1.1 Reuse

Reuse: an old dream which has always haunted information technology people.

Every other profession reuses:

- When constructing cars, motors, body work parts, and numerous other components are reused from one model to another;
- When constructing houses, prefabricated walls, standardised windows are used, as well as prefabricated skeletons.
- Electronic engineers assemble integrated circuits.

With information technology, none of this happens. With each new software project, everything is reinvented: functions to manipulate character strings, lists, stacks, and so on.

If the building industry had the same history as the software of today, to construct a house, the architect would start with designing the bricks and the cement (some would even invent, on the way, the scaffolding).

And yet, if we engaged in reuse.

1.2 Hopes for Reuse

Imagine that you have just constructed some software composed of 50% reused code!

Your productivity would have been incredible; if you had reused 50% of the code, that would mean that you had only developed 50% of the rest of the code. You would have certainly constructed the code 50% more quickly than you would if you had written everything.

The reliability of your software is exceptional. The reused code is used by other software. The greater part of its errors have therefore been corrected. Half of the software that you have just constructed is highly reliable. It can be assumed that it is at least twice as reliable than if you had to construct it all.

The maintainability of the code must be unbelievable. Half of the code has been maintained by other people. What is more, this code has been
used in very different contexts from your own. Some errors which perhaps you would only have found later, and with great difficulty, during the development of your code, have been corrected for free. What more could be asked for!

And extensibility! The code that you reused is certainly used as much by demanding people who always want more facilities than you, and wish their software always to be up-to-date. For this, they extend software components which you also use. Your software benefits from this without effort from you, you will even find new possibilities in your implementation. You are always at the forefront of progress in IT.

Choice! While designing your software, you have identified the elements which you need. Going to look in the reusable components, you will have certainly found several things that satisfy your needs. You have been able, without additional load, to test several solutions, indeed even to change a solution while it is being developed. Finally, the solutions which you have retained are the best and you can prove it.

The update of the software has been simplified: your developments have been incremental: in a first version, you have chosen basic components just to construct your application, then, during development, you have refined your choices, so as to produce a final solution. During the development, your client has been able, in an effective way, to follow the advancement of the project, and has had more choice.

Your job has changed. Before, your task consisted of assembling many lines of code. Today, you have become an architect who conceives his software by choosing and assembling components. Your profession has regained its nobility.

Errors have disappeared. Before, your applications had a very different appearance because you developed it at different times using still evolving technologies. The ways of using it were also very different. All of this has stopped. Now, thanks to reuse, all your software has the same appearance and is used in the same way, and, what is more, it is always of the highest technical standard.

Reuse is magic.

It is even, doubtless, too magical because in IT until now, without knowing why, the same things have always been reused: data structures, mathematical functions, graphic interfaces. There is still a long way to go before we can enjoy the benefits just described.

What can be done to make the dream reality?

1.3 Reuse is Object-Oriented

Reuse had always remained an impossible dream until something happened: object-oriented programming appeared (OO to the initiated)—a new magic.