# Chapter 5 Software effort estimation

## Further exercises and pointers

*1. The size (that is, the effort needed to complete it) of any task will depend on its characteristics. The units into which the work is divided will also differ. Identify the factors affecting the size of the task and the work units for the following activities:*

- *installing computer workstations in a new office*
- *transporting assembled personal computers from the factory where they were assembled to warehouses distributed in different parts of the country*
- *typing in and checking the correctness of data that is populating a new data base*
- *system testing a newly written software application*

As will be seen the exact dividing line between what influences task size and what is a productivity driver can be hazy.

- Installing workstations.

Task size drivers might include:

  - Number of workstations to be installed
  - Number of locations at which workstations are installed
  - Geographical spread

Productivity drivers might include:

  - Novelty of the technology/experience of installers
  - Type of workstation
  - Availability of access to premises

- Transporting assembled personal computers

Task drivers might include

  - Number of units for estimating loading/unloading
  - Number of loads (units going to one warehouse) – group of units in a delivery
  - Distance of warehouse from factory – traveling time

Productivity drivers might include

  - Time per unit needed for loading
  - Size of vehicles
  - Average speed of delivery vehicles
  - Time per unit needed for unloading

- Typing in and checking input

Tasks size drivers might include:

  - Number of records to be input
  - Number of items per record to be input

Productivity drivers might include:

  - Clarity/availability of input documents
  - Experience/capability of inputters – this could affect the number of records that need correction
  - Ease of making corrections

- Systems testing

Task size drivers might include:

  - Number of test cases
  - Number of requirements – could be used to estimate number of test cases

- Items to be input, items output per test case

Productivity drivers might include:
- Ease of test set up
- Errors found – this will require retesting
- Availability of test automation tools

*2. If you were asked as an expert to provide an estimate of the effort needed to make certain changes to an existing piece of software, what information would you like to have to hand to assist you in making that estimate?*

The estimation of the effort to change a software component requires a different approach to the development of completely new software as a major factor is the structure of the existing software.
The estimator might need to know:
- The nature of the changes required
- Ideally they should be told about the reasons for the change, as they may be able to suggest alternatives and more economic ways of meeting the requirement
- The structure of the software to be changed
- Some idea of which parts of the software need changing
- Who will be carrying out the changes – will it be someone who is already familiar with the software to be changed, or will additional time have to be allocated for familiarization?

3. *A small application maintains a telephone directory. The database for the application contains the following data types:*

*Staff reference*
*Surname*
*Forenames*
*Title*
*Department code*
*Room number*
*Telephone extension*
*E-mail address*
*Fax number*

*Transactions are needed which:*
- *i) set up new entries;*
- *ii) amend existing entries*
- *iii) delete entries*
- *iv) allow enquirers to list on line the details for a particular member of staff*
- *v) produce a complete listing of the telephone directory entries in alphabetical order*

*a) Use this scenario to produce an estimated Mark II FP count. List all the assumptions you will need to make.*
*b) Another requirement could be to produce the listing in (v) in departmental order. In your view should this increase FP count and if so by how much?*

a)

| *transaction* | *inputs* | | *outputs* | | *entities accessed* | |
|---|---|---|---|---|---|---|
| set up new entry | staff reference *to* fax number *in data type list* | 9 | error message | 1 | directory entry | 1 |
| amend (display) | staff reference | 1 | surname *to* fax number *or* error message | 9 | directory entry | 1 |
| amend (update) | surname *to* fax number | 8 | error message | 1 | directory entry | 1 |
| delete entry | staff reference | 1 | surname, forenames (as check) error message | 3 | directory entry | 1 |
| enquiry | staff reference or surname, forenames | 3 | full details + error message | 10 | directory entry | 1 |
| listing | trigger | 1 | full details | 9 | directory entry | 1 |
| totals | 23 x 0.58 = 13.34 | | 33 x 0.26=8.58 | | 7x 1.66 =11.62 | |
| grand total | | | | | | 33.54 |

b) It can argued that the data presented in the two reports are the same, so that they are logically the same and so should be counted only once. Most development environments have easy to use features for sorting data and so the actual amount of work needed to produce the second report would probably be negligible. However, if you have every tried to look up details in an unsorted list, the fact that it is sorted can provide considerable value.

There is therefore a debate among FP practitioners about whether FPs should reflect the value of the system to the user or the amount of work that is needed to develop it. This is particularly an issue in relation to the question of reusable components.

