



Methodologies for Developing Multimedia Systems: A Survey

Mark Szota & Kirsten Ellis

Monash University, Berwick Campus, Clyde Road, Berwick, Victoria 3806, Australia

{Telephone: +61 3 9904 7097, Fax: +61 3 9904 7169, Fax: +61 3 9904 7169, mark.szota@infotech.monash.edu.au}{Telephone: +61 3 9904 7132, Fax: +61 3 9904 7169, kirsten.ellis@infotech.monash.edu.au}

ABSTRACT

Many theoretical development practices exist for creating multimedia systems. Most of these development models are orientated towards building traditional information systems, where the requirements are usually well understood. Multimedia systems, like the industry itself, are evolving rapidly, therefore new tools and techniques are constantly being published. Such rapid change results in clients of such systems lacking full awareness of the capabilities of multimedia systems, thus making it difficult to define their requirements. Some of the current models are able to respond to scenarios such as this, but others cannot. This research surveyed multimedia developers within Australia in order to find the most widely used development model(s) within the industry, and the rationale for their use. The results indicate that there is no specific approach to creating multimedia systems. Developers tend to use a range of different methodologies. The motives for using a variety of approaches are also examined.

INTRODUCTION

The development of any computer software system has a common characteristic: a software development life cycle. This life cycle is a period of time that commences at the proposal to develop a system, and usually terminates when the system is complete and handed over to the client. It involves analysing the requirements of the system; designing and implementing a solution; testing and installation. In many cases, an ongoing support stage occurs, where the system may be modified or updated should the need arise. These stages may be performed repeatedly, or overlap with each other.

Developers of traditional Information Systems (IS) use a range of software engineering methodologies. With the progressive escalation of multimedia applications, it remains unclear if developers are applying these same methodologies. This paper investigates the most widely used methodologies to develop multimedia systems, and the reasons for their adoption.

DEVELOPMENT MODELS IN VOGUE

Current methodologies used for traditional system development can be categorised into three different forms: structured, iterative and evolutionary. Structured models (also known as traditional models) are in a linear form. Phases are completed sequentially until the cycle has finished. Iterative models are based upon a cycle that is repeated until it is complete. Some of these models combine iterative and sequential stages. Evolutionary models are new hybrids that generally do not fit into either category. They may be structured, iterative, or a combination of both.

Structured methodologies are one of the most commonly used software development techniques. A primary example of this is the Waterfall model. Several variants of this approach were uncovered, which suggests that there may be no fixed approach. Dennis and Wixom (2000), along

with Satzinger et al. (2004) suggest a linear, four-phased approach that involves pre-project planning, analysing, designing and then implementing a system. Pressman (2000) offers a similar interpretation with minor variations. The planning stage is removed, and a dedicated testing phase is inserted after the system has been built. Hoffer, George and Valchich (1999) propose a version that has extended planning and design phases. However, their model suggests a project can go back to a previous stage if required, which tends to violate the linear nature of the Waterfall approach. Vaughan (1998) promotes his own structured approach, which is oriented towards multimedia development. Each stage consists of several defined tasks, and the project can be aborted at any stage. Vaughan also supports the use of prototypes and Computer Aided Software Engineering (CASE) tools. Siegel (1997) presents a structured model that is focused upon web site development. Like traditional methodologies, it does not allow any backtracking, and has a similar four-phase approach.

Iterative methods use a highly repetitive cycle of development. The most common forms of iterative development are Rapid Application Development (RAD) and Prototyping. Dennis and Wixom (2000) present several versions of RAD. The key feature of RAD is the use of CASE tools, which are used to hasten the analysis, design and implementation phases. Joint Application Design (JAD) sessions can also be used to assist in the analysis of system requirements. Plfeeger (1998) describes a Phased Development model that breaks the overall system into a series of 'versions' that are developed sequentially. This process is repeated until the system is complete or becomes redundant. Prototyping uses a similar repetitive formula, but the same example is continually enhanced until it is complete (Dennis and Wixom, 2000). Throwaway prototyping uses prototypes that are designed to explore and understand a particular aspect of the proposed system. Once these issues have been resolved, the prototype is 'thrown away', and a linear progression of system development continues.

