

## Association for Information Systems AIS Electronic Library (AISeL)

---

ACIS 2004 Proceedings

Australasian (ACIS)

---

December 2004

# How Do Experienced Architects Use Architecture Development Methods?

Rob Barrow  
*RMIT University*

Keith Frampton  
*RMIT University*

Margaret Hamilton  
*RMIT University*

Bruce Crossman  
*IBM Australia*

Follow this and additional works at: <http://aisel.aisnet.org/acis2004>

---

### Recommended Citation

Barrow, Rob; Frampton, Keith; Hamilton, Margaret; and Crossman, Bruce, "How Do Experienced Architects Use Architecture Development Methods?" (2004). *ACIS 2004 Proceedings*. 58.  
<http://aisel.aisnet.org/acis2004/58>

This material is brought to you by the Australasian (ACIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in ACIS 2004 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact [elibrary@aisnet.org](mailto:elibrary@aisnet.org).

# How Do Experienced Architects Use Architecture Development Methods?

Rob Barrow<sup>a</sup>  
Keith Frampton<sup>a</sup>  
Dr Margaret Hamilton<sup>a</sup>

Bruce Crossman<sup>b</sup>

<sup>a</sup>School of Computer Science & Information technology  
RMIT University  
Melbourne, Australia  
rbarrow, keithf, mh @ cs.rmit.edu.au

<sup>b</sup>IBM Australia

## Abstract

*Software architecture methods play a central role in the development of large enterprise computer systems. However, the extent to which individual experienced IT architects employ a software architecture method is largely unknown. In this paper, we surveyed a group of experienced architects, and set up an “in practice” field study of some of these architects to explore the way they applied a well documented software architecture development method. Among the findings to emerge are that architects modify their method more regularly than previously recognised, and that tools for visual communication are the most commonly used tools by these architects.*

**Keywords:** Software architecture, methods, analysis

## INTRODUCTION

Software architecture has various definitions, which range from designing, creating and maintaining all the code of large computer systems to incorporating all the functioning of the physical and logical components of enterprise systems as well. For instance, the ANSI & IEEE (2000) define software architecture as “The fundamental organization of a system embodied in its components, their relationships to each other and to the environment and the principles guiding the design and evolution.”, while Bass et al. (2003) define it as: “... the structure or structures of the system, which comprise software elements, the externally visible properties of those elements, and the relationships among them.”

The creation and application of a well-documented software architecture development method is attractive to organisations developing large-scale systems because it offers them an opportunity to capture the experience and knowledge of skilled practitioners, which collectively form part of the intellectual capital of the organisation. The method provides a structured, repeatable mechanism for the development of software architectures and it provides a common language with which information and ideas can be exchanged between practitioners. For the employer, a good method can allow for a more unified, complete and reliable system. Tried and true methods, which have worked in the past to deliver the goods, are seen as providing a more easily maintainable system, and the documentation is there to explain the need for this particular device or input.

One of the most important perceived benefits to the organization which adopts a software architecture method, is that the risk of project failure and associated financial and organisational impacts will be minimized. If all architects follow the same method, they should theoretically produce the same product.

The research study reported in this paper is concerned with determining how faithfully software architects apply a particular architecture development method and with ascertaining the extent to which such methods require further input from software architects in order to produce a complete and viable solution. We first describe additional background to this research and comment on related research and literature, while the section entitled “Description of the IBM Custom Architecture Method” describes the concept of an architecture method, and in particular the relevant method developed by IBM. In the section entitled “Research Approach”, we explain the design of a questionnaire and workshop developed to elicit information from experienced architects. The results

and analysis are presented in the next section whilst conclusions and future work are outlined in the final section.

## **BACKGROUND**

An examination of the literature reveals that while some work has been done in assessing how well methods are applied in the domain of software development generally, and requirements engineering in particular, virtually no work has been done to examine methods with respect to the creation of software architectures. Fitzgerald's work (1997) for instance, considered software development methods.

Fitzgerald contended that the adoption of formalised software development methods would improve the process and product of systems development. He concluded that practitioners will not adopt formalised methods in their prescribed form, but that they may modify and omit aspects of methodologies in a very pragmatic and knowledgeable fashion. Other researchers have reached differing conclusions. Nguyen et al (2002) found that when examining the requirements engineering (RE) process, that "...the creativity of the RE process is hampered by strict adherence to engineering and science methodologies."

