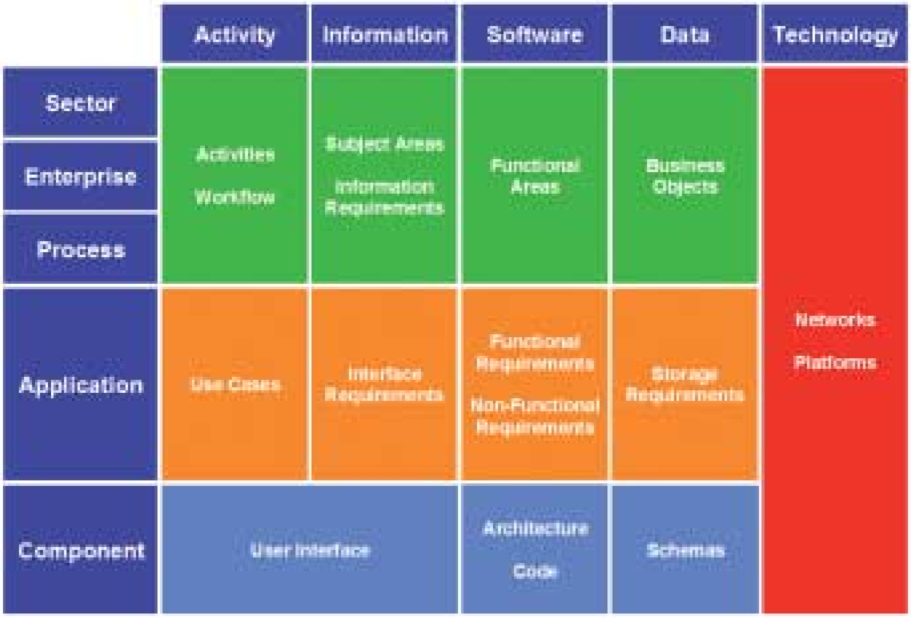
Extreme Architecture: Part 3, Applying the Framework

By Phil Robinson, Lonsdale Systems, and Floris Gout, Independent Consultant

* Part 1 was printed in Issue 19, Autumn 2004 and part 2 was printed in Issue 20, Winter 2004



	Activity	Information	Software	Data	Technology
Sector	Activities Workflow	Subject Areas Information Requirements	Functional Areas	Business Objects	Networks Platforms
Enterprise					
Process					
Application	Use Cases	Interface Requirements	Functional Requirements Non-Functional Requirements	Storage Requirements	
Component	User Interface		Architecture Code	Schemas	

Fig. 1: The eXtreme Architecture Framework

IT groups have a pressing need to identify those things worthy of their attention. Many of our clients have asked us, What do I need to know about and manage on behalf of my client?

This question led us to think about how we structured and presented our work. We wanted a framework we could use in our assignments and share with our clients.

We wanted the framework to be simple to recall, easy to use, and easy to explain. We also thought it essential that the framework should be compatible with concepts already familiar to our clients.

We explored traditional forms of architecture and urban planning, and applied that understanding to the more complex world of business and software systems. We researched frameworks developed by others.

We developed a framework that met our needs. We use this framework for our consulting assignments.

Using the eXtreme Architecture Framework, an IT Architecture can either be fully defined in single planning project, or gradually fleshed out over a period of time. We hope that the framework will provide a point of reference for business areas supported by IT management, IT project managers, developers and operations staff, frequently charged to do more with less in these lean times.

These groups may find that the framework offers them a much-needed lifeline.

Flashback

In this series of three papers, we have presented an Information Technology (IT) Architecture framework that encourages a minimalist approach to IT Architecture by exploring a number of extreme points of view.

In the first article, we used the metaphors of cathedral and shanty towns to discuss the extremes of perfection and chaos in IT systems. We implied a comparison between building and town planning, activities undertaken by humankind for several millennia, and software systems development, something that has only been performed for the last few decades.

