GRADUATES IN THE COMPUTER INDUSTRY : A CONSULTANT'S VIEW

J. McNeil

Rapporteurs: Mr. R.A. Snowdon
Mr. J.D. Gorrie

Introduction

Although my title is "Graduates in the Computer Industry: A Consultant's View", I intend to discuss only my view of a segment of the computer industry. In a strict sense, consultancy is an activity involving the application of experience to particular organizational problems, and then making recommendations in verbal or written form. One may ask how this strict definition can possibly involve graduates. As we all know, graduates lack experience and tend not to be good at reporting. It is natural to conclude that there must be at best a very limited scope for new graduates in an organization whose activities are restricted to consultancy.

In fact, I represent an area of business which does rather more than consultancy. To my mind, no one has yet come up with a wholly satisfactory title for the kind of organization we are. The most accurate description is, perhaps, "Professional Computer Services Company". This is rather cumbersome, and probably the implication that organizations outside our particular orbit but within the computer industry are somehow not professional will be resented. With this in mind, I shall nevertheless use the term "Professional Services Company", or "PSC" for short.

I shall deal with the extent to which the PSC uses new graduates, what we look for, and what additional skills and experience we would like to find in new graduates. In order to do this, I must first describe the nature of our kind of organization. I believe that this is of particular interest because it is a new field, and such organizations grow and change very rapidly. This is done most easily by describing Logica. However, I should emphasize that, while Logica is representative of our industry, it is not typical.

An overview of Logica

We are nearly five years old, and our annual turnover, from the last financial year, was £1.2 million. Currently, we employ 250 people, of whom 200 are professional staff, and this number is rising sharply. Of these 200,

three-quarters are graduates. Not all of them entered the PSC industry, or even the computer industry itself, directly from university. As you can see, we have judged degrees to be important. Usually these are in computer science, mathematics, engineering, or the physical sciences. The often quoted example of the modern languages graduate who absolutely eats up compilers is conspicuously absent in our environment.

By general company standards, we are small, but by the standards of our industry we are quite large. Interestingly enough, many new graduates no longer think of us as a small organization. This is, I feel, unfortunate, as I consider that we still demonstrate many of the attributes of a small company which usually attract graduates.

In general, our experience with graduates has been highly successful, both for us and for many of them. For example, several of our line managers and senior consultants entered a PSC straight from university, and have never worked in any other kind of organization. It is interesting that despite this apparent isolation, our ability to advise large companies appears not to be too bad.

One can divide our work into three main areas: consultancy, implementation, and research. I use consultancy in the strict sense in which I have described it. Under implementation, I include the supervision of an implementation project by a client's own personnel, as well as our own software production. The dividing line between consultancy and research is hard to define, but research does exist in its own right in our organization. Further, in each of these three areas there are two distinct categories into which each may be divided. These are those projects where the computer is primary to the project, such as in a real-time control system, and those projects where the computer is used as a tool, such as in operations research, which is an important, distinct area of work for us.

There are some general observations which should be made about our kind of organization. The first is that consultancy and implementation naturally coexist. This is to our benefit, for one can observe that a consultant may rapidly descend into technical obsolescence unless he is regularly exposed in a direct manner to new developments. This is also of benefit to the client, for in many instances a consultancy assignment naturally leads to involvement in the development of the proposed system. We have experienced a number of cases where a feasibility study has led to a detailed study of system requirements, and that has led in turn to a partial or total responsibility for implementation, although this progression was not intended initially.

Another point is that one tends to think of implementation in this context as meaning the production of software. Certainly software production is our largest identifiable activity, but we also need and have a very strong digital engineering capability. In my own group we now have fifteen people whose prime skill is engineering; in fact, some could not write a program if you asked them. It is still possible to find software houses whose staff are wholly software oriented. Personally, I do not see how they survive without a single engineer.

It is also significant that we, and several other such companies, have deliberately, as a matter of policy, sought out difficult and advanced projects. In doing so, we become more attractive to the new recruits that we need, and create new markets, and so a need for more people - very much like an organic process. This policy seems to have worked fairly well up to now. One normally may think of a service company as dragging along behind technology, doing only routine work, but very often we find ourselves forced to extend existing technology to solve our problems.

The organization and external appearance of our company and others like it is rather unconventional. A normal manufacturing company may derive its primary external identify from three sources: a dominant personality at its head, a long history, or a product identity. The last of these, product identity, is the most important in the market place and exists in almost all cases, but we do not have it. It is hard to find an early communicated common identify between our "products" per se. Our analogue of produce identity is specialization identity; that is, in the market place companies such as ours must be seen to have certain skills. At Logica, for instance, we feel we are known particularly for our work with data communication and the systematic analysis of computer performance. Many projects require experience in applications software, in complex central software, in engineering, in telecommunications, in operational research, and so on. Thus, as an organization, we must be structurally very flexible, and staff have certain added demands made on them in order that we may realise this flexibility.

