

Mutual Consultants

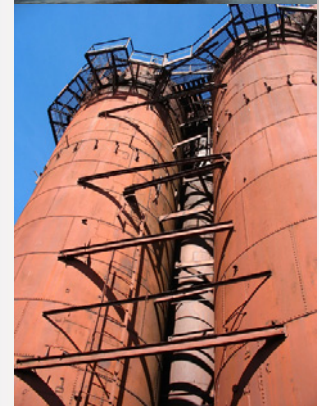


BSD0905

Bespoke Software Development

BSD

*“Development of
organisation- or function-
specific software solutions to
address particular non-standard
business needs which are designed
to improve effectiveness and efficiency
of the business”*



BSD

Overview of Bespoke Software Development (BSD)

The Need

Most organisations use 'standard' off-the-shelf packages for day-to-day needs. However, organisations increasingly require specific software to address a particular “non-standard” business need. In many instances, satisfying these particular needs can have a profound overall effect on the effectiveness and efficiency of the business.

Bespoke v Packaged Software

Originally most software applications were bespoke simply because packaged software did not exist. The arrival of the PC and Windows™ resulted in a sharp decline in bespoke software and a rapid rise in 'standard' or off-the-shelf applications. However, bespoke software is increasing in popularity for two main reasons:

- Organisations are increasingly finding that “standard” off-the-shelf packages do not provide a good “fit” or are simply not available
- Bespoke packages are now affordable because software development is often “modular” (ie applications are assembled from pre-written and fully tested modules).

Modular Software Development

Modern software development is much less costly than it used to be because there are high quality software tools available in “modular” format that can be assembled to create a full working application. Creating a bespoke application by interlinking a series of modules gives better initial functionality and very cost effective upgrades in the future.

Our Approach

No two bespoke software projects are the same (although most projects have similar development patterns). We follow a structured series of steps to ensure that the bespoke application meets your specific requirements. This is achieved by preparing and following a tailored Programme of Work for your application.

Mutual Consultants' Role

Our role is to work closely with the client both during software development and downstream to ensure that the resulting application does exactly what is required and continues to evolve in line with both the business needs and the software module capabilities.

Benefits

Bespoke software applications yield the following benefits:

- Tailored “Fit” - the application will be tailored to fit the business need exactly (ie no need to change the way you work to suit the software)
- Upgrade Path - the application can evolve in line with both the business needs and the software module capabilities
- Flexibility - Data migration from an existing system to the new application or the ability to attach a new application to an existing database (either replacing the existing front-end application or complementing what is already there).

Intentionally Blank

Bespoke Software Development

Contents

BESPOKE SOFTWARE DEVELOPMENT	1
BESPOKE v PACKAGED SOFTWARE	1
WHY PACKAGED SOFTWARE MAY NOT BE THE CHEAPEST SOLUTION	2
MODERN SOFTWARE DEVELOPMENT	3
OUR APPROACH.....	4
SOME EXAMPLES OF BESPOKE SOFTWARE APPLICATIONS	5
Maintenance Effectiveness Monitoring System ..	5
A Product Design Management Application	6
A Technical Document Management System	6
Rationalising and Balancing Software	6
A Secure Application	6
Extension to Existing System	7
A Fault Reporting System.....	7
Existing Package Enhancements	7
A Web Server System	7
Educational Software	7
Simulation Software Package	7
Existing Application Interface	7
MUTUAL CONSULTANTS' ROLE	8
Training	8
Development.....	8
Support.....	8
CONCLUSION	8
For More Information Please Contact.....	9

BESPOKE SOFTWARE DEVELOPMENT

Bespoke software is increasing in popularity for two main reasons:

- Organisations are increasingly finding that “standard” off-the-shelf packages are not a good enough “fit”, or are simply not available
- Bespoke applications are now affordable because software development is often “modular” (i.e. applications are assembled from pre-written and fully tested modules or components).

