Software Development Projects - Portfolio Selection & Staffing with Tool Support

Maria Coyle and William G. Tuohey*

School of Computing, Dublin City University, Glasnevin, Dublin 11, Ireland

Abstract

Staffing challenges faced by software development organisations are identified. The need for a strategic view to meet these challenges, based on project portfolio management, is highlighted. To be effective, portfolio management should be supported by suitable tools. A specific tool is presented which it is believed can be of practical benefit for decision makers on project selection and staff recruitment.

Author Keywords: Staffing challenges; Software development; Project portfolio management; Decision support tool, Implementing strategy, Human resource management

Introduction

Staffing challenges faced by software development organisations are introduced in the context of the changing economic environment. Existing literature on software project portfolio management is reviewed briefly. Some of the problems that may arise, and the corresponding need for an overarching strategic framework, are noted. The ***PortMan*** tool to support project portfolio selection is presented, including an overview of its architecture, data model and key functionality. This is followed by a summary of how ***PortMan***'s capabilities have been validated. The final section contains conclusions and notes some areas for future work.

Strategic considerations for staffing

Consideration of employment and revenue trends in the Irish software industry between 1991 and 2003 provides motivation and background for the research reported in this paper. The period 1991–1999 was an expansionary one for that industry with a corresponding need for more skilled employees. Indeed, statistics reported in [1] reveal an almost perfect correlation between revenue and employment growth, suggesting that revenue growth may have been limited by an inability to recruit suitable employees. In the period 1999-2000, revenue growth

* Corresponding author
Tel: +353-1-7008728
FaX: +353-1-7005442
Email: ltuohey@computing.dcu.ie

continued to increase but employment growth slowed. Employment actually contracted between 2001 and 2003, whereas revenue growth slowed in 2001-2002 but began to increase again in 2002-2003. In summary [1], "The industry has reacted to the world slowdown by cutting costs, including staffing costs. This has enabled it to ride out the storm and to return to growth in the current year (2004)."

These trends illustrate two scenarios:

- lack of skilled employees constraining revenue generation
- over-supply of labour resulting in employee redundancy.

that present staffing challenges for a software company:

1) How many resources are needed to meet its current workload?
2) When and how many resources will it need for future projects it would like to pursue?
3) Can it relate its changing work portfolio to its available staff profile?
4) What is the gap between its staff level and the effort required for its work portfolio?

In general, and particularly for the software industry, when a company loses staff or has staff lay-offs there are numerous intangible losses to the company. For this reason, it is argued in [2] that treating staff costs as expenses, as accounting convention dictates, is an inadequate approach for the software industry; instead, they should be treated as an investment. While staff do not conform to all elements of the normal accounting definition of an asset[1], it is clear that retention of staff must be an important strategic consideration when selecting project portfolios.

Project portfolio management, a brief review

Of several possible objectives for a firm, McLaney [4] suggests that on-going "maximisation of shareholders' wealth" (rather than say "maximisation of profit") is the most credible in terms of taking account of both return and risk, and in terms of balancing short- and long-term benefits[2]. Therefore, the decision on whether or not to pursue a specific project should not be taken in isolation - the project must be evaluated in relation to time and in relation to

[1] An asset is an item that has the following attributes [3]:
- It is likely to produce future economic benefits
- It has arisen from a transaction involving the firm in the past
- The right of access to it can be controlled by the firm.

[2] The "wealth" of a company is defined as the sum of the value of all its assets less its liabilities, while a company's "profit" is the net increase in its wealth over a defined period.

available alternatives. For example, in the context of bidding for software projects, [5] takes the view "that failure to consider the portfolio viewpoint can damage the bidding process and result in unexpected knock-on effects on other projects".

Project portfolio management should, in principle, be a mechanism for ensuring that an organisation selects the right projects within a strategic, long-term framework. However, practical problems commonly arise which, for the case of internal development projects, are summarised in [6] under six headings, as follows. These are (a) **Management of project-oriented business** (low priority for projects, rapid and continuing change, confusion on portfolio ownership), (b) **Commitment, rôles and responsibilities** (lack of clarity between portfolio, project and other decision-makers), (c) **Information management** (lack of definition or complete absence of information flows especially on projects, lack of a common database of projects), (d) **Resources, competencies and methods** (inadequate methods and guidelines for portfolio and project management, staff shortages), (e) **Portfolio level** (overlapping projects and tasks, weak project initiation (GO) and termination (KILL) decision-making), and (f) **Project level** (poor project definition and monitoring, overlong projects).