We delved into the differences between human activity systems and software systems, and classified both types of system into a hierarchy of sub-systems. We noted that, although the hierarchy is a convenient way of classifying systems, the true nature of business and software systems is to be independent and overlapping.

This led to the notion of interoperability as one of the key architectural issues. Interoperability can be defined as:

'…the ability of a system to successfully interact with other, specified systems.'[1]

We referred to a formal definition of IT architecture found in legislation passed by the US Congress; the Information Technology Management Reform Act of 1996 also known as the Clinger-Cohen Act.

'An integrated framework for evolving or maintaining existing information technology and acquiring new information technology to achieve the agency's strategic goals and information resource management goals.'

In the second article we described the 19 architectural elements that constitute the framework. A matrix with 'system types' as rows and 'architectural views' as columns was used to organise and group the architectural elements.

The complete framework uses a single, uncluttered diagram. This approach reflects our belief that the framework is simple to describe and easy to recall. However, the diagram is not trivial – it includes nineteen different elements that, taken together fully define an IT architecture.

We then went on to describe how each of the framework elements could be described using models, assessments, potential risks or rewards, visions or desired results, strategies and principles. The completed framework is in Figure 1.

In this paper, the last in the series, we are going to present some of our ideas and observations in regard to the Framework in use. We are going to discuss:

- The grouping of various elements into areas of the framework
- Using the framework to broadly demonstrate the responsibility and disciplines required of the people in the organisation.

Grouping the Architectural Elements

In addition to what we regard as the 'standard' grouping of cells shown in the framework diagram above, cells of the matrix can be grouped in a number of other ways. Grouping cells is the main technique we use to highlight focus areas of the framework.

1.1 Rows

When we partitioned the cells of the framework into rows and attempted to name them, we were rewarded with a major insight into the framework. The names we chose tended to reflect the major disciplines associated with the different phases of the systems development life cycle:

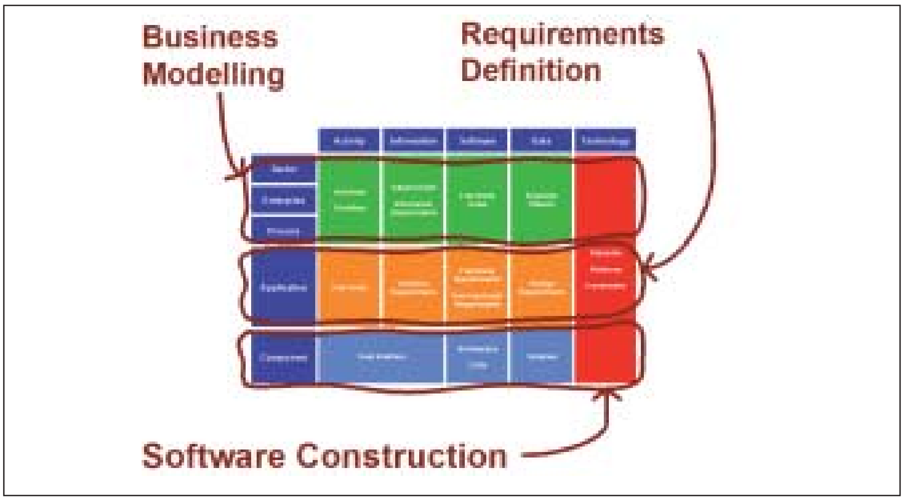


Fig. 2: Project discipline areas

The Business Modelling row describes 'what' the enterprise does and how its activities are supported by its software systems. The row includes the activities performed, workflows between activities, information requirements, business objects, a high-level grouping of software functions into a number of functional areas and the technology platforms and networks used to support business activity.

This row provides the context for the individual Applications of the enterprise.

The Requirements Definition row defines the requirements for a single Application. We expect this row to contain the Application's use cases, interface requirements, functional and non-functional requirements, data storage requirements, and an elaboration of technology requirements.