4*. The following details are held about previously developed software modules.*

| *module* | *inputs* | *entity types accessed* | *outputs* | *days* |
|---|---|---|---|---|
| *a* | *1* | *2* | *10* | *2.60* |
| *b* | *10* | *2* | *1* | *3.90* |
| *c* | *5* | *1* | *1* | *1.83* |
| *d* | *2* | *3* | *11* | *3.50* |
| *e* | *1* | *3* | *20* | *4.30* |

*A new module has 7 inputs, one entity type access and 7 outputs. Which of the modules a to e is the closest analogy in terms of Euclidean distance?*

| module | inputs | entity types accessed | outputs | days | euclidean distance from new |
|---|---|---|---|---|---|
| a | 1 | 2 | 10 | 2.6 | 6.78 |
| b | 10 | 2 | 1 | 3.9 | 6.78 |
| c | 5 | 1 | 1 | 1.83 | 6.32 |
| d | 2 | 3 | 11 | 3.5 | 6.71 |
| e | 1 | 3 | 20 | 4.3 | 14.46 |
| | | | | | |
| new | 7 | 1 | 7 | | |

Module c would appear to provide the best analogy as it is at the least Euclidean distance from the new module. This provides a base estimate of 1.83 days

*5. Using the data in further exercise 4 above, calculate the Simons Mark II FPs for each module. Using the results, calculate the effort needed for the new module described in additional exercise 4. How does this estimate compare to the one based on analogy?*

| module | inputs | entity types accessed | outputs | days | FPs |
|---|---|---|---|---|---|
| a | 1 | 2 | 10 | 2.6 | 6.50 |
| b | 10 | 2 | 1 | 3.9 | 9.38 |
| c | 5 | 1 | 1 | 1.83 | 4.82 |
| d | 2 | 3 | 11 | 3.5 | 9.00 |
| e | 1 | 3 | 20 | 4.3 | 10.76 |
| totals | | | | (a) days | (b) fps |
| | | | | 16.13 | 40.46 |
| productivity FPs /per day (b/a) | | | | | 2.51 |
| new | 7 | 1 | 7 | new FPs ( c ) | 7.54 |
| | | | | | |
| estimate for new (c/(b/a)) | | | | | **3.01** |

Note that this is higher than that estimate given in 4. Partly this is because the new project may be closest to module c but it is bigger in terms of the number of inputs and outputs that it has. It also has an estimate bigger than that for module a because Mark II FPs assume that input procedures are more difficult to implement than outputs and therefore gives them a higher weighting.

6. Given the project data below:

| *project* | *inputs* | *outputs* | *entity accesses* | *system users* | *program-ming languag e* | *developer days* |
|---|---|---|---|---|---|---|
| *1* | *210* | *420* | *40* | *10* | *x* | *30* |
| *2* | *469* | *1406* | *125* | *20* | *x* | *85* |
| *3* | *513* | *1283* | *76* | *18* | *y* | *108* |
| *4* | *660* | *2310* | *88* | *200* | *y* | *161* |
| *5* | *183* | *367* | *35* | *10* | *z* | *22* |
| *6* | *244* | *975* | *65* | *25* | *z* | *42* |
| *7* | *1600* | *3200* | *237* | *25* | *y* | *308* |
| *8* | *582* | *874* | *111* | *5* | *z* | *62* |
| *X* | *180* | *350* | *40* | *20* | *y* | |
| *Y* | *484* | *1190* | *69* | *35* | *y* | |

Note X and Y are new projects for which estimates of effort are needed.

*a) What items are size drivers?*
inputs, outputs, entity accesses (system users for certain aspects) – the more of these there are the larger the development job

*b) What items are productivity drivers?*
Programming language – the number of lines of code that can be produced in a day will depend, in part, on the programming language