Several Evolutionary models of development exist. MacCormack (2001) presents an Evolutionary-Delivery model of software development that involves breaking down a project into several micro-projects. The objective of each micro-project is to deliver a portion of the functionality into the overall system. This method provides early feedback on how the development is progressing, and is highly flexible. Pressman (2000) presents an Incremental model that combines structured development with the iterative processes of prototyping. Sommerville (2001) discusses the Spiral model, which combines the iterative processes of Prototyping with the clinical and systematic approaches of structured methodologies. It has the potential to rapidly produce incremental versions of a given system. Boehm et al. (1998) proposes a WINWIN Spiral model that improves on the previous approach by allowing for negotiations between the client and the developer. The overall objective for these negotiations is to have a 'win-win' scenario for both parties. Hoffer et al. (1999) present an Object Orientated Design model that is heavily based on Object-Oriented (OO) theorems to develop a system that is based upon 'objects' rather than information or processes.

The research has shown that there are several methodologies in use for system development. Some of these are oriented towards multimedia system development, however most of them are focused on the construction of traditional IS. With so many alternatives available, developers of such systems must use a variety of approaches. This paper investigates if this phenomenon applies to multimedia system development.

INVESTIGATING THE ISSUE

In order to discover the most widely used methodologies for creating multimedia systems, a survey was conducted that investigated the approaches used by multimedia developers within Australia. The questionnaire was distributed via electronic mail, and comprised of eleven questions that were designed to elicit both qualitative and quantitative data from organisations. These questions covered topics such as the size of each organisation, and the various platforms they develop multimedia applications for, such as the World Wide Web (WWW), CD-ROM, DVD, standalone multimedia systems (e.g. interactive kiosks), and standalone audio/video productions (e.g. DVD-Video). Other topics discussed on the questionnaire included current and previous methodologies used by developers for a given platform, and their justification for using or discontinuing a chosen model. From the information ascertained, various trends relating to the use of traditional methodologies for multimedia products were uncovered. From the 254 surveys sent out to multimedia developers, 50 were confirmed to have read the survey, and 18 responded, which translates to a 36 percent response rate.

METHODOLOGIES USED TO CREATE MULTIMEDIA SYSTEMS

Ninety-four percent of respondents indicated which development models they used to create multimedia systems. A summary of the range of methodologies used by developers is shown in Table 1. A majority of developers used more than one methodology to create multimedia products. Eighteen percent use five different development models, which may indicate that some developers feel a multiplicity of approaches is required. Thirty-five percent use only one, and six percent used no methodology at all.

METHODOLOGIES USED FOR VARIOUS PLATFORMS

Specific methodologies are defined as development processes that are always applied for a given platform. For example, a developer may only use the Waterfall model for developing web sites: therefore, that is deemed as a 'specific' approach. Alternatively, developers may have several suitable models that could be deployed for a given platform. Each one of these is defined as an 'in-specific' approach. Tables 2 and 3 show the utilisation rate (on a specific and in-specific basis) of a development model for a given platform.

The clear majority of developers (66 percent) tend not to use a specific model when developing web-based multimedia. Twenty-one percent specifically use their own customised development process. Phased Development is seldom used, and there are some organisations that do not employ any methodology. The results indicate that almost all of the methodologies detailed in previous discussion are used on a sporadic

Table 1. Range of methodologies used by developers

Number of Methodologies	Total (%)
Zero	6
One	35
Two	24
Three	6
Four	6
Five	18
Six	6

basis. Developers used their own customised processes thirty-three percent of the time. Other methodologies are used less frequently. There were no developers found that employed the Spiral or WINWIN Spiral models.

Methodologies used for creating CD-ROM applications show that a clear majority of developers do not use any specific approach. Only a quarter of developers specifically use their own customised approach. Like Web development, a range of approaches was used on an in-specific basis. Customised development models are again the most likely to be used, with theoretical methodologies less favoured.

