must have

$$MLS\overline{N} = N \tag{1}$$

Software science defines the effort required to write a program of length N as

$$\text{Effort} = \frac{\text{Volume}}{\text{Level}}$$

where volume and level are defined in [3]. If we assume that $\eta_1 = \eta_2 = \frac{1}{2}\eta$ and $N_2 = N/2$, then the effort equation can be simplified to

$$\text{Effort} = \tfrac{1}{4} N^2 \log \eta$$

where η may be computed from the equation

$$N = \eta \log \eta/2$$

If we have a segment of length $\overline{N}$, the effort required to write a program of that length under the assumptions made above is

(2a) $$\overline{e} = \tfrac{1}{4} \overline{N}^2 \log \overline{\eta}$$

with

(2b) $$\overline{N} = \overline{\eta} \log \overline{\eta}/2$$

Since there are in all MLS segments the effort required to write all these segments will then be

(3) $$\overline{E} = MLS\overline{e}$$

III. The Interfacing Model

The effort required to write a program consisting of MLS segments must also include the effort required to assure proper interfacing of these segments and modules. As we increase the number of modules or segments we increase the overhead required for proper interfacing. This interfacing arises because various I/O parameters will be common to several, or perhaps, all modules at the same level.

Software science defines the potential volume of a program as

$$V^* = (2 + \eta_2^*) \log (2 + \eta_2^*)$$

where η_2^* is the number of conceptually unique operands required for that problem. If we are given N and the programming language, then the formulas of Software Science permit us to determine η_2^* uniquely.

Our model for the effort required to interface properly is as follows. Let η_2^* be the number of conceptually unique operands in program P. At

level 1, we assume that all of these unique operands will appear in each of the M subprograms. Let $v_1 = \eta_2^*$, represent these operands. We now assume that the unit effort required to assure proper interfacing is proportional to the effort required to write a program with a vocabulary of length v_1. From software science this effort is

(4) $$I_1 = \frac{1}{4} N_1^2 \log v_1, \quad N_1 = v_1 \log v_{1/2}.$$

We must now be sure that each variable appears correctly in each of the M subprograms. Since there are M(M-1)/2 paths correcting these M subprograms, the subtotal interface effort at level 1 is given by

(5) $$E_1 = 1/2\ M(M-1) I_1$$

At level 2 each of the M subprograms will contain L modules. We assume that these L modules must interface with each other (but not with modules in any other subprograms).

Let $v_2 = k_2 \eta_2^*$ be the number of I/O variables required for interfacing. Generally we would expect $0 < k_2 < 1$. Then as argued above the unit interface effort at level 2 is

(5) $$I_2 = \frac{1}{4} N_2^2 \log v_2, \quad N_2 = v_2 \log v_{2/2}$$

and the subtotal effort required at level 2 for interfacing the modules in the subprogram will be

(6) $$E_2 = 1/2\ L(L-1) I_2$$

At level 3, we define

$$v_3 = k_3 \eta_2^*$$

and as above arrives at

(7) $$I_3 = \frac{1}{4} N_3^2 \log v_3, \quad N_3 = v_3 \log v_{3/2}$$

$$E_3 = 1/2\ S(S-1) I_3$$

for the subtotal effort for the S segments within one module.

The factors k_i in the formulas

(8) $$v_i = k_i \eta_2^* \quad (i = 0,1,2,3)$$

should satisfy

$$0 < k_i \leq 1$$

and should decrease with i. One possible choice might be

(9) $$k_i = 1/2^{i-1} \quad (i = 1,2,3)$$

but in general, the choice must be based on experimental evidence. Note that if the k_i are chosen according to (9), then $I_3 < I_2 < I_1$.

Notice that the unit interface effort at any level is independent of the number of subdivisions in that level.

The overhead model is summarized in Table 1.