This paper reports on an investigation of the application of a specific commercial software architecture development method, the IBM Custom Application Development method, in practice. We prepared a questionnaire in which we sought to determine which tools and techniques software architects use in developing an architectural solution to a problem. We also organised a workshop where architects were asked to apply this method to a task. We documented our observations and analysis of the application of the method to the task with a particular interest in ascertaining where practitioners deviated from the method, the nature of the deviation (omission of method elements, modification of method elements or adding to the method) and the stated rationale for their non-adherence to the method. This work differs from that of Nguyen and Fitzgerald then in that it examines the development of software architectures rather than requirements, and uses a combination of ad-hoc and post-hoc techniques.

In considering the types of modifications that can be made to a methodology, we made the following distinctions and definitions:

Tailoring – a general approach to modifying a methodology across one or more organisations. An example of this would be the IBM Custom Application Architecture Method, which attempts to provide IBM staff with a tailored methodology for the development of architectures by IBM

Customising – modifying a methodology for a specific project for a specific customer. An example of this would be the specific instance of the IBM Custom Application Method produced for a project following the Method Adoption Workshop (one of the first tasks undertaken in the Method).

Adapting – the dynamic, ad-hoc modification to a customised methodology undertaken by practitioners as they apply the method to the specific project.

Our contention is that experienced software architects will engage in adaption of an already tailored and customised methodology on a frequent basis. This would be consistent with the findings of Carroll and Swatman (1999), who in their study of requirements engineering, found that "...methodologies need to be selected for the contingencies of any particular project....but also how, having selected a methodology, this choice may have to be adapted for the contingencies of the situation as it unfolds".

The work of Fitzgerald (1997), reached a similar conclusion, however, the study we have conducted differs from that of Fitzgerald's in at least three respects:

- Fitzgerald's study examined software development methodologies as opposed to software architecture development methodologies;
- We observed architects while they were using the method as well as asking them questions after they had used the method, and
- Fitzgerald's focus seems to have been on what we describe as tailoring rather than customising, whereas our study is focused on what we have described as adaption.

## **DESCRIPTION OF THE IBM CUSTOM ARCHITECTURE METHOD**

The IBM Global Services Method, (Galic et al. 2003, pp 18-26), commonly referred to simply as the Method, is used for developing and delivering client business solutions across all aspects of IBM Global Services business throughout the world. The promise to customers is that the technical and industrial experience of the global community of IBM practitioners will minimise project risk and expedite project delivery time based on more predictable results derived from previous projects.

The Method structure provides detailed definition and guidance for descriptions of standard deliverable components - tangible project artefacts - including clear input and output work product requirements for each phase. "Engagement Families" are defined as groupings of possible engagements or connections within a common area of customer needs such as for e-business or learning services or security and privacy. Every engagement family comprises a number of engagement models, which are specified in terms of a work breakdown structure, which illustrates the phases, activities within every phase, tasks and subtasks. Every task/subtask is defined further in terms of the roles required for implementing the task, and what work products are required as inputs to the task and what work products are outputs from the task, either new work products or updated input work products.

The Method includes both project management and technical methods. The technical methods consist of plan templates (paths) and techniques to support a wide range of projects. At the base of The Method is a core set of 283 Work Product Descriptions, which are tangible, reusable artefacts produced as a result of one or more tasks performed on an engagement. The Work Products contained in the Method cover a broad range of work performed on engagements and are logically categorized into 6 high level domains: engagement, business, organisation, application, architecture and operations.

In addition to the Work Product Descriptions, the Method also provides guidance on how engagements should be conducted. This guidance is delivered through 84 Engagement Models that represent many of the typical projects conducted in IBM Global Services. The Method contains engagement models in familiar areas such as Custom Application Development and Package Integration, as well as new engagement models in key areas such as e-business, IT Management, Solutions Consulting Integration, Enterprise Architecture, and Business Intelligence.