We work closely with the client both during software development and downstream to ensure that the resulting application does exactly what is required and continues to evolve in line with both the business needs and the software module capabilities.

BESPOKE v PACKAGED SOFTWARE

In the early days of computers most applications were bespoke simply because 'standard' software did not exist; these bespoke applications were often over-budget, inflexible and late. The arrival of the PC and Windows™ resulted in a sharp decline in bespoke software and a rapid rise in 'standard' or off-the-shelf application packages.

Another clear trend during the advent of the PC was for companies to develop their own in-house applications using spreadsheets or basic databases. Some of these were good solutions to specific problems (it also has to be said that some were not!). Over a period of time, however these applications tend to fall into disrepair, become dated or (very often) the in-house 'developer' moves on. In-house software development can work well but only if there is sufficient work to justify keeping the expertise within the organisation - outsourcing software development is more popular. **Cost remains the main driver which determines whether to opt for bespoke or packaged software.**

Today, organisations are increasingly finding that “standard” software packages do not always offer the best solution (they complain that the “standard” software package is “**not exactly what we want**” or “**we only use a fraction of what the software is capable of doing**”). Furthermore, bespoke software is still perceived as an expensive solution.

In summary, therefore, bespoke software used to be the norm but 'standard' off-the-shelf applications resulted in the decline of either the external or internal software developers. For specific or specialist applications organisations either write an application in-house (using a spreadsheet or a database) or manage without any software tools. **Bespoke applications are now very affordable - they can be created by assembling pre-written, fully tested modules. This means that organisations have an opportunity to improve their cost effectiveness and efficiency by having a bespoke application written to address a particular need.**

WHY PACKAGED SOFTWARE MAY NOT BE THE CHEAPEST SOLUTION

Cost remains the main driver which determines whether to opt for bespoke or packaged software. However, organisations often believe that packaged software is always the cheaper than bespoke software. This may have been the case in the past but is not necessarily so today.

By clinging to the "packaged software is cheaper" mantra, organisations often cause themselves problems in the long term. For example:

- Working practices may have to change (often making them less efficient) to accommodate the 'standard' software. **The cost of these changes and resulting inefficiencies (over the whole organisation, for the useful life of the software) are often far greater than paying for bespoke software initially**
- Where software is used to store information or data, users frequently complain that "the information is there but I cannot get it out in the format that I need". In many cases data is re-entered manually into a word processing or spreadsheet package so that the data can be printed in the required format. **In bespoke applications, the output is either in the format that the user requires or facilities are provided so that the user can customise reports to suit their individual needs**
- Many software packages come with built-in features that are not required (they were probably included to make the software appeal to a larger market). This often overwhelms users with features they will never use (or understand) and comes with an initial cost, an on-going support cost and an up-front training cost. **This can be avoided in bespoke applications.**

Other solutions that companies adopt is to develop their own in-house applications using standard software in the form of spreadsheets or databases. Some of these are good solutions to specific problems - it also has to be said that some are not! Over a period of time, however, these in-house solutions often to fall into disrepair, become dated or (very often) the in-house "developer" moves on. These home-grown solutions (particularly spreadsheets) are also prone to tinkering by the (over-) enthusiastic amateur programmer which in turn can result in errors creeping into the calculations!

In summary, **packaged software is not always the cheapest solution** - modern software development tools mean that bespoke software is often a considerably cheaper solution, especially if the true cost over the life-cycle of the software is taken into account.

MODERN SOFTWARE DEVELOPMENT

Modern software development is much less costly than it used to be because software tools are available in "modular" format that can be assembled to create a full, working application.

Creating a bespoke application by interlinking a series of modules has a number of advantages:

- In "assembling" a bespoke application, the amount of code that has to be written from scratch is small – **this saves programming time and simplifies testing and debugging**
- The modules that we use are written by specialist programmers, are often extremely sophisticated and are constantly evolving. This means that it is possible to develop cutting-edge applications today that can be readily upgraded in the future - **at a fraction of the cost of developing the same functionality from scratch**
- The focus of the software development is no longer on "getting the code to work" but on **addressing the particular functionality that the client requires.**

In summary, it is now feasible to have bespoke software written to address a particular need. This, in turn, gives organisations the opportunity to improve cost effectiveness and efficiency (where it would not have been viable in the past).