Table 1

Overhead Effort

Level	Unit Overhead	Subtotal Effort	Total Effort
0	0	0	0
1	I_1	$1/2\ M(M-1)I_1$	$E_1 = 1/2\ M(M-1)I_1$
2	I_2	$1/2\ L(L-1)I_2$	$E_2 = 1/2\ ML(1-1)I_2$
3	I_3	$1/2\ S(S-1)I_3$	$E_3 = 1/2\ MLS(S-1)I_3$

The total effort E required to write all MLS segments and to properly interface them will then be

(10) $$E = MLS\bar{e} + E_1 + E_2 + E_3 .$$

IV. The Optimization Problem

We note that E is a function of M,L,S and $\bar{N}$. However since

$$MLS\bar{N} = N$$

one of these is not independent. Hence E is a function of 3 independent variables. The optimum choice of these independent variables (assumed to be M,L,S) must satisfy the equations

$$\frac{\partial E}{\partial M} = 0$$

(11)
$$\frac{\partial E}{\partial L} = 0$$

$$\frac{\partial E}{\partial S} = 0$$

Qualitatively we can see what happens by considering restricted choices of the variables M,L,S,$\bar{N}$. For example if we fix the length of each segment $\bar{N}$ and also take M fixed, then we will have a tradeoff between L and S, since $LS = N/\bar{N}M$. The first 2 terms of (N), namely $MLS\bar{e}$ and E_1 will be fixed under these conditions but as L increases, S must decrease. An optimum value must therefore exist for each fixed M and $\bar{N}$.

The optimization equations (11) can be expanded to obtain

$$\frac{\partial E}{\partial M} = LS\bar{e} + MLS\frac{\partial \bar{e}}{\partial M} + \frac{\partial E_1}{\partial M} + \frac{\partial E_2}{\partial M} + \frac{\partial E_3}{\partial M} = 0$$

(12)
$$\frac{\partial E}{\partial L} = MS\bar{e} + MLS\frac{\partial \bar{e}}{\partial L} + \frac{\partial E_1}{\partial L} + \frac{\partial E_2}{\partial L} + \frac{\partial E_3}{\partial L} = 0$$

$$\frac{\partial E}{\partial S} = ML\bar{e} + MLS\frac{\partial}{\partial S} + \frac{\partial E_1}{\partial S} + \frac{\partial E_2}{\partial S} + \frac{\partial E_3}{\partial S} = 0$$

Now

$$\frac{\partial \bar{e}}{\partial M} = -\frac{1}{2}\frac{\bar{N}^2}{M}\log\bar{\eta} - \frac{1}{4}\frac{\bar{N}^3}{M}\frac{\log e}{\bar{\eta}\log e\,\bar{\eta}/2}$$

$$= \frac{-2}{M}\ \bar{e} - \frac{1}{4}\ \frac{\bar{N}^3}{M}\ \frac{\log e}{\bar{\eta}\ \log e\ \bar{\eta}/2}$$

$$\frac{\partial \bar{e}}{\partial L} = \frac{-2}{L}\ \bar{e} - \frac{1}{4}\ \frac{\bar{N}^3}{L}\ \frac{\log e}{\bar{\eta}\ \log e\ \bar{\eta}/2}$$

$$\frac{\partial \bar{e}}{\partial S} = \frac{-2}{S}\ \bar{e} - \frac{1}{4}\ \frac{\bar{N}^3}{S}\ \frac{\log e}{\bar{\eta}\ \log e\ \bar{\eta}/2}$$

and hence (12) becomes:

$$\frac{\partial E}{\partial M} = LS\bar{e} - 2LS\bar{e} - \frac{MLS}{4}\ \frac{\bar{N}^3}{M}\ \frac{\log e}{\bar{\eta}\ \log e\bar{\eta}/2} + 1/2\ (2M-1)I_1 + 1/2\ L(L-1)I_2 + 1/2\ LS(S-1)I_3 = 0$$

$$\frac{\partial E}{\partial L} = MS\bar{e} - 2MS\bar{e} - \frac{MLS\ \bar{N}^3}{4L}\ \frac{\log e}{\bar{\eta}\ \log e\bar{\eta}/} + 1/2\ M(2L-1)I_2 + 1/2\ MS(S-1)I_3 = 0$$

$$\frac{\partial E}{\partial S} = ML\bar{e} - 2ML\bar{e} - \frac{MLS\ \bar{N}^3}{4S}\ \frac{\log e}{\bar{\eta}\ \log e\bar{\eta}/2} + 1/2\ ML(2S-1)I_3 = 0$$