## **RESEARCH APPROACH**

Regardless of how well trained, educated or experienced the architects are, or how well defined the method or procedures are, the extent to which individual certified architects employ a particular software methodology has not been published in the literature. The aim of this research is to try to understand the extent to which experienced architects adhere to a well-defined method and to understand the architecting process better. Specifically, we were trying to ascertain the level of adaption which experienced architects apply to a well-defined method.

There were two elements to the work undertaken:

1. The design, conduct, and analysis of a survey of experienced IBM architects and how they used the IBM Custom Application Development method; and
2. The design, implementation, and analysis of workshops in which experienced IBM architects were asked to apply the IBM Custom Application Development method to a case study.

### **Design of questionnaire**

A questionnaire was constructed, using approaches described by Neuman (2003, pp 263-307), to determine which tools, methods, and defined work products the architects use. In addition, the questions on tools and techniques were only semi-structured because there was no prior research indicating what a viable list of alternatives would be appropriate. The structure and framing of the questionnaire was based on interviews with three experienced IBM practitioners with extensive usage of the method in real customer projects, two of the authors' extensive commercial experience (over 50 years combined), and from detailed reviews of the method phases, activities, tasks, work products and deliverables.

The questionnaire was designed to take less than 10 minutes for a complete response, and for straightforward distribution and usage through common email clients.

The questionnaire had four major sections. These were:

1. An introduction and content section, including instructions for completion and contact information for questions and/or concerns about the questionnaire and the overall research,
2. Two semi-structured questions asking the respondents what tools and techniques they used to supplement the method in actual practice, (definitions of what was a tool and what was a techniques were provided as part of the questionnaire),
3. Two structured questions, asking the respondents to rate their frequency of use and modification of all the work products described in the Method, (these questions reflected the on the IBM method), and,
4. Some basic career related biographical information, including experience within both the IT Industry and IBM.

The questionnaire was pre-tested with senior IBM IT architects and after refinement it was e-mailed to a significant proportion of the members of the IBM IT architect profession in Australia and New Zealand. This choice of sample was deliberate as the intention was to determine how appropriately qualified users of the method supplement and use the various work products defined within the macro design phase of the method and was most appropriate for this research as only trained IBM IT architects have access to this method.

### **Design of workshop**

A workshop was designed to determine how closely software architects followed the IBM Custom Application Development method, the ways in which they deviated from it and the reasons why. The workshop was structured in three phases:

- 1) The overall research context was presented and the results of the survey discussed,
- 2) The participants then applied the method to solving a problem posed in a case study, and
- 3) The resultant solutions were presented and the participants discussed their usage, or not, of the method, and other personal conclusions and observations resulting from the workshop.

A pilot was conducted with two experienced software architects to refine the case study and the experimental procedure.

Participants in the workshop were highly experienced IBM Architects, with a combined total of more than 70 years of experience between them. Over the course of the workshop, participants, working in a small team, applied the method to the case study. The dialogue of the team was audio recorded and in addition, a scribe was present to record observations and to capture additional contextual information not available from a tape. Further, as the architects developed diagrams and other artefacts on a whiteboard, digital images of these were captured. The workshop was conducted on IBM premises and with usual IBM working conditions and facilities in order to reduce any possible negative research implications caused by unfamiliarity or discomfort for the participants.

The size and complexity of the method mitigated against a case study in which all phases of the method were applied and instead, attention was focussed on a single phase, the Macro Design phase. The Macro Design phase is the third of seven phases in the method. The two preceding phases are concerned with project commencement and with providing the customer with an understanding of the costs, schedule and risk information to make an informed investment decision regarding a potential new system. The Macro Design phase however, is concerned with the development of a “robust architectural framework upon which to build agile releases.”

## **RESULTS & ANALYSIS**

### **Analysis of questionnaire responses**

The survey was emailed to fifty experienced IBM IT architects in Australia and New Zealand, and eighteen responded, giving a thirty-six percent response rate.

For question one, the architects were asked to identify the tools they most often use. The most commonly used tool was Visio (14/18), followed by Word (8/18), PowerPoint and Rational Rose (6/18), Excel (2/18) and all other tools each with (1/18).