The majority of developers do not use a set approach for creating DVD applications. This reinforces the same trend that was prevalent for both WWW and CD-ROM based multimedia. Twenty-nine percent specifically use their own customised approach. For organisations that do not apply a specific model, several methodologies may be applied, although the majority still choose to apply their own proprietary model.

Again, developers tend to use no specific approach when developing standalone multimedia systems, with fifty-seven percent preferring to use a range of different methodologies. However, twenty-nine percent of organisations still prefer to use their own custom development methods. If they do not use a specific model, alternative methodologies are used less frequently.

The current trend for specific approaches remains when creating standalone audio / video productions. The majority of developers tend to use a variety of approaches. A more diverse range of approaches are employed by developers who choose not to implement a specific methodology, with ten alternative models identified. Customised methodologies remain most favourable approach.

For multimedia that is developed for mobile computing platforms, all the respondents indicated that they did not use any specific approach. Instead, a range of approaches are used. Developers used their own customised approach forty percent of the time. Waterfall, RAD or Prototyping models were used less often.

MOTIVES FOR UTILISING CHOSEN METHODOLOGIES

Seventy-two percent of organisations provided some insight into the motives behind using a given development model. Interestingly, cost seemed to be an irrelevant issue, with only four organisations citing this as a reason for using their chosen models. One developer mentioned 'budget constraints' as a reason for using either RAD or the Waterfall model for developing multimedia systems. Conversely, another organisation (who did not indicate the processes that they used) noted

Table. Utilisation rate of methodologies (specific and in-specific) for creating WWW, CD-ROM and DVD multimedia

Model	WWW		CD-ROM		DVD	
	Specific (%)	In-specific (%)	Specific (%)	In-specific (%)	Specific (%)	In-specific (%)
Waterfall / Structured	0	3	0	5	0	0
RAD	0	10	0	14	0	11
Phased Development	7	13	6	11	14	11
Prototyping	0	10	6	11	0	11
Throwaway Prototyping	0	3	0	3	0	0
Incremental	0	6	0	5	0	0
Evolutionary-Delivery	0	6	0	8	0	11
Spiral	0	0	0	0	0	0
WINWIN Spiral	0	0	0	0	0	0
Object-Oriented	0	13	0	14	0	0
Custom	21	33	25	26	29	56
None	7	3	6	3	0	0
Other	0	0	0	0	0	0
Don't Use A Specific Model	65	N/A	57	N/A	57	N/A

Table 3. Utilisation rate of methodologies (specific and in-specific) for creating standalone multimedia, audio / visual and mobile computing systems

Model	Standalone MM		Standalone AV		Mobile Computing	
	Specific (%)	In-specific (%)	Specific (%)	In-specific (%)	Specific (%)	In-specific (%)
Waterfall / Structured	0	8	0	9	0	20
RAD	0	15	0	9	0	20
Phased Development	14	15	9	9	0	0
Prototyping	0	15	0	9	0	20
Throwaway Prototyping	0	0	0	5	0	0
Incremental	0	0	0	9	0	0
Evolutionary-Delivery	0	0	0	9	0	0
Spiral	0	0	0	0	0	0
WINWIN Spiral	0	0	0	0	0	0
Object-Oriented	0	8	0	9	0	0
Custom	29	39	18	27	0	40
None	0	0	9	5	0	0
Other	0	0	0	0	0	0
Don't Use A Specific Model	57	N/A	64	N/A	100	N/A

that the Waterfall model became 'incredibly expensive' if modifications to a system were required. This same developer also concluded that '\$\$\$ [sic] almost always make the final decision' when deciding upon a given development model.

Deadlines also received little attention amongst developers, with only four indicating this influenced their decision. One developer noted that the Waterfall model was a good choice if they knew precisely what was required. Another organisation customised their development process to fit within the client's preferred timeline of development. A developer who used a range of methodologies also indicated that deadlines prejudiced their decision to implement either a traditional model, or their own customised approach.