In [7], an integrated framework for project portfolio selection is described which addresses several of the problems listed in the previous paragraph - see [8] also for somewhat similar ideas. The framework, based on a set of 11 propositions, consists of three phases: **strategic considerations**, **individual project evaluation** and **portfolio selection**. It is envisaged that "optimal portfolio selection" takes place as part of a well-defined sequence of activities. Thus, it is proposed that (i) each project proposal be given an initial screening to assess its suitability; then, (ii) suitable projects should be subjected to detailed individual analyses after which (iii) it is again considered whether they should be screened out. It is only at this stage that (iv) the remaining "screened-in" projects are submitted to the "optimal portfolio selection" process. Strategic considerations are enforced in that they are the basis for definition of objective project screening criteria and of more detailed portfolio and project evaluation methods, as well as for overall budgetary parameters. It is also required that existing projects be included in the regular portfolio selection updates. A decision support system for portfolio selection is described in [7], a key element of which is a central portfolio database to serve as a data repository as well as a transfer site for all the constituent models and methods.

Reference [9] reports findings on staff allocation in a multi-project environment within an R&D setting. Such a setting is comparable to software development, particularly in the sense that staff are the main resource. A three-level human resource allocation plan is proposed, consisting of long (yearly), medium (quarterly) and short (biweekly) term processes. The long term process is concerned with resourcing the organisation's strategic business plan, while the medium term process' output is a "rough cut capacity planning" for the project portfolio. The short-term process is to do with day to day allocation of staff by project leaders. There is regular feedback and associated adjustments between short term and medium term, and between medium and long term plans. It appears that it is within the strategic business plan that challenges such as (1) to (4) of section “Strategic considerations for staffing” of this paper should be addressed.

Two recommendations, significant for human resource planning, emerge in [9]:

> (1) Split staff into three categories namely, "all-round project members", "experts" and "service employees"
>
> (2) Minimise project scatter factor[3] - as far as possible, allocate each project "all-rounder" to a single project fulltime. On the other hand, "experts" will be needed from time to time in all projects; "service employees" perform routine activities that are needed in all projects.

As noted in [9] "efforts to get efficiency between projects is far less effective than the efficiency that can be gained within each project itself". Clearly, allocation of project staff will be much simplified if recommendations (1) and (2) can be put in place. Moreover, it seems intuitively clear that these recommendations, if implemented appropriately, would lead to increased team cohesion ([10], [11]), leading to greater job satisfaction and hence to increased staff retention.

PORTMAN tool to assist in portfolio selection

PortMan Context and overview

This paper’s main focus is on the portfolio selection process. In particular, a software tool (called ***PortMan***, [12]) to support that process is presented. Thus, following [6], the paper is concerned particularly with alleviating aspects of problem areas (c), (d) and (e), as listed in the previous section. ***PortMan*** fits neatly within the integrated framework of [7] supporting,

[3] The number of staff needed to fulfil a one person-year task; for example, if 20 staff (part time) do a four person-year job then the project scatter factor is 5.

as detailed below, the “optimal portfolio selection process” (activity (iv) of [7]). It is noted that, if recommendations (1) and (2) of [9] (see under “Strategic considerations for staffing”, above) are adopted then utilization of tools such as ***PortMan*** should be both simpler, in terms of reduced inter-project coupling, and more effective.

PortMan is a portfolio based project evaluation and maintenance system designed for Software Development Houses. It provides users with functionality to evaluate the feasibility and desirability of adding potential projects to their existing portfolio. The system is designed with commercial enterprises in mind so that the ultimate measure used to rate the desirability of a project or set of projects is the projected margin of profit for the company.

PortMan is a project selection rather than a project management tool (such as Microsoft Project). As such it seeks to select the most profitable set of projects an organisation can achieve taking account of:

- The resources required to deliver the current project portfolio.
- The resource availability within the time-period defined by the deadlines of the potential projects under consideration.