We realised that the contents of this row reflected the sections of a Requirement Definition document.

The Software Construction row describes the physical artefacts that implement a single Application. Here the elements such as User Interfaces, Software Architecture, program Code and database Schemas are the artefacts of software development, along with further elaboration of the technology infrastructure.

1.2 Columns

When we partitioned the framework into columns we were again rewarded with a major insight into our framework. The names we chose for the columns tended to reflect the management disciplines necessary to properly manage the IT architecture.

The **Process Improvement** column includes the elements that are the focus of business process re-engineering projects or continuous improvement initiatives. Activities would be used to define the scope of the improvement, while Workflows define the improved processes. Use Cases and User Interfaces describe how an Application will support the improved process.

The **Information Management** column includes the elements that need to be properly managed in order for an organisation to make effective use of information. It includes the grouping of Information Requirements into Subject Areas that are independent of business processes, and the management of all electronic and paper-based records.

The **Software Portfolio Management** column includes the elements that define an organisation's software portfolio. The portfolio is likely to a mixture of bought-in

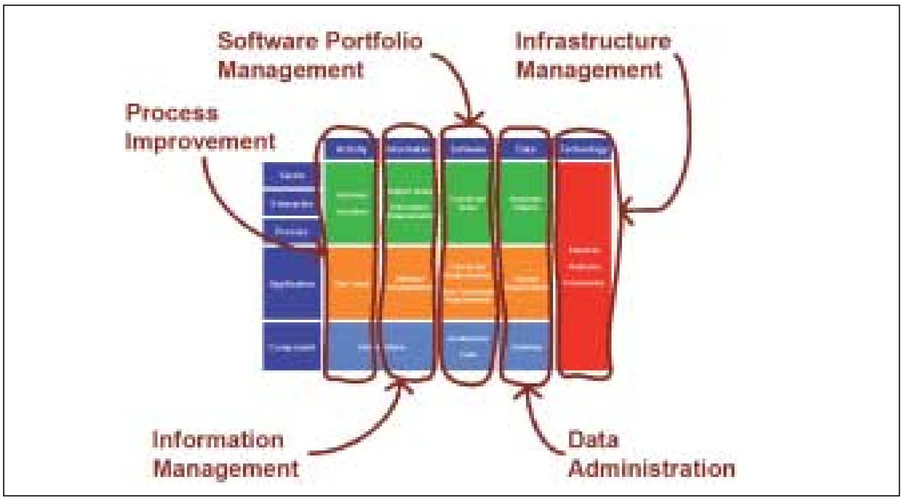


Fig. 3: Management discipline areas

packaged software and custom in-house developed software. A major concern of those who focus on this column is the integration of disparate software.

The **Data Administration** column should be very familiar to readers of this magazine. While many of us regard Data Administration as an on-going function, we should not forget that it is also the discipline that drives data quality improvement or data integration projects.

The **Infrastructure Management** column is often viewed as the 'operations' domain. The elements in this column represent the organisation's hardware and software platforms, together with the networks that interconnect them. The 'operations' group also manages the technical frameworks underlying the technology. While the discipline is mainly concerned with guaranteeing 'the smooth running' of technology infrastructure projects such as the technology conversion, standard operating environments or rationalisation of networks and technology also fall into this domain.

1.3 Staffing the management disciplines

The management discipline we know best is Data Administration, and so we shall focus on the implications of staffing this discipline. No doubt readers more familiar with other disciplines can provide their own experiences.

Over the years during which we have been involved in IT, the importance of the Data Administration discipline has waxed and waned. In the past many organisations have appointed people to data administration positions, but most would agree the outcomes have been disappointing.

Our framework seems to confirm two things:

- Management of data spans high-level business descriptions of enterprise data all the way down to the detailed data definitions contained in a database schema.
- Apart from the occasional appointment of an 'Infrastructure Manager', the Data Administration discipline has not been married with the other disciplines reflected in the columns. Managers charged with managing the software portfolios are few and far between.