The second question asked which techniques they used while performing as architects. The techniques most commonly used were identified as brainstorming, scenarios, proof of concept, interviews, workshops and peer review sessions.

### Use of work products

In order to rank the usage of work products, the architects were asked to identify which products they rarely, sometimes, often or always use. A score was allocated to each product calculated by assigning a 4 for always, 3 for often, 2 for sometimes and 1 for rarely and totalling over all the responses. <sup>1</sup>

<b>Label</b>	<b>Input Work Product Name</b>	<b>Score</b>
A	Architectural Template	29
B	Component Model	56
C	Architectural Decision	57
D	Architecture Overview Diagram	66
E	Change Cases	31
F	Non-Functional Requirements	61
G	Reference Architecture Fit-Gap Analysis	40
H	Standards	55
I	Technical Prototype	36
J	Viability Assessment	41
K	Current IT Environment	63
L	Deployment Unit	32
M	Operational Model	60
N	Service Level Characteristic Analysis	33
O	Software Distribution Plan	23
P	Parametric Costs	18
Q	Performance Model	33

Table 1 Input Work Products

---

<sup>1</sup> The ordering of the work products in these tables matches the ordering of the work products in the Macro Design phase of the method.

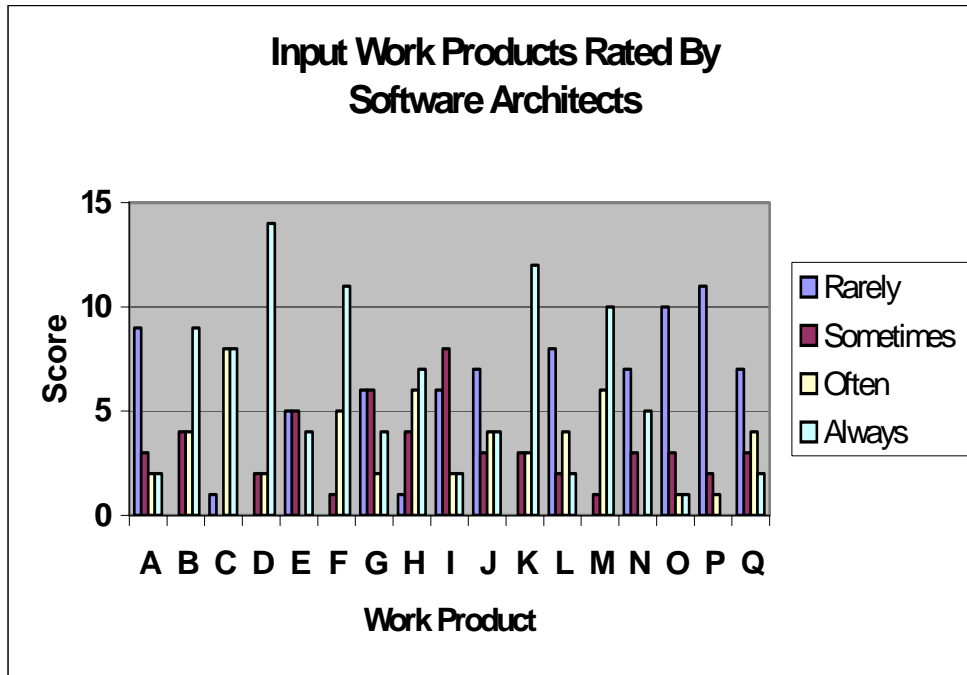


Figure 1 Input Work Products as used by Software Architects

The responses from the Software Architects about which work products they rarely, sometimes, often or always use are plotted in Figure 1. This data indicates that for input work products, the Architecture Overview Diagram, Non-Functional Requirements product and Current IT Environment artefacts are almost always used, while the Parametric Costs product is the least commonly used or modified, closely followed by the Software Distribution Plan and the Architectural Template.

However for the Output Work Products there were clearly five work products which were more heavily used or modified, as can be seen in the table below and in Figure 2.