Simplifying

$$\frac{\partial E}{\partial M} = -LS\bar{e} - \frac{LS\bar{N}^3}{4}\gamma + 1/2\ (2M-1)I_1 + 1/2\ L(L-1)I_2 + 1/2\ LS(S-1)I_3 = 0$$

$$\frac{\partial E}{\partial L} = -MS\bar{e} - \frac{MS\bar{N}^3}{4}\gamma + 1/2\ M(2L-1)I_2 + 1/2\ MS)S-1)I_3 = 0$$

(13)

$$\frac{\partial E}{\partial S} = -ML\bar{e} - \frac{ML\bar{N}^3}{4}\gamma + 1/2\ ML(2S-1)I_3 = 0$$

where

(14)

$$\gamma = \frac{\log e}{\bar{\eta}\ \log e\bar{\eta}/2}$$

Finally

$$\text{a)}\quad \frac{\partial E}{\partial M} = -\bar{e} - \frac{\bar{N}^3}{4}\gamma + \frac{(2M-1)I_1}{2LS} + \frac{(L-1)}{2S}I_2 + \frac{(S-1)I_3}{2} = 0$$

(15)

$$\text{b)}\quad \frac{\partial E}{\partial L} = -\bar{e} - \frac{\bar{N}^3}{4}\gamma + \frac{(2L-1)I_2}{2S} + \frac{(S-1)I_3}{2} = 0$$

$$\text{c)}\quad \frac{\partial E}{\partial S} = -\bar{e} - \frac{\bar{N}^3}{4}\gamma + \frac{(2S-1)I_3}{2} = 0$$

V. Solving the Optimization Equations

The equations (15) can be solved by a secant-type iteration. The procedure is as follows:

Step 1. We are given N and η_2^*.

Step 2. Pick 2 starting values for $\bar{N}$, say $\bar{N}_{i-1}$, $\bar{N}_i$ (e.g. $\bar{N}_{i-1} = 100$ $\bar{N}_i = 110$)

Step 3. Compute I_1, I_2, I_3 (these depend on N and η_2^* but not $\bar{N}$)

Step 4. Compute
$\bar{e}(\bar{N}_i)$, $\gamma(\bar{N}_i)$, $\bar{e}(\bar{N}_{i-1})$, $\gamma(\bar{N}_{i-1})$

Step 5. Solve the System (15) in the order
(15c) for S_i, (15b) for L_i, (15a) for M_i and also for i-1.

Step 6. We now can evaluate
$f(M_i, L_i, S_i, \bar{N}_i) = M_i L_i S_i \bar{N}_i - N$
$f(M_{i-1}, L_{i-1}, S_{i-1}, \bar{N}_{i-1}) = M_{i-1} L_{i-1} S_{i-1} \bar{N}_{i-1} - N$

Step 7. Update $\bar{N}$ using
$$\bar{N}_{i+1} = \bar{N}_i - f(\bar{N}_i)\frac{\bar{N}_i - \bar{N}_{i-1}}{f(\bar{N}_i) - f(\bar{N}_{i-1})}$$

Step 8. Evaluate $f(\bar{N}_{i+1})$

Step 9. If $|\bar{N}_{i+1} - \bar{N}_i| < \varepsilon$, Stop

Else

If $|f(N_{i+1})| < \delta$, Stop

Else return to step 4

(Choices of $\varepsilon = .1$, $\delta = 1$)