We believe that we can see the result of these two observations firstly in the copious amounts of time spent developing high-level business descriptions of data, but with a disappointing lack of follow-through when it comes to implementing database Schemas.

Secondly, Data Administrators have not had the necessary partners with which to form a team focused on all aspects of IT Architecture.

We believe that the decline of Data Administration will be reversed as the need for Enterprise Architecture becomes more apparent. However, the key to success will be to ensure that IT architecture efforts are realistic and pragmatic. It will be important to avoid the tangential departure into a pipedream of an IT architecture for its own sake, that delivers very few tangible benefits to those that fund it.

1.4 Business and IT responsibilities

There is another way to look at the Framework. As we explored the management disciplines, we began to see the reasons for the frequent tension between business groups and IT groups.

It was clear to us that the business groups should be responsible for the elements in the Activity and Information columns, and IT groups should be responsible for the Data and Technology columns.

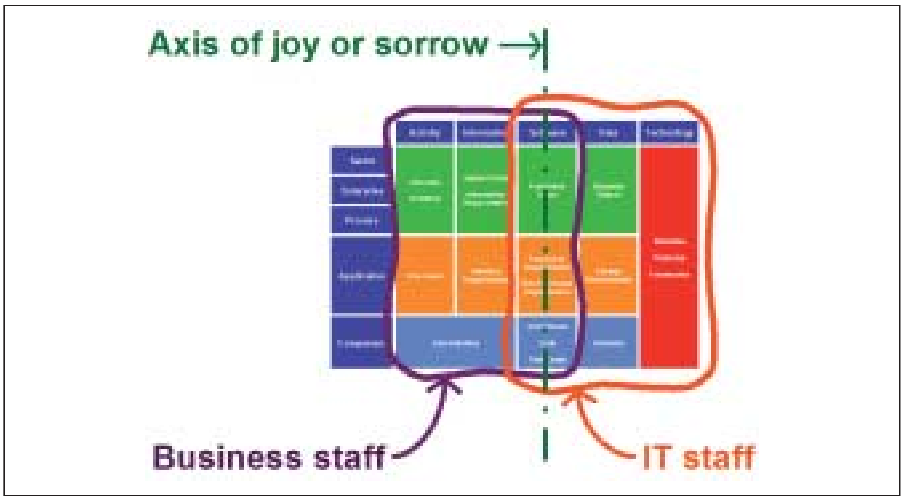


Fig. 4: Axis of Joy or Sorrow

At first we wanted to include the Software column as an IT responsibility. After all, IT staff build the software portfolio, in many cases identify the functional requirements, write the code, etc. However, there is a point of view that argues that business groups must be actively involved in the management of the software portfolio if it is to meet their needs, and be properly aligned to the goals of the enterprise.

Our conclusion is that both groups must share responsibility for the software portfolio. This represents the boundary between the two groups; it is where there are benefits to be gained from collaboration and from joint responsibility. For many enterprises, this column will come to represent either an axis of joy or an axis of sorrow.

1.5 Where rows meet columns

The cells of the matrix where life cycle disciplines and management discipline intersect also reveal areas where cooperation is required and conflict is common. For example, the intersection between the Component row and the Data column could involve interaction between the software developers and the Data Administrator. We are certain that many readers will have experienced cases where developers design a data structure that does not conform to standards. Sometimes the developers have compelling reasons to do this, but it will inevitably bring them into conflict with the Data Administrator.