For example, it may be thought that in an identifiable area (for example, real-time systems based upon mini-computers) there will be certain processes which can be seen as structuring a production process (an analysis phase, a coding phase, testing phase, delivery to the client). These elements do not represent the important structural elements of the company. We have deliberately not established a central programming pool and indeed it is not possible to identify within the company, individuals whose sole objective is to program. All of our staff are, to a degree, analyzing. Our deliberate choice has been

not to adopt a production line flow of program production. This may appear contrary to the developments of software engineering which indicate that the process must be highly systematized. However we believe that variety in a person's tasks improves his overall performance and prevents the boredom and dissatisfaction that are now seen to be inherent in conventional production line methods. Consequently, we find ourselves adopting an individual workshop approach to the construction of programs.

Each year we take on a number of new graduates. In our first year we employed three; this year we employed fifteen and had it been possible we would like to have employed more. In addition, this year we have had four sandwich students working with us for six months. We do have certain apparent difficulties in attracting and using graduates; nevertheless, our experience has been very favourable.

It is apparent that as an industry we have failed to train our clients (as management consultants have learned to do) to appreciate the benefits of the presence of such inexperienced personnel at strategic meetings. Hence, there will occasionally be problems in using graduates even when they could, in our judgement, contribute.

This problem is aggravated in that we have only a very small amount of internally generated work under our own complete control. That is, our investment in research and development proper is small, because we deliberately avoid the production of software "packages". In a conventional product company, private in-house development can provide a controlled environment within which to nourish graduates. As the vast majority of our work is directly client-funded, new graduates must immediately begin to work with our clients. The client is soon aware of the presence of an inexperienced person.

As a result of the homogeneity of our work, we are not structured in the conventional way of research, design, development, production and marketing. This creates certain problems in constructing medium term training programmes similar to those evolved by product-based organisation. The best we have so far achieved is to take engineers and give them programming and software analysis experience.

Some projects

A brief description of some of our projects will serve the dual purpose of clarifying further how a company like Logica operates, and showing some of the ways in which new graduates have been used in real situations.

About a year ago, we commissioned quite a large real-time system for the British Gas Corporation. This is a network of twelve machines distributed around the British Isles to control the U.K. natural gas grid. This project involved the development of a considerable amount of software. There were two distinct areas where we used graduates.

The real time executive for the MODULAR 1 was developed by a two-man team. One of these two was a new graduate in computer science whose productivity was similar to that which we might expect from someone with two or three years experience. What is interesting is that on completion of this nine-month task, this man wanted to write anything but another operating system. It was not that he was not interested or that he felt that he could not contribute, but that he did not like the idea of "being in a rut". It is quite a problem for us to convince new and able graduates that such short-term expertise (what they might call "ruts") is not bad for them.

The second area of interest in this project is that of the applications programs. The graduate we employed here was an outstanding programmer who strongly opposed the use of assembler, and suggested using BCPL instead. I argued against BCPL for two reasons. I was not familiar with the language and, more important, a change to BCPL was contrary to the contract specification. Such a change was likely to reduce the faith of the client in our team. However, I proposed that this man should write out his arguments in support of BCPL. This he failed to do. This illustrates something I have often noticed. While graduates often have very firm and often correct opinions, they have neither the training nor the drive to channel such opinions into a well-reasoned case.

We are currently involved in some extensions of the Market Price Display System for the London Stock Exchange. This is an example of how a job progresses. Initially this project involved one man, an engineer, for three months. We were then asked to procure the necessary equipment and subsequently to tender for the software. Finally, the client asked if we would produce the fifty specialized video-display units he required. We are using a number of relatively new graduates on this succession of projects. For example, a man with a second degree in engineering is leading the development of the V.D.U's. A sandwich student is being used in the development of software quite satisfactorily.

We are doing also a piece of work for the Computer Board. This is to determine how best to satisfy demands for computing power given a limited amount of resources. This could well be considered a research project in which we are studying ways of quantifying demands and resources, and of establishing mappings between these. Our hope is that, with the universities, we will build interactive models of such a system. The extent to which we can use a first degree graduate on such a project is limited, although we are using an M.Sc. graduate on some of the evaluation of alternative models.

A final example involved one of our few internal projects. This was concerned with automatically determining the necessary components of a computer system. A new graduate was used, as this seemed to be a relatively straightforward task. I must admit to some ignorance on my part in that I did not explain to her precisely what I wanted. She produced a program specification rather than a functional specification. However, I subsequently discovered that she had no comprehension of what was meant by a functional specification, despite the fact that she had recently completed a three-year computing science course. Now, having worked for nine or ten months with people dealing with functional specifications, she can do them beautifully.