VI. Other Overhead Models

From SS an alternative formula for the effort is

$$E = \frac{V^{*3}}{\lambda^2} \quad . \tag{16}$$

If we have M subprograms, at level 1, then we must make log M mental comparisons of the interface variables for one subprogram and hence M log M comparisons in all. The effort, however, to interface all variables will be greater than M log M. Inspired by (16) we can hypothesize that the total effort for interfacing is given by

$$E_1 = (M \log M)^3$$

At level 2 we similarly hypothesize that the effort for interfacing in this one subprogram is $(L \log L)^3$ and since there are M subprograms the subtotal effort at level 2 is

$$E_2 = M(L \log L)^3$$

At level 3 by a similar argument we arrive at $(S \log S)^3$ for the effort within one module and for all of the ML modules we get

$$E_3 = ML\ (S \log S)^3$$

The total effort then is

$$E = MLS\bar{e} + E_1 + E_2 + E_3 \tag{17}$$

The value of M, L, S which give a minimum for (17) most now satisfy

$$\frac{\partial E}{\partial M} = \frac{\partial E}{\partial L} = \frac{\partial E}{\partial S} = 0$$

Written out and using Equations (13) we get

$$\frac{\partial E}{\partial M} = -LS\bar{e} - \frac{LS\bar{N}^3\gamma}{4} + 3(M \log M)^2 (\log M + \log e) + (L \log L)^3 + L(S \log S)^3 = 0$$

$$\frac{\partial E}{\partial L} = -MS\bar{e} - \frac{MS\bar{N}^3\gamma}{4} + 3M(L \log L)^2 \log eL + M(S \log S)^3 = 0$$

$$\frac{\partial E}{\partial S} = -ML\bar{e} - \frac{ML\bar{N}^3\gamma}{4} + 3ML(S \log S)^2 \log eS = 0$$

(18)

a) $$\frac{\partial E}{\partial M} = -\bar{e} - \frac{\bar{N}^3\gamma}{4} + \frac{3}{LS}(M \log M)^2 \log eM + \frac{(L \log L)^3}{LS} + \frac{(S \log S)^3}{S} = 0$$

b) $$\frac{\partial E}{\partial L} = -\bar{e} - \frac{\bar{N}^3\gamma}{4} + \frac{3}{S}(L \log L)^2 \log eL + \frac{(S \log S)^3}{S} = 0$$

c) $$\frac{\partial E}{\partial S} = -\bar{e} - \frac{\bar{N}^3\gamma}{4} + 3(S \log S)^2 \log eS = 0$$

Equations (18) combined with

$$f(\bar{N}) = MLS\bar{N} - N = 0$$

can be solved as before to obtain the minimum values of M, L, S and $\bar{N}$.

VII. Some Numerical Results

Table I contains some numerical results for the one, two and three level models with different level factors based on the overhead model of Secton III. The values of M,L,S and $\overline{N}$ for which a minimum effort is attained are almost never integers since we assume in the model that they are continuous variables. However, in the table we have rounded off M,L,S to the nearest integer and then obtained $\overline{N}$ using $MLS\overline{N} = N$ and the corresponding effort using (10).

The results in the table are interpreted as follows. If we look at the ~~first~~ case corresponding to a program of size N = 10,000 and if we desire one level of modularization then the minimum effort will be achieved by dividing the program into 13 modules of average size $\overline{N} = 769$. With 2 levels of modularization minimum effort will be achieved if we divide the program into 5 subprograms at level 1 and 8 modules at level 2. This will give a total of 40 modules each of length $\overline{N} = 250$. With 3 levels of modularization the minimum will be achieved for M = 2, L = 5, S = 9 or a total of 90 segments each of length $\overline{N} = 111$. Note that the minimum effort decreases as the level increases. The results in the row labeled 3' were obtained by changing the level factor k_3 from 1/4 to 1/3. The effect of increasing k_3 is to increase the unit effort at level 3 and thus to decrease the number S of segments at level 3. In general, increasing k_3 will also increase the value of $\overline{N}$ and of E at the minimum.