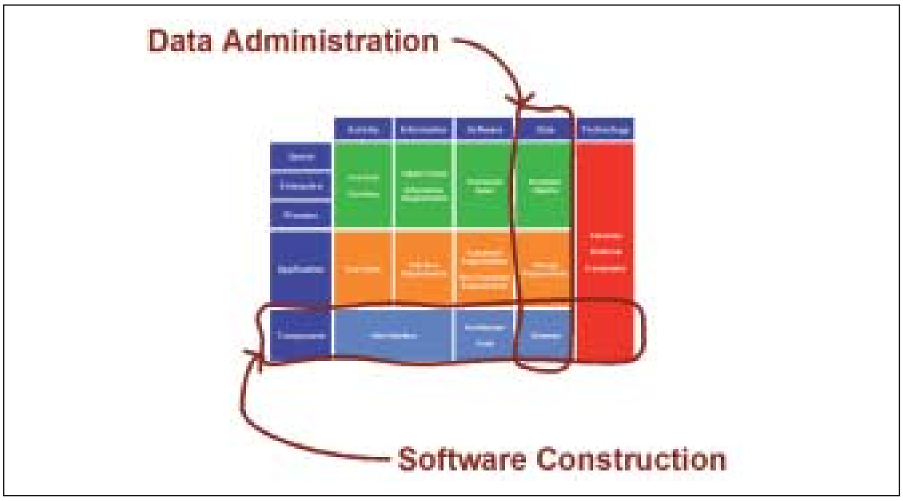


Fig. 5: Intersection of disciplines

We believe that the framework offers a way out. This framework provides a platform for both parties to present their arguments, while at the same time offering an opportunity to acknowledge the importance of the opposing point of view.

1.6 Arbitrary Areas

As well as the rows and columns described above, the cells of the matrix can be grouped into any number of arbitrary areas. This approach can be very useful for scoping areas of special interest. As an example, we will briefly discuss how we have used the framework to help us scope consultancy projects.

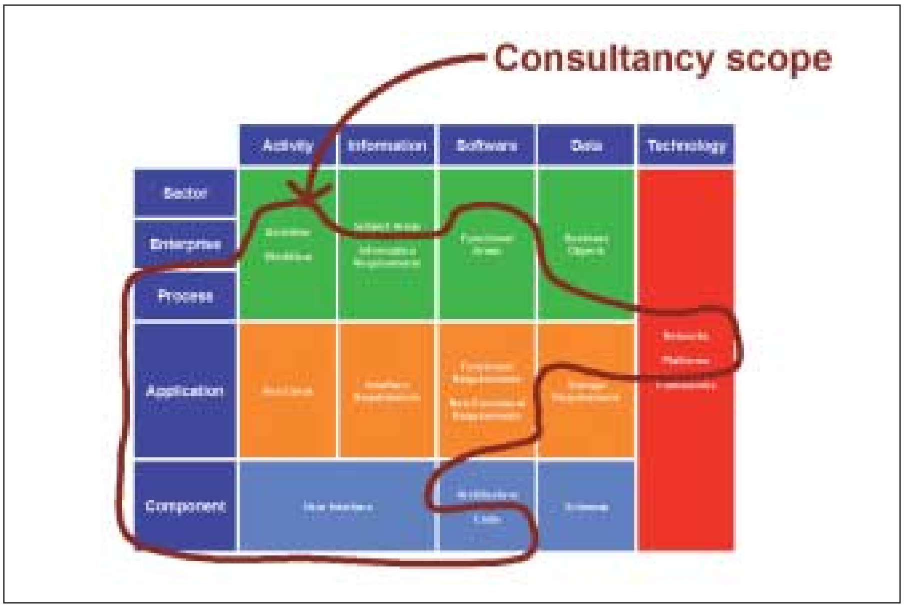


Fig 6: Consultancy scope

In one project, the framework was used to plan the development of a 'Requirements Traceability Matrix'. The client wanted to be sure that the application they had acquired fully satisfied their documented requirements, and would integrate with other applications. The basic question they wanted to answer was, 'Have we covered everything?'

To determine the scope of the assignment, we used a laminated version of our Framework reference card. After a brief explanation of the framework, we negotiated the scope of the assignment using the reference card as a mechanism for surveying the possibilities.