PortMan therefore needs to know about projects that are live, those that are potential and those that having been considered are deemed unachievable. This information is recorded in a “project status table” within the system.

PortMan sees the future working time of employees, from their project finish date forward, as unassigned. The working time of these unassigned employees provides the capacity to undertake some, all or none of the potential projects available to the organisation. Knowing how much capacity is becoming available, and when, is key to ensuring the long-term profitability of an organisation.

If an organisation chooses to pursue a group of projects, ***PortMan*** needs to update its schedule to maintain an accurate measurement of the time left available for other potential projects. To facilitate this, some simple functionality is provided to allow the user assign un-booked employee time in the future to the selected projects. This functionality is deliberately limited as detailed assignment of employee time is more properly dealt with during the project planning and project management stages (cf [9]). For ***PortMan*** to advise users on project selection, it needs to know how much employee time is un-booked between the present and the deadline for a potential project. ***PortMan*** provides a capability for a user to assign employees to a project as they become available, over a number of different weeks

say; thus a project will not be prevented from starting just because its full team is not yet available.

PortMan Architectural overview

PortMan, written in Microsoft Access, consists of two MDB files. One called PortMan.MDB acts as a front-end. It contains all forms, reports etc. used to provide the system GUI. It is written in VBA and uses embedded SQL to query a back-end database. The second MDB file PortManDB.MDB is used as the back-end database for the system. Microsoft Access allows tables from another MDB to be linked to very simply. Advantages of splitting the system data from the system GUI include,

- If the front-end GUI becomes corrupt, the system data are not automatically corrupted.
- Enhanced functionality can be added to the front-end without changing the database.
- It facilitates networking – may store system data on a server, front end on client machines.

While ***PortMan*** version 1.0 may be considered a prototype whose modular structure facilitates replacement of its front or back end or both, there are good arguments for retaining the Microsoft Access architecture:

Cost

Moving to a more elaborate development platform such as VB/Java and a larger database such as Oracle/SQL Server would increase the cost of the system greatly. It is debatable that this can be justified. A very small number of people would ordinarily be involved in making the final decisions on what projects will be pursued by an organisation. Therefore, there is no strong argument for ***PortMan*** to be scalable or capable of handling a large number of users simultaneously. The notion that all software should come in the form of n-tier enterprise level applications is an expensive fallacy. Much value can be obtained through the intelligent development and deployment of small, inexpensive tools that support larger systems.

Compatibility

PortMan does not attempt to “re-invent the wheel”. It relies on input that needs to be generated externally including, in particular, the size of each project and the margin it will contribute to the organisation. MS Access interfaces very easily with commonly used accounting spreadsheet software such as MS Excel and Lotus 1-2-3. It also interfaces easily with MS Project. Some of the scheduling information used in ***PortMan*** could conceivably be imported from MS Project. It can also link using ODBC to larger databases such as Oracle and SQL Server and can easily import data from a variety of file types including HTML, fixed width and comma delimited text files. So whatever software an organisation uses to

estimate the size of projects and generate its pricing, there is a high degree of likelihood that it will be capable of outputting results in a format that MS Access can import if not link to directly. This makes it an extremely agile tool capable of harvesting information from a number of different technologies.

Development Time

MS Access is a very suitable environment for Rapid Application Development. This will help contain costs and increase the efficiency with which changes can be made and enhancements added.

PortMan Data model

PortMan's data model needs to be capable of storing enough data to generate all of the following:

- The development capacity of the organisation
- The value of the potential projects to the organisation
- The development effort these projects would require
- The time constraints on delivering the potential projects
- The total effort required to deliver the current projects in the organisation's portfolio
- The free capacity available to undertake the new projects
- Time(s) at which more free capacity becomes available (requires a work schedule)
- Does the flow of free development capacity allow the time constraints of a potential new project to be met?

PortMan stores information on the user Organisation and their Clients, Staff, Projects (live and potential) and a Schedule (recording staff utilisation on a weekly basis) in a set of six tables depicted in Figure 1.