The results seem to indicate that 3 level modularization is always best since E is always least at that level. There may be good management reasons to reject 3 level modularization, at least for smaller programs, since the number of segments at level 3 may be too large. In the case N = 10,000 we would need 90 segments of length $\overline{N} = 111$. Good management practice

Table I

Summary of Results for one, two, three level Models

Level Factors: $k_1=1$, $k_2=\frac{1}{2}$, $k_3=\frac{1}{4}$ or for Level 3' ($k_1=1$, $k_2=\frac{1}{2}$, $k_3=1/3$)

N	Levels	$\bar{N}$	M	L	S	$E*10^{-6}$	N	$\bar{N}$	M	L	S	$E*10^{-6}$
10K	1	769	13			20.5	2K	222	9			.963
	2	250	5	8		6.1		63	4	8		.299
	3	111	2	5	9	2.1		18	2	5	11	.095
	3'	93	3	6	6	2.7		33	2	5	6	.123
20K	1	1250	16			74.1	4K	364	11			3.64
	2	370	6	9		21.4		125	4	8		1.10
	3	123	3	6	9	7.4		40	2	5	10	.353
	3'	185	3	6	6	9.5		67	2	5	6	.484
30K	1	1667	18			156.5	6K	500	12			7.85
	2	476	7	9		44.3		150	5	8		2.37
	3	185	3	6	9	15.3		60	2	5	10	.769
	3'	278	3	6	6	20.0		100	2	5	6	1.08
40K	1	2105	19			265.4	8K	615	13			13.5
	2	571	7	10		74.1		200	5	8		4.03
	3	247	3	6	9	25.6		80	2	5	10	1.34
	3'	278	4	6	6	32.8		111	2	6	6	1.80
50K	1	2500	20			399.2						
	2	714	7	10		110.0						
	3	309	3	6	9	38.7						
	3'	298	4	7	6	48.8						
100K	1	4000	25			1409.4						
	2	1111	9	10		373.3						
	3	397	4	7	9	130.2						
	3'	595	4	7	6	166.7						
200K	1	6667	30			4936.4						
	2	1818	10	11		1255.2						
	3	571	5	7	10	440.4						
	3'	714	5	8	7	548.4						

suggests that an average module size should be 50-80 higher language level source statements which corresponds roughly to $300 < \bar{N} < 500$. Thus, it would be reasonable to use 3 level modularization for programs of length $N > 30,000$, to use 2 level modularization for programs of length N between 6000 and 30,000, and to use one level modularization for programs of length $N < 8000$.

In this model, the level factors k_1, k_2, k_3 play a critical role since they determine the overhead effort at each level. We should be able to determine reasonable values for these factors by comparing numerical results achieved by the model against realistic data on implemented projects. To some extent we might expect these factors to vary with the type of program. A command and control program, for example, might be expected to have larger interfacing overhead, and hence larger level factors, than a straight forward application program

VIII. Experimental Validation of Model

We first examined a library of 32 Fortran programs (See Table II) ranging in size from 3345 lines of code to 55 lines of code, or in terms of the software science length from N=17609 to N=353. We counted the number of subroutines M within each program and equated these with the number of modules assuming one level modularization. The mean program size was N=5764, the mean module size was $\overline{N}$=446 and the mean number of modules was 13.

Applying the one level modularization model we found the optimum value of the parameters M and $\overline{N}$ for a program of size N=5764 to be M=12, $\overline{N}$=480. It would appear that the natural division of the average size program into subroutines is not too far from optimal. In any case the model seems to conform closely to this experimental evidence.

Additional evidence is of course needed to confirm the validity of the model, and in particular the choices of the parameters k_1, k_2 and k_3.