*c) What are the productivity rates for programming languages x, y and z?*

x 10 FPs a day
y 7 FPs a day
z 12 FPs a day

*d) What would be the estimated effort for projects X and Y using a Mark II function point count?*

| project | inputs | outputs | entity accesses | system users | program-ming language | developer days | FPs |
|---|---|---|---|---|---|---|---|
| 1 | 210 | 420 | 40 | 10 | x | 30 | 297.4 |
| 2 | 469 | 1406 | 125 | 20 | x | 85 | 845 |
| | | | | sub-totals | | 115 | 1142.4 |
| | | | | productivity (FPs/day) | | 9.9 | |
| 3 | 513 | 1283 | 76 | 18 | y | 108 | 757.15 |
| 4 | 660 | 2310 | 88 | 200 | y | 161 | 1129.48 |
| 7 | 1600 | 3200 | 237 | 25 | y | 308 | 2153.42 |
| | | | | sub-totals | | 577 | 4040.05 |
| | | | | productivity (FPs/day) | | 7.0 | |
| 5 | 183 | 367 | 35 | 10 | z | 22 | 259.66 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 6 | 244 | 975 | 65 | 25 | z | 42 | 502.92 |
| 8 | 582 | 874 | 111 | 5 | z | 62 | 749.06 |
| | | | | sub-totals | | 126 | 1511.64 |
| | | | | productivity (FPs/day) | | 12.0 | |
| | | | | | | | |
| X | 180 | 350 | 40 | 20 | y | | **261.8** |
| Y | 484 | 1190 | 69 | 35 | y | | **704.66** |

The above table gives the FP counts. Using the productivity rate for programming language y, the estimate for Project X would be 262/7 i.e. 37 days, and for Project Y 705/7 i.e. 101 days

*e) What would be the estimated effort for X and Y using an approximate analogy approach?*
Project X seems closest to Project 5 which provides an estimate of 22 days, and Project Y seems to be closest to Project 3 which gives an estimate of 108 days
.
*f) What would have been the best estimating method if the actual effort for X turns out to be 30 days and for Y turns out to be 120 days? Can you suggest why the results are as they are and how they might be improved.*
If we use a measurement based on the percentage error, calculated as absolute (actual-estimate)/actual we get the following:

| | Project X | Project Y |
|---|---|---|
| FP method | 23% | 16% |
| analogy | 27% | 10% |

This illustrates that no one estimating method can be assumed to be the most accurate in all cases.

The analogy estimate for Project X might have been improved by adjusting for the fact that Project X is to be written in programming language y while Project 5 was written in programming language z

*7. A report in a college time-tabling system produces a report showing the students who should be attending each time-tabled teaching activity. Four files are accessed: the STAFF file, the STUDENT file, the STUDENT-OPTION file and the TEACHING-ACTIVITY file. The report contains the following information:*

*Teaching activity reference*
*Topic name*
*Staff forenames*
*Staff surname*
*Title*
*Semester (1 or 2)*
*Day of week*
*Time*
*Duration*
*Location*
*For each student:*

*student forename*
*student surnames*

*Calculate the Mark II FPs that this transaction would generate. Can you identify the factors that would tend to make the two methods generate divergent counts?*

**Mark II**

| | | | |
|---|---|---|---|
| Inputs | 1 ( a trigger) x 0.58 | = | 0.58 |
| Entity accesses | 4 x 1.66 | = | 6.64 |
| Outputs | 12 x 0.26 | = | 3.12 |
| | | | |
| Total | | | **10.34** |

*8. Suppose you are the manager of a software project. Explain why it would not be proper to calculate the number of developers required for the project as a simple division of the effort estimate (in person-months) by the nominal duration estimate (in months).*

A simple division of total (estimated) effort by the total (estimated) duration implicitly assumes constant team size. But, the team size should show a Raleigh distribution for efficient utilization of the manpower. So, a constant team size arrived from a simple division of the effort estimate (in person-months) by the nominal duration estimate (in months) would lead to wastage of manpower and consequent increase of cost and duration.

*9. Suppose that a certain management information system (MIS) software product costs £50,000 to buy off-the-shelf and that its size is 100 kdsi. Assuming that in-house developers cost £2000 per programmer-month (including overheads), would it be more cost-effective to buy the product or build it? Which elements of the cost are not included in COCOMO estimation model? What additional factors should be considered in making the buy/build decision?*

Using COCOMO equation we get the effort estimate as $2.4*(100)^{1.05}$ person-months.
Thus, the nominal effort required to develop the product is 302.14 person-months.
In-house developers cost £2000 per programmer-month (including overheads).
Therefore, cost to develop the product would be 302.14*£2000= £604,280

However, before a buy or build decision is made, the following points must be taken into account.

- How quickly is the product required? It may take nearly 22months (by using COCOMO expression for duration) to develop the product.
- Number of copies required.
- Whether the developed product can be sold?
- Whether any maintenance of the product be required in the future and what would be the cost of the maintenance?
- The risks associated with development such as schedule slippage, cost escalation, etc.
- Would the experience gained in the product development be of use?