Label	Output Work Product Name	Score
A	Architectural Template	29
B	Component Model	62
C	Architectural Decision	65
D	Architecture Overview Diagram	66
E	Non-Functional Requirements	62
F	Reference Architecture Fit-Gap Analysis	33
G	Technical Prototype	36
H	Viability Assessment	37
I	Deployment Unit	41
J	Operational Model	62
K	Service Level Characteristic Analysis	37
L	Software Distribution Plan	26
M	Parametric Costs	23
N	Performance Model	35
O	Technical Transaction Map	27

Table 2 Output Work Products

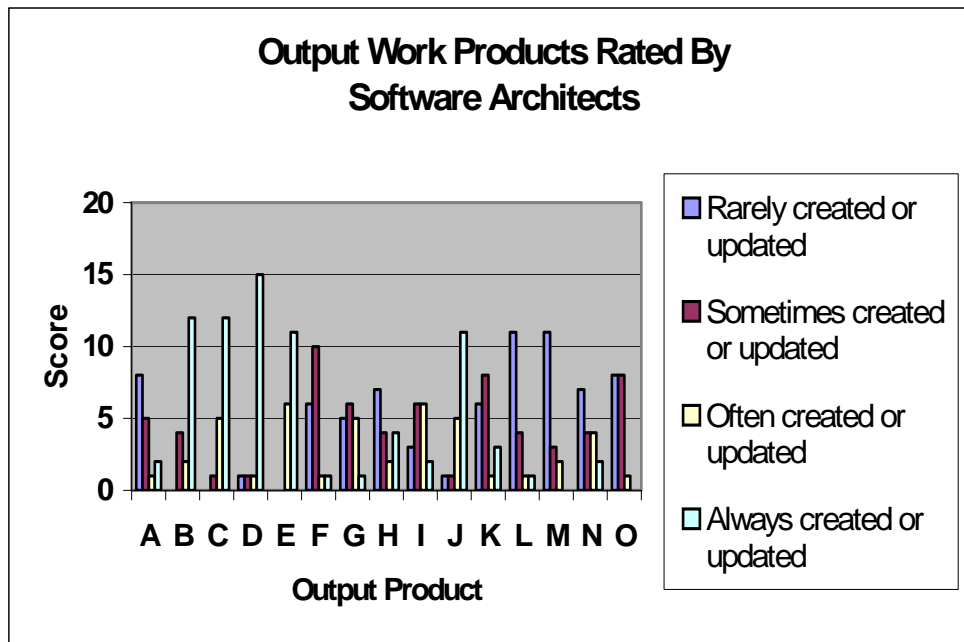


Figure 2 Output Work Products as used by Software Architects

Of significance is the fact that of the five most commonly used Output Work Products, three (Architectural Decision, Architecture Overview Diagram and Non-Functional Requirements) are typically used to capture and communicate understanding of the problem and solution. The Software Distribution Plan, Parametric Costs and Architectural Template are the least used or modified artefacts, both for the output and input work products.

### Analysis of workshop

The results of the workshop support the contention that experienced software architects do not rigidly apply a software architecture development method, but instead make frequent use of their experience to adapt the method. This is consistent with the conclusions drawn by Nguyen et al (2002) in their study of the application of requirements engineering methods. Further, we observed that architects apply a method differently if they are fulfilling the role of a team leader compared to the way in which they apply it when they take on the role of team member.

Whilst no specific software tools were used in preparing the architectural solution for the case study, a very heavy reliance was placed on the use of diagrams to communicate understanding between the architects and to represent the solution.

As described previously, the architects were given a case study and asked to generate a solution by applying the Macro Design Phase of the Method.

In watching them work through this case study, and asking questions of them afterwards, we noted that the architects did indeed work methodically to a plan. The team applied a dynamically generated, that is, adapted, version of the method in which only the necessary method elements were used.

What was observed and further affirmed in the extensive debriefing that followed the workshop was that the team knew, as a consequence of their experience, which “activities” were relevant for a problem of the scope posed in the case study. As a result, many of the specified “activities” and their associated “tasks” were not performed. In addition, even for those “activities” which were undertaken, the application of the method was not rigid, with the consequence that many “tasks” determined as unnecessary through experience by the architects were not performed.

In addition, there were several instances when a member of the team suggested they should check certain aspects with the customer before further development.



