

## **Reinventing the Wheel**

My business is to get people to think. The issues that I get them to think about include System Design, Test Strategies, Project Management, Database technology, OO design and (more or less) straightforward programming. The range is pretty broad and challenging. The locations of this activity are equally varied, spanning banks, small software houses, Government institutions and so on. Do I spend a lot of time saying the same thing in different words? Yes, of course. Is this reinventing the wheel on a large scale? Very much so!

## **We are all Special!**

Most establishments feel that their unique pressures cause them to respond to project management in a unique way. However, the degree of uniqueness does not seem to justify the many varieties of management for running projects. For historical reasons, especially in the IT field, companies have tended to work in isolation and to focus on their local and short-term issues. The isolation of IT departments is well known and for many years has been an issue for parent organisations. Like the poor, this problem seems always to be with us. Or is it?

As our IT industry matures we have started to recognise certain ways of working, some good and some bad. At the moment our industry appears to be entering a new phase, where ad hoc practices will be replaced by popularly accepted standard methods of working, focussed on delivery of business benefits. This is not just happening in one segment of IT. It is appearing in Analysis, Design, Development, Test, Project Management and others.

## **Patterns in IT**

The driver for this is the recognition that there are patterns of work which are generally accepted as effective and appropriate. These patterns acquire names which enter our IT vocabulary as powerful and generally understood concepts. Examples of these are appearing in Design (where the UML provides a broadly understood standard), Development (where Open Standard development tools provide a common basis) and Test (where powerful tools such as JUnit provide a standard mechanism for low-level tests). At lower levels, there are books consisting of nothing but pattern definitions, expressed either in UML or a specific implementation language.

But for Project Management, what patterns can we identify? Isn't each project uniquely different? Well, the answer of course is "No". Getting this answer accepted however, is altogether a different matter.

## Patterns for Project Management

When patterns are described in common language, they can be analysed, discussed and improved by many. When they are described in some ad-hoc language, they tend to wither, the audience being small and hence discussion and improvement limited. Often patterns are reinvented in some other (ad hoc) language. There goes our reuse! Also going out of the door is any hope of continuous improvement. The ROI is too low to justify the effort, when the audience is so small. In Project Management we need a very broadly recognised terminology and corresponding set of processes, controls and techniques which are so widely used that improvement becomes economically sensible. In the UK and much of Europe the “pattern” for Project Management is PRINCE2®.

## PRINCE2 as a Pattern

One definition of a Pattern is:

*“Each pattern describes a problem which occurs over and over again in our environment, then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice”.* In our case the “problem” is “Managing a Project”.

This definition is taken from Christopher Alexander who was actually writing about the architecture of buildings at the time. Since then the idea of Patterns has been taken up in many industries. In our industry the awareness of Patterns has underscored just how poorly we have applied reuse to our work and management practices.

If we look at PRINCE2, we see the definition of a generic approach to Project Management. It provides standard terminology, processes and (some) techniques. Like all good patterns, it is configurable. Even better, it is spoken and understood by thousands of practitioners. That means we can argue about it, refine it and eventually improve it, to the benefit of all, across many industries. Contrast that with the private and introverted approaches of many organisations to IT Project Management. Without standard PM we would expect to see islands of specialisation, misunderstandings, continued investment in dead-end solutions and little opportunity for improvement. This may just sound horribly familiar to many.

## Summary

We work in a maturing industry. We are developing effective and standard methods of working in the technical area. The sooner our standards for Project Management can become accepted the sooner we can focus on our primary tasks of delivering effective systems, to our cherished objectives of “on target, on budget”.

## **About the Author**

George Brooke is a Technical Trainer / Consultant as well as being a PRINCE2 Approved Trainer. As Head of Software Development at Siemens Nixdorf he managed solutions development for the Financial, Retail, Hotel and Energy markets for UK/European departments and customers worldwide, from concept through to delivery and roll-out. He runs a training and consultancy company based in Cambridge UK focusing on OO design with UML, programming and implementation techniques and IT project management . Contact him at George Brooke

## **References:**

### **A Pattern Language: Towns, Buildings, Construction (Centre for Environmental Structure Series)**

by Christopher Alexander, Sara Ishikawa, Murray Silverstein **ISBN:** 0195019199

### **Design Patterns**

by Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides **ISBN:** 0201633612

### **Applying UML and Patterns : An Introduction to Object-Oriented Analysis and Design and Iterative Development (3rd Edition)**

by Craig Larman **ISBN:** 0131489062

### **Patterns of Enterprise Application Architecture**

by Martin Fowler **ISBN:** 0321127420