We physically sketched a number of possible boundaries to the assignment on the reference card. What we found was that the framework helped the client to articulate what it was that they wanted us to look at. Our final sketch became the scope of the assignment, and the basis of the project charter.

1.7 Slicing and dicing the framework

So, in summarising the various groupings of the architectural areas, we have concluded that:

- The horizontal rows describe the disciplines associated with projects, and are applied to various phases of the system development lifecycle.
- The vertical columns represent the management disciplines that are required to manage enterprise architecture.
- Groupings of vertical columns represent areas of responsibility, cooperation, or conflict.
- Arbitrary areas represent the scope of unique types of work – typically project assignments.

Conclusion

So in this article we conclude the presentation of our framework, which we use to undertake our consulting assignments. The benefits have been stated; they are simple and easy to describe, as well as unifying disciplines and easing communications between groups.

As consultants, we developed this framework in response to a question asked by an IT provider we worked with:

What do I need to know about and manage on behalf of my client?

We think that our response, in the shape of the eXtreme Architecture Framework, offers IT providers a practical tool that highlights precisely what they need to manage on behalf of their clients.

In elaborating our response, we have presented the elements of a framework, which links human activity systems with software systems. We have shown how this framework can be used to explore and classify an understanding of both types of system. We have also demonstrated how the framework can be used to focus attention on specific issues, and to clarify the roles played by various parties.

The framework helps us to visualise the various disciplines associated with growing the enterprise's architecture through application development. It also verifies the management processes associated with maintaining a healthy architecture. We looked at the intersection of responsibilities between development and managing the architecture. The Framework satisfies the existence of processes 'for evolving or maintaining existing information technology and acquiring new information technology...'

The framework is used as the basis for creating work products at each stage of the development lifecycle.

It also provides an opportunity for the provider's clients to see a clear relationship between their responsibilities and those of the provider. The framework presents a simple and consistent view to client and provider alike.

We have yet to formalise the links between elements of the architecture. Whilst our consultancies explored links vertically and to some degree horizontally, we have not yet achieved a formal but simple method to do so – a method that is not onerous for IT groups. We anticipate that this will occur as our consultancy experience widens. This is an interesting area because it would support audits and traceability. The impact on testing régimes should be lessened, but we have not experienced any empirical evidence of this yet.

In conclusion, we would like to remind the reader of the image of the suburban house nestling between Rheims cathedral and a shanty town. We would like to think that the framework invites participants to be extreme by avoiding extremes! We hope the result is vision, knowledge, calm realisation, and enlightenment for all involved.

About the Authors

Floris Gout gained his Bachelor of Applied Science (Information Science) at Edith Cowan University in Perth, Australia. Whilst studying at ECU he worked at the University of Western Australia, building research databases for epidemiological studies. Floris was then employed at the Department of Justice from 1991 till 1999. He became Data Administrator, and was Project Manager for its first data warehouse. Floris has been an independent contractor since 1999 and he is still enjoying new and creative challenges. He can be contacted at floris@floris.com.au

Phil Robinson has been involved in the planning, analysis and implementation of a diverse range of business, scientific and technical information systems. Phil is an experienced workshop facilitator and has led numerous workshops in the course of his consulting assignments. Phil has presented training courses for organisations in Australia, Thailand, Hong Kong, Singapore and Indonesia. As well as presenting courses, Phil has authored numerous courses for industry and three University units. He has also had two books published on programming Apple computers. The books were published in a number of countries including the USA, UK, and translated into German and French. More recently, he co-authored two award-winning articles describing an original organisational theory. He can be contacted Lonsdale@iinet.net.au

More material can be found at www.extremearchitecture.org

References

[1] The Open Group Architecture Framework (TOGAF), Version 8, 'Enterprise Edition', 2002, available at http://www.opengroup.org/products/publications/catalog/i912.htm