What we look for in a graduate

There are certain attributes which we look for in a graduate. Similarly, there are some deficiencies which we would like to be remedied.

The attributes may be classified under three headings, none of which should be surprising.

1. Relevance of Training

It is not my intention to comment upon the role of a university for vocational training. Nonetheless, we must evaluate the graduates as to their immediate worth to us. This is especially true at Logica as we do not have recourse to a significant internal training scheme. Indeed, for us, under this heading the ability to merely pump out FORTRAN quite accurately, is almost good enough as a start. What is important is that a new graduate must be able to do something of obvious benefit for the client. Provided that the graduate can be of some use, he can develop skills while contributing as a part of a team.

2. General Perception of Computing

In this I look for the role which a new graduate sees for himself in the field of computing. I also assess his appreciation of the social context of computing for he must be sensitive to the clients' possible anxieties.

3. Potential

It is important to have some grasp of how to build on an individual's experience. We are willing to place new graduates in relatively key roles because we can take for granted their ability to produce reasonably error free code which is in some sense optimized. We also take for granted some capability for analysis.

We also expect some deficiencies, and some of these have been touched on already. I shall particularly avoid the technical areas of what should form part of the curriculum. Perhaps I should discuss it, but taking a parochial view, we are currently not finding the overall nature of the technical education to be the primary problem.

It seems to be generally recognized that new graduates cannot write reports. This is probably not a new phenomenon, but it means to us that, although a man may be technically competent, he is unable to present his ideas. This deficiency has occasionally prevented a man from attaining a senior position in consultancy with us. To remedy this, I would propose that undergraduates be given some experience at arguing a case to a group of non-believers. Admittedly this is not the same as being able to present a case at graduate level, as it would have to be done in an artificial environment where they will not be forced to stand behind their proposals. However, such experience would improve their effectiveness in a business environment, where persuasion must be done by carefully reasoned and well-formed arguments for one's beliefs.

Another deficiency arises from a lack of perception regarding the use of computers. For example, new graduates all have pre-conceived ideas of the glamour areas, as well as the mundane ones. Commercial data processing, especially payroll programming, is considered by the new graduate to be unworthy of his skills. Our view is that it really makes little difference what one's first job is, as its most important role is in training the graduate to his new environment.

A related problem is the lack of perception as to the relevance of certain areas. Currently, writing compilers is glamorous, and all new graduates seem to want to do it, but they fail to realize that there is a very limited commercial demand for it.

Another problem arising from these glamour areas is that while a new graduate may be keen to tackle such a task, his knowledge of the techniques involved will in no way be balanced by his knowledge of the practical implementation problems. The accurate forecasting of the time and money necessary

to implement a piece of software, and of the time necessary to test and integrate that piece into a system, is something for which almost every graduate has no feeling, but these are the things of paramount importance to our clients.

A final deficiency I would like to point out is the impractical view most graduates have of their career progression. It is most difficult to persuade graduates to join long-term projects, that is, those lasting a number of years, and yet these are usually the most rewarding ones. Graduates should be grateful of an opportunity to experience the design and implementation of a complicated system, in the course of which many mistakes will no doubt be made from which they can learn a great deal.

Summary

In summary, let me say that we do not expect graduates to have an appreciation for our commercial organization in general, characterized by a need to work to deadlines, to generate reports for our clients, and to participate in a team effort in an organized way. However, I believe that it would be in the best interests of the computer science student if he could be exposed to some of these practical considerations he must face in industry, and I would hope that, by hearing of Logica's experiences with graduates, this view will be better appreciated.

Discussion

Professor Galler questioned what view people at Logica had of themselves in terms of accomplishment and development as a result of the lack of job classification. Mr. McNeil replied that they had, at one time, identified people as "programmers" in order to satisfy clients. However, this had led to individuals becoming preoccupied with their title and worrying as to when they became "programmer-analysts". It is the current belief at Logica that people working in a team perceive their own roles in that team and that this is a more satisfactory approach as it tends to prevent inhibition due to "pigeon-holing".

Professor Wells was interested in the observation that new graduates were unwilling to present formal proposals. He wondered whether this might be a reflection of their view that such demands indicated an attempt to avoid confronting the issues they were raising and asked whether there had been examples of graduates actually presenting formal arguments which had been recognized.

Mr. McNeil was able to cite two instances of such proposals being well argued and hence accepted. These were, however, in non-technical areas concerned with policies on expense allowances and with the organization of the career appraisal system. Technically, graduates have wide opportunities to contribute to the work of their teams. However, often the key project decisions are taken early by senior staff working together with client staff. Changing such decisions for "technical" reasons alone is often undesirable or impossible. That is a problem that many graduates find difficult to face. He felt, however, that Logica operates on a minimum of procedures and that there is no attempt to dissuade new suggestions by blocking their way with bureaucracy. If the suggestions are right then it should be possible to respond to them.