A second test for this model is provided by F. Akiyama's data on a large software project published in 1971. This large project, which required a reported 100 man-months to complete, was broken down into 9 large modules. The number of lines of assembly code for each module was given as well as other data. Software Science metrics, were, of course, not reported but by making some rough approximations they could be deduced. Halstead in [2] obtains the following software science metrics for these modules:

Module	N	η	E(millions)
1	8064	913	170
2	2658	356	15
3	10906	1184	323
4	3348	432	28
5	4102	504	100
6	5026	609	66
7	1398	207	6
8	7584	855	59
9	6824	790	136
SUM	49910		903
Mean	5546		

Our restricted one-level model can be applied to this program with N=49,910 and $\overline{N}$=5546 with M=9 and η_2^*=112. We can compute I_1 = 719,805 and $\overline{e}$ = 72.072 millions, and from these

$$E = M\overline{e} + \tfrac{1}{2} M(M-1)\ I_1 = 674.56 \text{ million emd's}$$

Assuming a 40 hour week with 4 1/6 weeks per month and 18 discriminations per second, there would be 10.8 million discriminations per man-month. Thus

$$\text{Man-months} = E/10.8 \approx 63$$

Considering the rough nature of the approximations as well as the data, this result when compared with the reported charged 100 man-months of effort for this project is actually quite good. This is especially so when we recall that the software science E assumes concentrated programming effort whereas the reported effort almost certainly does not.

From Table I we see that for a program of size N=50,000 the optimum choice of M would be 20 with $\overline{N}$=2500. The effort E would then have been reduced from 674 to 399.2 million emd's.

Table II

Fortran Program No.	LDC	N	Modules	$\overline{N}$
1.	3345	17609	48	367
2.	685	3800	23	165
3.	2132	13442	49	274
4.	582	2831	14	202
5.	179	1156	1	1156
6.	192	1089	7	156
7.	111	647	2	324
8.	131	1077	2	539
9.	2559	15530	22	706
10.	227	1457	5	291
11.	81	763	4	191
12.	84	424	3	141
13.	55	353	2	177
14.	190	1594	14	114
15.	458	2975	4	744
16.	752	4518	6	753
17.	2042	14344	31	463
18.	1372	15704	18	872
19.	2164	15954	72	222
20.	2883	15437	35	441
21.	386	2122	11	193
22.	189	1422	1	1422
23.	1133	8088	14	578
24.	42	358	1	358
25.	90	731	2	366
26.	994	5328	20	266
27.	676	4481	13	345
28.	1825	12464	37	337
29.	360	2678	4	669
30.	1978	14869	44	338
31.	115	613	4	153
32.	110	604	3	201
Σ	28122	184462	516	14268
Mean		5764		446

A third set of data for validation of the model was provided by a large software house and is summarized in the table below:

Project	N	KLOC	#Modules	Avg. Mod. Size	T-MM
A	232K	45.4	176	1318	254.1
B	589K	116.6	354	1666	419.0

We ran this data through the unrestricted 2-level and 3-level model and selected the results which appeared to match most closely the reported effort. The best results are given below:

Project	Model-Level	$\bar{N}$	M	L	S	$\hat{T}$-MM
A	2	2549	13	7	0	207
B	3	1886	7	5	8	412

In these models we used $k_1 = 1$, $k_2 = 3/4$, $k_3 = 3/8$ for the unit interface effort calculation

The results show that for optimal effort Project A should be divided into 91 modules of average length 2549 while Project B should be divided into 280 modules of average length 1886. In both cases the predicted effort T is quite close to the reported effort T. An important decision by a model builder is to select the proper level. If we used a 2-level model for Project B for example the predicted effort T would have been 754 MM. It is evident that for very large projects a 3-level model is the most appropriate. On the other hand if we had used a 3-level model for Project A, the predicted effort T would have been 150 MM, a much worse result than the 2-level model gave. Some additional research is needed to decide on which level is most appropriate for projects of various sizes.