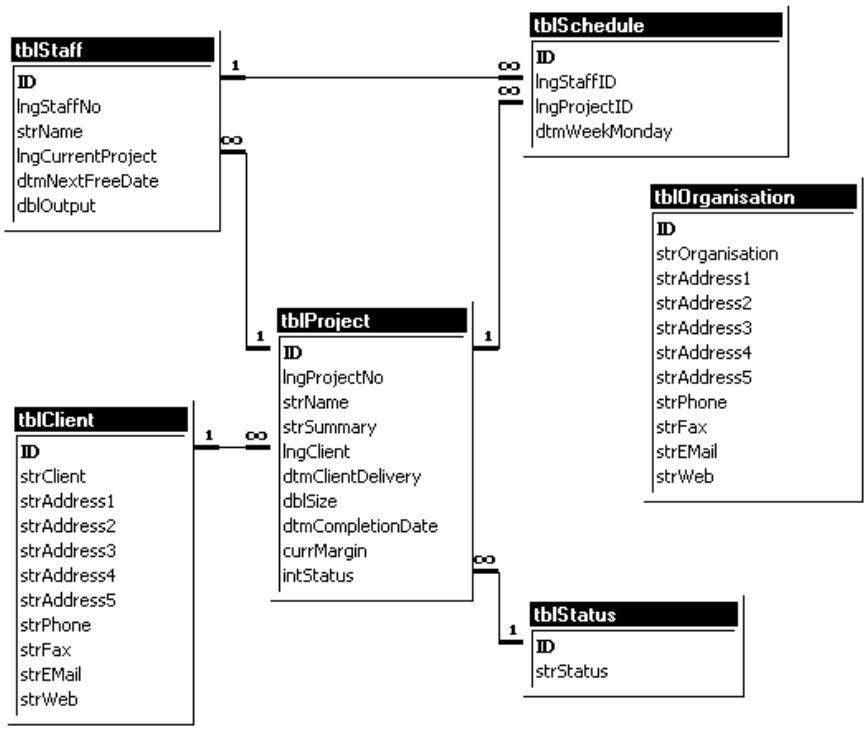


Figure 1: The ***PortMan*** Entity Relationship Diagram

Three of the tables (tblProject, tblStaff, tblSchedule) provide most of the information, see [12].

Field Name	**Description**
ID	Unique ID
LngProjectNo	Project Number
StrName	Project Name
StrSummary	Summary description of the project
LngClient	Client – Looked up from the Client Table
DtmClientDelivery	Delivery date specified by client (*absolute deadline*)
DblSize	Size of project
DtmCompletionDate	Expected project completion date ($\leq$ *DtmClientDelivery*)
CurrMargin	Expected profit margin this project will generate
IntStatus	Status– Looked up from the Status Table

Table 1(a): Fields within TblProject

Table 1(a) stores information on all current and potential projects. The projects marked live (IntStatus) constitute the current portfolio of work. DblSize is used to store the size of the project. Currently the system expects this to be expressed in Lines of Code, a unit of measurement that is commonly used in the industry, for example in association with Functional Point Analysis [13] and CoCoMo [14]-[16]. CurrMargin is the revenue the company expects to make from the project; it enables the system to identify the most profitable projects for the company within the portfolio. It is assumed that this is calculated elsewhere and supplied to ***PortMan***. An internal or expected completion date (DtmCompletionDate) is included to enable modeling of a realistic duration for such pre-release activities as QA audits or acceptance testing.

Field Name	**Description**
ID	Unique ID
LngStaffNo	Staff Number
StrName	Full name
LngCurrentProject	Current Project – Looked up from Project Table, tblProject
DtmNextFreeDate	Date the staff member is next free
DblOutput	Output in Lines of Code Per Week

Table 1(b): Fields within TblStaff

Table 1(b) stores the details of the organisation's employees. Fields DblOutput and DtmNextFreeDate together enable ***PortMan*** to determine how much development capacity is being used or is becoming free. The fact that ***PortMan*** provides a means of storing the output of each employee individually, rather than using averages, enables a user to overcome two common price estimation problems, namely the variation between programmer performances and not making allowances for non-programming tasks.

