

Guidelines for Industrial Year Reports

Students registered for the BEng and MEng degrees in Software Engineering, BEng in Internet Engineering, BSc in all single honours degrees in the Computer Science department which require an industrial year, and the BSc in Computer Science with a European Language, are required to produce a report on their industrial year; the assessment of this report constitutes 50% of the assessment of their industrial year, through which it contributes to the final degree assessment.

The report should be around 5,000 words in length. You must hand in two copies to the department office by the end of the first week of the first term after the industrial year (the exact date will be notified; check the Industrial Year web page); however, we strongly advise you to submit them by post before you return. If you are studying Computer Science with a European Language, the report should be in the language you are studying.

For single honours students, 30% of the marks for the report are awarded for the structure of the report (i.e. how well you have chosen the material to include in the report and how well that material is arranged), 30% for its presentation (including spelling, grammar, typesetting and style) and structure, and 40% for the content. For language students, 70% is awarded for style, structure and linguistic accuracy, and 30% for content.

Because students' experience on their industrial years is so varied, we do not prescribe a structure for the reports. Nevertheless, most reports will include the following elements:

- a description of the organisational environment in which you worked;
- a description of the technical and application environments in which you worked;
- a description of the work you carried out;
- a critical evaluation of the experience you gained.

Organisational Environment

This is concerned with the nature of the organisation for which you worked and your place within the organisation. It might, for example, contain paragraphs like the following:

Megalomaniac Enterprises is a conglomerate operating in over 120 countries, with its head office in Liechtenstein. It is a private company owned and run by a Mr Aurwen Geldmann. Its operations range from mining to pharmaceuticals and from laundries to banks. There are 37 separate operating companies within the group.

During my industrial year I was employed by one of the banking subsidiaries, the Burma Rangoon International Banking Emporium, at its London office near the Houses of Parliament. The London office is divided into four divisions, three operational divisions (private banking, commercial banking, special services) and a common services division. The common services division is responsible for providing infrastructure (e.g. computing and telecommunications) for the rest of the office. I worked in the communications department of the common services division.

Within the communications department, there are several separate teams. I was initially assigned to the maintenance team and given particular responsibility for the local area network in the office. After the first three months I was offered the opportunity, which I accepted, to transfer to the special projects team. Although this team reported up to the head of the London office, in the normal way, it also, on occasions, received instructions direct from Mr Geldmann himself.

If you worked for a large company, this section is likely to be quite lengthy. If you worked for a small company, it may be quite short.

Technical and Application Environments

In this section you should describe the hardware and software environment within which you worked. This should be reasonably comprehensive without being too detailed, e.g.

The company's network consisted of a number of Sun Microsystems Sparc Station 10s running as file and database servers, with some 200 PC clones as clients. These machines had Intel quad core processors and ran Microsoft Windows Vista; the smallest had 512 Mbytes of memory and a 80 Gbyte hard disc. Standard application software included Microsoft Office 2007 professional (and....). Network file access was provided by PC-NFS.

All multi-user applications made use of Oracle databases on the servers. The standard Oracle utilities were used for rapid prototyping. Various different test data generators, including Testman, Debugit, and several others, were undergoing trials, with the intention of selecting one as a standard. A utility called Panvalet was used for configuration management.

... and so on.

It may well be appropriate to include some comments on your experience with this environment.

Under the application environment heading, you should describe what the computer systems you worked on were doing, e.g. general ledger, production control. You need to put in enough detail for someone who is unfamiliar with the application area to understand what is going on. In some cases, this may be a suitable place to describe the flow of information, etc., within the environment.

What you did

This should start with a clear and well-structured summary, e.g.

When I joined the company, I was sent on a four week training course to learn how to write Ruby, and how to use the various utilities that were in regular use in the department. The course also introduced me to the company's standards and procedures.

On completion of the course, I joined the five person team responsible for maintaining the purchase ledger system. The main task of the team at that time was to modify the system so that managers could make on-line enquiries about outstanding purchase orders. I was given two small programs to write and conduct unit tests on. A detailed design for each of the programs was already available. The programs were each about 200 statements long; writing and testing the two took me about eight weeks.

I was then asked to help in producing the test data for the on-line version of the system. This involved careful reading of the functional specification and I came to understand fully what the system did.

Along with a systems analyst, it took me ten weeks to produce the necessary data. The test data was reviewed by other members of the team and we took a further two weeks to incorporate the modifications requested.

...

You should then pick out and amplify, in separate subsections, those parts of your work that seem the most interesting.

Critical Evaluation

This is the place where you should say what you learned from the experience and in what respects it was good or bad. You should try to avoid being trivial. The following are the sort of comments that might appear:

I was surprised at the extent of the testing that was carried out. From my course, I knew that testing was important but I hadn't realised that this meant going systematically through the functional specification and testing every statement it made about the functions of the system. It took longer to test a module than to write it.

Given how seriously testing was taken, it was odd to find that other forms of quality management were almost non-existent. There was no formal reviewing of specifications and there were no code walk-throughs. There was no configuration management and several errors occurred because wrong versions of modules were used.

Technically, I don't think I learned a lot from my year, except how to write Ruby – an experience which I hope never to have to repeat! However, I learned a lot about the way software development teams work and what it means to have real users.

As well as discussing what you learned from a technical point of view, you should try to say something about your non-technical experiences, in such areas as teamwork and communication.

MFB

10.5.95

revised 11.9.96, 1.9.99

Revised AIS 9/10/03

Revised 7/8/09 AIS

Revised 25.1.11 MFB