Among the key techniques which were used during the workshop were:

- Questioning – there was much use of questioning techniques by the team. They undertook questioning of the “stakeholder” and also questioned documented assumptions in the case study.
- Application of assumptions – there were numerous instances of the team member to whom the other members deferred as “leader”, challenging assumptions as the group attempted to refine their understanding of the problem.
- Drawing on previous experience – there were many instances of members of the team referring to “previously applied patterns” and also using and referring to information held by IBM on similar projects.

Finally, we observed in the workshop, that there are certain team roles, specific architect experience, and the leader of the team drove the adaptation and application of the method.

## **CONSTRAINTS ON THE RESEARCH**

One of the constraints that we have identified in using a workshop, as one of the principal research tools, was the potential that being observed may cause the participating architects to work in a manner different from that which they ordinarily use.

This phenomenon has been the focus of considerable study in the field of behavioural psychology. One definition of it, found in Draper (2004) is “An experimental effect in the direction expected but not for the reason expected; i.e. a significant positive effect that turns out to have no causal basis in the theoretical motivation for the intervention, but is apparently due to the effect on the participants of knowing themselves to be studied in connection with the outcomes measured.”.

In the context of this study, we accept then that the way in which the participants worked may be different to the way in which they usually work and that this difference may be due to knowing that they are being observed, however, no specific analysis of the implications of this potential external influence has been attempted.

## **CONCLUSIONS AND FUTURE WORK**

As a result of this research, we have identified that architects modify their methodology in three ways, tailoring, customising, and adapting, to suit the problem, the team and the customer. In future work, it would be interesting to see if this is related to their level of experience as an architect as well.

Architects rely very heavily on visual tools to communicate their understanding of the problem and their representation of the solution to each other and to the customer. Further research exploring why and how the architects use these types of tools is planned. It is also our intention to conduct further workshops to investigate whether cultural and personal differences between software architects affect their application of a method. Since IBM is a global Business, it would also be interesting to investigate whether our findings apply internationally.

The heavy emphasis on communication and collaboration within the team has particular significance for the teaching and certifying of software architects and for commercial organisations looking to employ software architects.

## **REFERENCES**

ANSI/IEEE 2000, *Std 1471-2000 Recommended Practice for Architectural Description of Software-Intensive Systems*, IEEE.

Bass, L, Clements, P & Kazman, R 2003, *Software architecture in practice*, 2nd edn, SEI series in software engineering, Addison-Wesley, Boston.

Carroll, JM & Swatman, P 1999, 'Managing the RE Process: Lessons from Commercial Practice', in *Proceedings of the Fifth International Workshop on Requirements Engineering: Foundations of Software Quality*, pp. 3-17.

Draper, S 2004, *The Hawthorne Effect*, viewed 15 October 2004, <<http://www.psy.gla.ac.uk/~steve/hawth.html>>.

Fitzgerald, B 1997, 'The Use of Systems Development Methodologies in Practice: A Field Study', *Information Systems Journal*, vol. 7, no. 3, pp. 210-2.

Galic, M, Adams, J, Bell, JA, Disney, R, Kanerva, V-M, Matulevich, S, Rebman, K & Spaas, P 2003, *Patterns: Applying Pattern Approaches (SG24-6805-0)*, Patterns for e-business Series, IBM.

Nguyen, L, Armarego, J & Swatman, P 2002, *Understanding Requirements Engineering: a Challenge for Practice and Education*, Deakin University School of Management Information Systems, Melbourne.

## **ACKNOWLEDGEMENTS**

The authors wish to acknowledge the cooperation and assistance of IBM in this study. In particular, the contribution and participation of the experienced IBM architects in the survey and workshops was invaluable. In addition, the readability of the paper was substantially improved by a review done by the partner of one of the authors.

## **Copyright**

Barrow, Frampton, Hamilton, Crossman © 2004. The authors assign to ACIS and educational and non-profit institutions a non-exclusive licence to use this document for personal use and in courses of instruction provided that the article is used in full and this copyright statement is reproduced. The authors also grant a non-exclusive licence to ACIS to publish this document in full in the Conference Papers and Proceedings. Those documents may be published on the World Wide Web, CD-ROM, in printed form, and on mirror sites on the World Wide Web. Any other usage is prohibited without the express permission of the authors.