Field Name	**Description**
ID	Unique ID
LngStaffID	Employee – Looked up from the Staff Table
LngProjectID	Project – Looked up from the Project Table
DtmWeekMonday	The date on which Monday falls for the week

Table 1(c): Fields within TblSchedule

Table 1(c) maps staff to the project on which they will work each week of the next two years. It has one record a week for each staff member describing the project they are working on and giving the date of the first day of the week. On weeks where there is as yet no live project assigned to the employee, project zero is assigned.

PortMan Key functionality

The key functionality offered by ***PortMan*** is to advise users on what projects are most complementary to their existing portfolio. A core notion is that all projects a software development house can potentially take on are not equal. Some will be more beneficial, normally meaning more profitable, than others. Others, while highly profitable, will be beyond the organisation's resources to deliver in time. Even when the most profitable potential project is achievable with available resources, it needs to be considered if a combination of other potential projects that are less profitable individually could collectively be:

- Achievable in the given same time span
- Deliver more profit in total.

PortMan takes a portfolio approach to advising users on project selection. In essence this means that, constrained by an organisation's existing commitments, it takes account of the following factors when considering potential projects:

Time: Available time as determined by the client-specified project delivery deadlines.

Productivity: Available development capacity for the considered planning horizon.

Size: Size of the potential projects.

Using these factors ***PortMan*** can identify:

- Achievable (feasible) projects – satisfying capacity constraints.
- Best combination of achievable projects in terms of delivering maximum profit.

The heuristic algorithm defined in Figure 2 is used.

ROUND 1

1. Order potential projects by margin (profitability) so that the most profitable projects bubble to the top of the list. This ensures that as ***PortMan*** attempts to identify and reserve time for projects it does so in a way that prioritises those that are most profitable.
2. Assume that the next viable date to start any of the Potential Projects is (the) next Monday. This gives a value for earliest start date.
3. For each project in the list, beginning at the first (most profitable) one:
   DETERMINE IF THE PROJECT IS FEASIBLE (see below)
   If it is feasible, add it to the list of feasible projects

ROUND X

1. Take the Round X-1 list of projects.
2. Take the first project in that list.
3. Remove it from the list to be assessed.
4. Assume that the next viable date to start any of the Potential Projects is (the) next Monday. This gives a value for earliest start date.
5. For each project in the new list, beginning at the first (most profitable remaining) one:
   DETERMINE IF THE PROJECT IS FEASIBLE (see below)
   If it is feasible, add it to the list of feasible projects
6. This is the end of the current round. Its results are stored.

DETERMINE IF THE PROJECT IS FEASIBLE (sub-algorithm)

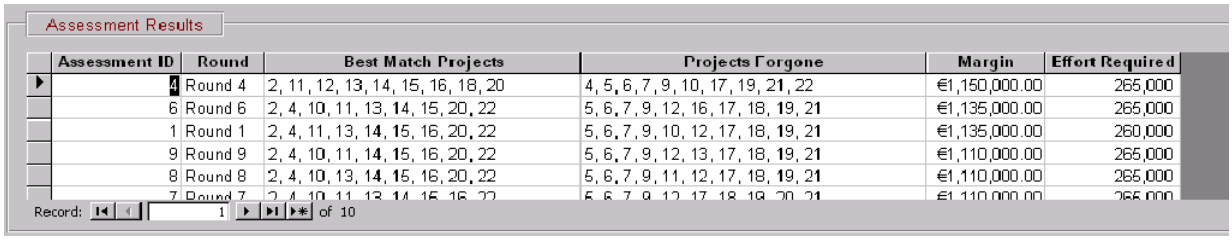
A. Identify the internal completion date (dtmCompletionDate) of the project.
B. Using the earliest start date and the project completion date as parameters, query the schedule to establish the total staff LoC that are unassigned in this period. This is the maximum spare capacity for the period.
C. Determine what the available spare capacity for this project is by subtracting any capacity that has already been committed to feasible potential projects so far (capacity used so far) from the maximum spare capacity. (For the first project assessed the available capacity will always be the same as the maximum spare capacity as no feasible projects will have been identified at that point).
D. If the Available Capacity is greater than the Project size then this is a feasible project.

Figure 2: Algorithm to determine optimal combination of feasible projects

Completion of ROUND 1 of this algorithm results in identification of the list of most profitable projects that are feasible for the organisation. Let ***TotalFeasible*** = Number of elements in this list.