The functionality of the modular software tools is very sophisticated which means that it is possible to develop cutting-edge applications that are also extremely flexible.

In practice, this means that we are often able to attach a new application to an existing database; the new application may either replace the existing front-end application or complement what is already there.

- For example, you may already have a basic database and you wish to add more complex functionality (such as improved analysis, reporting and charting) or provide electronic links from an existing application to another application
- Alternatively, you may already have a sophisticated database with a front-end application that includes a large amount of complex functionality (which in turn requires extensive training in order to use). In practice, however, most users only require basic functionality in which case a bespoke application could be written for, say, data input or data reading that is specifically tailored to suit the user.

The flexibility afforded by the modular software tools means that existing applications can be enhanced without having to start again from scratch.

OUR APPROACH

No two bespoke software projects are the same although most projects follow similar development patterns. In any event, we maintain regular contact and seek end-user feedback throughout the development of the application.

The following are the typical steps we follow for a bespoke database application:

- **Define the application purpose and goals** - extensive dialog with the end users is essential (this continues through the project). This step is essential if the bespoke software is going to meet the client needs
- **Determine the data structure required to meet the goals and decide on the best user interface** - design the look, feel and how data is to be put in and retrieved from the application. This ensures that the application is flexible, can evolve over time and is easy to use
- **Identify the software modules to be used** - eg treeviews, grids, report generators, charting objects, printing facilities etc. This maximises the ease of use for the end-users and minimises the application development time
- **Create the physical data files and insert the software modules** - to match the data structure and to create a "skeleton" application for user review and comment. This allows end-users to have their say before it is too late
- **Write the application program code to attach the software modules to the data files** - ie creating the "full" application
- **Fully test each code module as it is written and test the application as a whole** - Involving the end users to provide feedback. This provides a further opportunity to receive end-user feedback
- **Fully document the application** - context-sensitive on-screen help and Acrobat PDF documents
- **Create a full installation and setup system on CD-ROM** - to enable users to install software unaided (and migrate existing data if required)
- **Provide user training downstream support** - as required.

In this way each bespoke software development project has a tailored Programme of Work.

SOME EXAMPLES OF BESPOKE SOFTWARE APPLICATIONS

Each bespoke application is written to provide a solution to a particular client's requirements. It is unlikely, therefore, that two clients would have exactly the same requirements needing exactly the same solutions.

The following list illustrates the range of software that we have written for clients in recent years:

Maintenance Effectiveness Monitoring System

This system allowed field engineers who were reporting corrective actions in their CMMS to:

- Quickly choose the relevant failure mode from their RCM database
- Easily answer 1 or 2 multiple-choice questions about the failure.



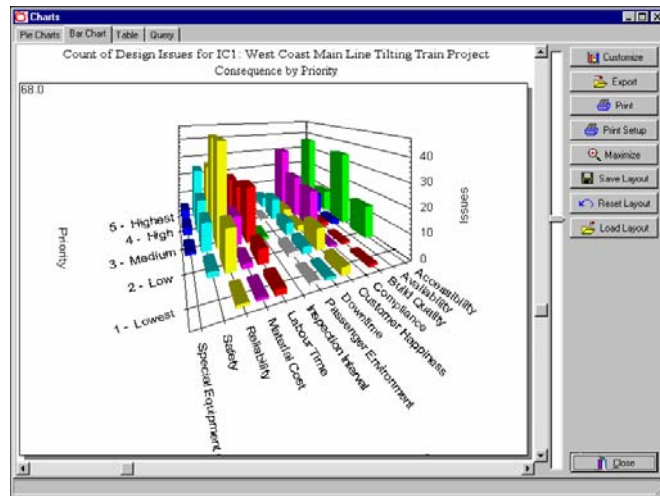
The system then assigns codes which indicate the effectiveness of the RCM decision that was made for that failure mode.