IX. The Restricted Modularization Problem

There is considerable intuitive evidence to support the practice that the average module or segment size should be restricted in length to between 50 and 80 lines of source code. Modules of this size are just within the immediate comprehension of the average programmer. Sullivan [ ] has shown that programs of this size are more likely to be error free than larger programs. Thus, both for reliability and ease of comprehension, software managers may limit the size of modules to the range between 50 and 80 LOC whether or not these correspond precisely to natural functional modules.

Even if one decides to restrict the average module size to be within a certain range, we must still decide on how many levels of modularization to use and on the number of modules at each level. The model described above can easily be modified to produce for a specified level the optimum choice of subprograms, modules and segments.

In what follows we assume that N and $\overline{N}$ are given and that all the other assumptions are maintained.

In the one level case since N and $\overline{N}$ are given M is determined from the equation

$$M\,\overline{N} = N$$

Hence, no optimization is possible. The total effort is given by

$$E = M\overline{e} + \tfrac{1}{2}\, M(M-1)\, I_1$$

In the two level case we have

(19a) $$ML\overline{N} = N$$

and we wish to find M and L which will minimize

(19b) $$E = ML\bar{e} + \tfrac{1}{2} M(M-1) I_1 + \tfrac{1}{2} ML(L-1) I_2$$

From (19a) L is determined once M has been found, hence E is essentially a function of one variable, say M. Hence, the minimization equation is

(20) $$\frac{dE}{dM} = \frac{d(ML\bar{e})}{dM} + \tfrac{1}{2} (2M-1)\ I_1 + \tfrac{1}{2} L(L-1)\ I_2 + \tfrac{1}{2} M(2L-1)\ I_2 \frac{dL}{dM} = 0$$

Now

$$\frac{d(ML\bar{e})}{dM} = 0$$

since $ML\bar{e}$ is a constant if $\bar{N}$ is fixed, and

since

$$L = \frac{1}{M}\ \frac{N}{\bar{N}}$$

then

$$\frac{dL}{dM} = -\frac{c}{M^2}$$

where we set $c = N/\bar{N}$.

Substituting into (20) and simplifying we obtain the equation

(21) $$2M^3 - M^2 = c^2\ I_2/I_1$$

Thus given $c=N/\overline{N}$, I_2 and I_1, we can solve (21) for M and then obtain L from $L=c/M$. These values of M and L will minimize the effort.

Similarly we can consider the restricted 3 level minimization problem. The equations which must be satisfied are

(22) $$MLS\overline{N}=N$$

$$E = MLS\overline{e} + \tfrac{1}{2} M(M-1)\ I_1 + \tfrac{1}{2} ML(L-1)\ I_2 + \tfrac{1}{2} MLS(S-1)\ I_3$$

There are essentially 2 independent variables, say M and L, since S is determined from $S = \frac{I}{ML}\ \frac{N}{\overline{N}}$.

We therefore have 2 minimization equations to satisfy:

(23a) $$\frac{\partial E}{\partial M} = \frac{2M-1}{2}\ I_1 + \frac{L(L-1)}{2}\ I_2 + \frac{c}{2}\ \left(\frac{-c}{M^2L}\right)\ I_3 = 0$$

(23b) $$\frac{\partial E}{\partial L} = \frac{M(2L-1)}{2}\ I_2 + \frac{c}{2}\left(-\frac{c}{ML^2}\right) I_3 = 0$$

where

$$c=N/\overline{N}$$

and where we have used the fact that $MLS\overline{e}$ is a constant.