However, this is not necessarily the most profitable set of projects overall. A combination of two or more projects, that individually have smaller margins than one of the projects on the list, might be feasible and contribute a greater total margin. For this reason, ***PortMan*** takes the list of projects established at the end of Round 1 (more generally, at the end of the previous round) and attempts to identify what other projects could be done if one project at a time is excluded from consideration. This is the role of ROUND X (see Figure 2) which is repeated for X = 2 to ***TotalFeasible***.

At the end of this process ***PortMan*** takes all of the lists it has compiled as being feasible and sorts them by the total combined margin (profit) each list would offer so that the most profitable list bubbles to the top. This information is displayed in the "Assess Potential Projects" screen results section (see Figure 3 for an illustration).


Assessment Results

Assessment ID	Round	Best Match Projects	Projects Forgone	Margin	Effort Required
4	Round 4	2, 11, 12, 13, 14, 15, 16, 18, 20	4, 5, 6, 7, 9, 10, 17, 19, 21, 22	€1,150,000.00	265,000
6	Round 6	2, 4, 10, 11, 13, 14, 15, 20, 22	5, 6, 7, 9, 12, 16, 17, 18, 19, 21	€1,135,000.00	265,000
1	Round 1	2, 4, 11, 13, 14, 15, 16, 20, 22	5, 6, 7, 9, 10, 12, 17, 18, 19, 21	€1,135,000.00	260,000
9	Round 9	2, 4, 10, 11, 14, 15, 16, 20, 22	5, 6, 7, 9, 12, 13, 17, 18, 19, 21	€1,110,000.00	265,000
8	Round 8	2, 4, 10, 13, 14, 15, 16, 20, 22	5, 6, 7, 9, 11, 12, 17, 18, 19, 21	€1,110,000.00	265,000
7	Round 7	2, 4, 10, 11, 13, 14, 15, 16, 22	5, 6, 7, 9, 12, 17, 18, 19, 20, 21	€1,110,000.00	265,000

Record: 1 of 10

Figure 3: Illustrative results of ***PortMan***'s assessment of potential projects

In the example of Figure 3, we see that the most profitable set of feasible projects was established in Round 4, with Round 1 (the list of most profitable projects that are feasible) only coming in third over all.

Note: Figure 3 illustrates the desirability of a simple enhancement of ***PortMan***, namely that in the event of a tie under profit "Margin", rounds should be ranked by "Effort Required". If this had been implemented, Round 1 would be ranked second, as it requires less effort than Round 6.

Summary of *Portman* validation results

Reference [12] provides a detailed account of the results of validation tests performed on the ***PortMan*** tool. A summary of the findings is presented in Table 2.

No	Feature	Validation outcome
1	Development capacity	***PortMan*** identifies the development capacity of an organisation and displays it
2	Potential value	***PortMan*** identifies potential projects, displays them and calculates their value
3	Required effort	***PortMan*** identifies potential projects, displays them and calculates the effort they require
4	Time constraints	***PortMan*** identifies potential projects and displays the delivery dates required by each
5	Total current effort	***PortMan*** identifies current live projects, displays them and calculates the effort they are taking
6	Free capacity	***PortMan*** identifies how much development capacity is unassigned currently
7	When more capacity?	***PortMan*** identifies how much development capacity is unassigned currently and in the future
8	Project feasibility	***PortMan*** identifies what potential projects can be achieved with current resources and what potential projects cannot be achieved
9	Advisory only	User can take a different course of action to the one recommended as best by the system
10	Feasible list	***PortMan*** compiles a list of projects that are feasible
11	Most profitable set	***PortMan*** identifies the possible combinations of projects that are feasible with current resources and ranks them by profitability
12	Total value foregone	***PortMan*** identifies the total value of the projects foregone (because of lack of staff)
13	How many extra staff?	***PortMan*** estimates, based on the average output of a member of staff, how many new developers would need to be employed to facilitate undertaking the projects foregone

Table 2: Summary of successful validation tests on ***PortMan*** [12]

This table indicates that ***PortMan*** can indeed help to alleviate several problems that can arise in project portfolio management (see under “Project portfolio management, a brief review”, above). In particular, ***PortMan*** provides visibility on an organisation’s current capacity, both committed and free, it determines what combinations of potential projects are feasible and

which of these is the most profitable, and it indicates how many new staff would be required to accommodate projects foregone due to capacity limitations.