The codes can be analysed to give an accurate indication of the effectiveness of the overall RCM maintenance strategy.

RCM Decision -->		On-condition		Discard / Restoration		Failure Finding			No Task		All
Analysis Date	Vessel	Failure was Predicted	Failure was Not Predicted	Premature Failure	Task Done Too Late	Detected by Task	Detected by Chance	Multiple Failure Occurred	Failure Expected (NSM)	Not Analysed	All Failures
05 Jun 2008 16:46:49	Offshore Platform	5	3	1	1	1	1		8	4	23
05 Jun 2008 16:46:49	Offshore Platform	3	2			1			9	5	20
05 Jun 2008 16:46:49	Offshore Platform	2	3	2	4	2	1	3	4	6	27
05 Jun 2008 16:46:49	All Vessels	5	10	5	5	4	2	3	21	15	70

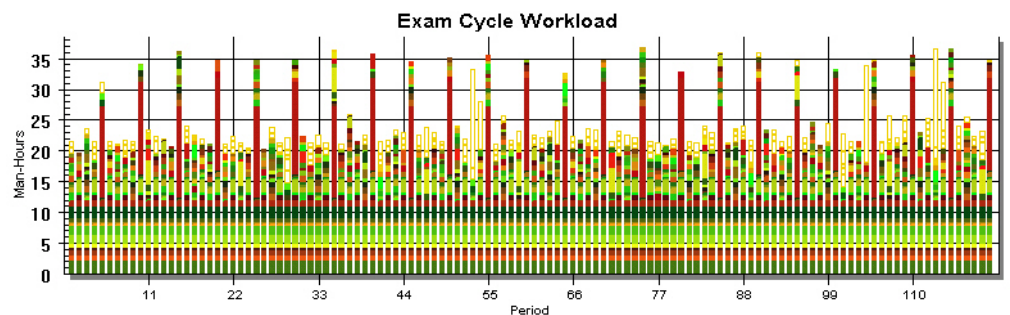
RCM Decision -->		On-condition		Discard / Restoration		Failure Finding			No Task	
Analysis Date	Vessel	Failure Prediction	All	Premature Failure	Task Done Too Late	Detected by Task	Detected by Chance	Multiple Failure Occurred	Failure Expected (NSM)	Not Analysed
05 Jun 2008 16:46:49	Offshore Platform	100 %	17 %	13 %	4 %	100 %	0 %	0 %	100 %	17 %
05 Jun 2008 16:46:49	Offshore Platform	60 %	0 %	0 %	0 %	100 %	0 %	0 %	56 %	25 %
05 Jun 2008 16:46:49	Offshore Platform	40 %	22 %	7 %	15 %	33 %	17 %	50 %	17 %	22 %
05 Jun 2008 16:46:49	All Vessels	33 %	14 %	7 %	7 %	44 %	22 %	33 %	34 %	21 %

A Product Design Management Application To ensure that the client's large volume of design issues are readily tracked and that design errors not repeated on future projects.



A Technical Document Management System To manage all aspects of the production of a client's technical documentation.