Solving (23b) for M we obtain

(24a) $$M = \frac{c}{L\sqrt{2L-1}}\ \left(\frac{I_3}{I_2}\right)^{\frac{1}{2}}$$

and we can rewrite (23a) as

$$(24b) \qquad f(L) = (2M-1)\ I_1 + L(L-1)\ I_2 - \frac{c^2}{M^2 L}\ I_3 = 0$$

We can solve (24a) - (24b) by a secant type iteration as follows: Let L_{i-1}, L_i be 2 starting values; compute M_{i-1}, M_i from (24a). Evaluate $f(L_i)$, $f(L_{i-1})$ from (24b). Update L using

$$L_{i+1} = L_i - f(L_i)\ \frac{L_i - L_{i-1}}{f(L_i) - f(L_{i-1})}$$

and iterate until convergence is achieved.

X. Numerical Results for the Restricted Minimization Problem

In Table III we show the optimum value of M, L and S for various program sizes and for level 2 and level 3 modularization. We have not rounded off these values to their nearest integer since that would change the values for $\bar{N}$. However, in our discussion we will round mentally to the nearest integer.

To interpret the results we will examine in detail the case N=50,000. For $\bar{N}$=250 and 2-level modularization we would need M=14 and L=14 for effort minimization. As $\bar{N}$ increases, M and L both decrease gradually but still in such a way that M and L are approximately equal. For the same case 3-level modularization and any value of $\bar{N}$ between 250 and 400 it appears that we should choose M=3, L=6, S=9. Intuitively the 3-level model appears more natural for the N=50,000 case since it leads to a purer tree structure. Indeed it appears that for N > 20,000 a 3-level model is to be preferred. Of course as M approaches 1 as it does for N < 10,000 it is apparent that we must use a 2-level model, or even for very small programs, a 1-level model.

Table III

Restricted 2 and 3 Level Minimization Model ($k_1=1$, $k_2=\frac{1}{2}$, $k_3=\frac{1}{4}$)

	Level 2			Level 3		
N	$\bar{N}$	M	L	M	L	S
2K	250	1.68	4.78	.68	1.87	6.27
	300	1.51	4.42	.65	1.72	5.92
	350	1.38	4.14	.63	1.61	5.63
	400	1.28	3.90	.62	1.51	5.37
4K	250	2.63	6.08	.89	2.60	6.94
	300	2.35	5.67	.83	2.41	6.63
	350	2.14	5.34	.75	2.26	6.36
	400	1.98	5.06	.76	2.14	6.13
8K	250	4.18	7.66	1.23	3.42	7.59
	300	3.72	7.17	1.14	3.20	7.30
	350	3.37	6.77	1.07	3.02	7.05
	400	3.10	6.45	1.02	2.87	6.84
10K	250	4.85	8.24	1.38	3.71	7.80
	300	4.32	7.72	1.28	3.48	7.51
	350	3.92	7.30	1.19	3.29	7.27
	400	3.60	6.95	1.13	3.13	7.06
20K	250	7.76	10.31	2.01	4.70	8.47
	300	6.89	9.67	1.84	4.43	8.19
	350	6.24	9.16	1.71	4.20	7.96
	400	5.72	8.74	1.60	4.02	7.75
50K	250	14.47	13.82	3.38	6.28	9.43
	300	12.83	12.99	3.07	5.93	9.15
	350	11.59	12.32	2.84	5.65	8.91
	400	10.62	11.77	2.65	5.42	8.71
100K	250	23.16	17.26	5.05	7.73	10.24
	300	20.55	16.22	4.58	7.32	9.94
	350	18.56	15.40	4.22	6.99	9.70
	400	16.99	14.71	3.93	6.71	9.49
200K	250	37.13	21.54	7.59	9.48	11.12
	300	32.90	20.26	6.87	8.98	10.81
	350	29.71	19.24	6.31	8.58	10.55
	400	27.00	18.00	5.87	8.25	10.33

References

1. Lattanzi, L.D., An Analysis of the Performance of a Software Develop-Software Development Methodology, COMPSAC Proceedings, 1979.

2. Turner, J., The Structure of Modular Programs, CACM, May 1980, pp. 272-277.

3. Halstead, M.H., Elements of Software Science, Elsevier, North-Holland, 1977.