Conclusions and Future Work

Software development organizations should take a long term, strategic view when allocating staff for current and future projects. A well thought-out process for management of a company's current and potential portfolio of projects is essential. Such a process should be supported by suitable software tools. A particular such tool ***PortMan*** has been presented in this paper which it is believed can indeed enhance the portfolio management process.

PortMan is a relatively simple tool based on quite standard and commonly available technology. This is seen as being to its advantage in terms especially of computability with most common office software. Nevertheless, there is scope for future development in terms for example of exploiting products such as MS Project to enhance the information ***PortMan*** can illustrate, of providing reports in HTML, or in making ***PortMan*** platform independent (at present it runs on MS Windows only).

Another area for future work would be further investigation of the algorithm (Figure 2) to assess the best projects set to an existing portfolio. It is possible that investigation of the extensive body of literature on job shop scheduling algorithms could help identify refinements of the ***PortMan*** algorithm to improve its performance.

An important area for future work is to evaluate how ***PortMan*** performs in practice. It is very desirable that an experiment be conducted whereby ***PortMan*** would be used and assessed by a selection of software development companies for a trial period. ***Portman***'s relative simplicity and its lack of a need for expensive infrastructure should increase the willingness of companies to participate. This is likely to be particularly true of small companies which, at least in Ireland, are in the large majority.

References

[1] Enterprise Ireland, Software Industry Statistics for 1991-2003, http://www.nsd.ie/htm/ssii/stat.htm (accessed December 17, 2004).

[2] DeMarco, T. and Lister, T., Peopleware: Productive Projects and Teams, 1999, 2nd Edition, New York : Dorset House Publishing.

[3] Reid, W. and Myddelton, D. R., The Meaning of Company Accounts, 1996, 6th Edition, Brookfield Vermont : Gower.

[4] McLaney, E. J., Business Finance – Theory and Practice", 1997, 4th Edition, London : Pitman Publishing.

[5] Pickard, L., Kitchenham, B., Linkman, S. and Jones, P., Can Software Bidding Practices be Improved? - Bidding and Portfolio Management Practices in Software and Other Industries, *Technical Report TR/SE-0003,* 2000, Keele: Keele University, http://www.keele.ac.uk/depts/cs/se/e&m/tr0003.pdf (accessed December 20, 2004).

[6] Elonen, S. and Artto, K. A., Problems in managing internal development projects in multi-project environments, *International Journal of Project Management*, 2003, **21**, 395-402.

[7] Archer, NP and Ghasemzadeh, F., An integrated framework for project portfolio selection, *International Journal of Project Management*, 1999, **17**(4), 207-216.

[8] Light, M., Project portfolio management, Gartner Symposium ITXPO, San Diego, California, 23-27 March 2003, http://symposium.gartner.com/docs/symposium/2003/spg5/documentation/spg5_17c.pdf (accessed December 15, 2004).

[9] Hendriks, MHA, Voeten, B. and Kroep, L., Human resource allocation in a multi-project R&D environment, *International Journal of Project Management*, 1999, **17**(3), 181-188.

[10] Hoegl, M. and Proserpio, L., Team member proximity and teamwork in innovative projects, Research Policy, 2004, **33**, 1153-1165.

[11] Jones, M. and Harrison, A., IS project team performance: an empirical assessment, Information & Management, (1996), **31**, 57-65.

[12] Coyle, M., Staffing challenges, work portfolio management and project selection in the software industry, M.Sc. in Information Technology dissertation, (2004), Dublin City University.

[13] Endres, A. and Rombach, D., A Handbook of Software and Systems Engineering: Empirical Observations, Laws and Theories, 2003, Addison Wesley.

[14] Boehm, B., Software Engineering Economics, 1981, Prentice-Hall.

[15] Boehm, B. et al, Software Cost Estimation with COCOMO II, 2000, Prentice-Hall.

[16] Hughes, B. and Cotterell, M., Software Project Management, 2002,3rd Edition, McGraw-Hill.