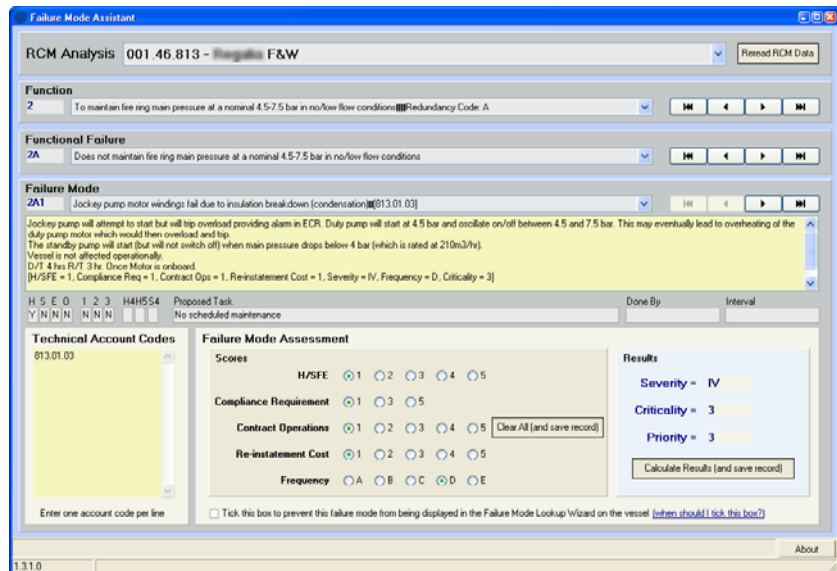
Rationalising and Balancing Software For optimising large numbers of plant and equipment maintenance tasks



A Secure Application To enable the client's suppliers to provide Life Cycle Costing data in a format for importing into the client's main LCC database

Extension to Existing System

Extension to an existing off-the-shelf application which significantly improved the end-user data entry, productivity and accuracy. Included a "wizard" application to enable end-users to look up information in a different database and automatically generate the codes required when inputting this data into a third application.



A Fault Reporting System

For collecting, structuring, analysing and reporting equipment failures

Existing Package Enhancements

To better suit the client's specific industry and provide a multi-lingual user interface

A Web Server System

To process and record customer credit card payments, issue software licence codes and handle refunds

Educational Software

(Multi-lingual) for foreign language students

Simulation Software Package

Capable of simulating almost all activities at any railway vehicle maintenance depot.

Existing Application Interface

With Lotus Notes in order to make information accessible to company users world-wide

It is very unlikely that we have already written a bespoke application that would provide an immediate solution to your specific requirements. We should, however, be happy to discuss your specific requirements with you (without any obligation) and also demonstrate some of the software that we have developed for other clients.

MUTUAL CONSULTANTS' ROLE

Our objective is to work with you at every stage of the bespoke software development process in order to provide you with an application that fits your needs precisely.

Training We aim to make all our bespoke applications very intuitive to use and fully supported with context-sensitive on-screen help and Acrobat PDF documentation. We provide downstream training as required.

Development After a period of time, many clients ask us to enhance their application in line with user requests (ie evolving the software in parallel with the business) and with the development of the software modules. When we develop an application we aim to structure the software and data so that future enhancements can be incorporated readily.

Support We provide software support for the applications we write either on an ad hoc basis or via an annual support contract.

CONCLUSION

Bespoke software is increasing in popularity because "standard" off-the-shelf packages are not what organisations want or are simply not available. Bespoke applications are very cost effective because sophisticated software development modules are readily available.

In many instances, satisfying a client's particular software needs can have a profound overall effect on the effectiveness and efficiency of the business.

By definition, bespoke software is written to address a specific need. It is very unlikely, therefore, that we have written an application that would provide an immediate solution to your specific requirements.

We should, however, be happy to discuss your specific requirements with you (without any obligation) and also demonstrate some of the software that we have developed for other clients.

*For More Information
Please Contact*

Simon Deakin or Steve Bailey

MUTUAL CONSULTANTS LTD

Eastlands Court

St Peter's Road

Rugby

Warwickshire

CV21 3QP

Telephone: +44 (0)1788 555000

Fax: +44 (0)1788 555010

Web site: <http://www.mutualconsultants.co.uk>

Email: info@mutualconsultants.co.uk

Tel: +44 (0)1788 555000
Fax: +44 (0)1788 555010
Email: info@mutualconsultants.co.uk

Mutual Consultants Ltd

**Eastlands Court
St Peter's Road
Rugby
Warwickshire
CV21 3QP
United Kingdom**

www.mutualconsultants.co.uk