A lack of experience in using other development models was cited by only one developer as a reason for adhering to their chosen methodologies, which were the Waterfall Model and RAD.

Project requirements were the main reason for using a diverse range of approaches, which was cited by forty-six percent of organisations. One developer used a customised approach, based upon many traditional models, in order to meet the client's needs. They felt that this process has been beneficial because their clients had generally been satisfied with the outcome of their project. An alternative approach, used by the same developer, was to have a list of pre-defined prototypes from which a client can pick and choose. Another developer noted that they used Prototyping for all platforms, particularly when user requirements were imprecise. In addition, one organisation felt that no two projects are ever the same, so they need to use a range of approaches for the various platforms for which they develop.

A solitary organisation also mentioned their reasons behind use their own proprietary method. They believe that their process covers all aspects of software development, and can be used for creating any form of multimedia, as well as other varieties of software. This company also believed that this process was highly flexible and was able to meet a various range of project requirements.

A developer who did not use any methodology claims they do not see a purpose for deploying any kind of model. Instead, their employees are encouraged to work using whatever process they deem applicable, and by any guidelines that are specific to a given project.

Few developers (16 percent) acknowledged that they had previously used other development methodologies. One developer, who did not specify which approach they used, suggested that none of the development models could apply to all types of development. Another had tried to utilise Object-Oriented methodologies, but found the design aspect too difficult and awkward.

CONCLUSION

The findings show that there are many and varied methodologies used to create multimedia systems, but there is no solitary approach that is more suitable than others. Developers tend to use a wide range of approaches when developing multimedia systems. From this range of approaches, it appears most likely that they will use their own proprietary methods. Developers also use established development methods, but not as frequently as they will use a customised approach.

Most developers do not rely upon one specific approach when creating multimedia, instead utilising a range of methodologies. This would indicate that one specific model cannot cater for the various platforms that do exist. Obviously, there are many physical differences between each platform, so this may indicate that their approaches must also change.

Project requirements are the main influence behind a developer's decision to select a given methodology. Other factors, such as cost and timelines also play some part in the choice of development model, but are not as important as the requirements of a project. It also appears that developers will stick to the models they know, as few indicated they had discontinued the use of other development methodologies.

Customised approaches are the most likely to be used by developers for creating any kind of multimedia system. This may indicate that established methodologies are inappropriate for creating multimedia. As the origins of many established methodologies come from traditional IS development, this provides some foundation to such an argument. However, considering that there are several developers who do utilise such theoretical approaches, such a statement tends to be unfounded. Perhaps not all multimedia developers have the same education or training as more traditional computing employees. This could reflect the lack of formal models used in multimedia systems development. A more troubling statistic is that there are developers who do not employ any methodology for creating multimedia. Any kind of methodology will bring structure and control to the development process, so there are obvious benefits available.

Another angle that could be adopted is that there is a need for more methodologies to be created specifically for multimedia. Developers could have difficulty trying to adopt traditional IS methodologies for multimedia, therefore there is a trend to use their own methods which are easier to understand. It is clear there are models that have been successfully applied for multimedia development, but their origins appear to lie in the IS arena.

Whilst the trend for proprietary approaches remains apparent, further research into such models could reveal more details about how those models are put to work. It is possible that there are certain qualities within these customised models that reflect those found in traditional methods. This would then indicate that there is an underlying theoretical influence within these approaches. A wider international study with a larger response rate may ratify the current indicators of methodology usage, or perhaps provide new answers. A study that measured the levels of overall project success and client satisfaction with a given methodology may offer a more credible insight into what are the most suitable methodologies for creating multimedia systems.

Overall, the majority of developers use some form of methodology. However, it appears that traditional methods are not well accepted amongst multimedia developers, so there is a tendency to use customised approaches. Many possibilities have been discussed as to the cause of this finding. Until more multimedia-specific development methodologies are conceived and promoted, it would appear that this current trend is likely to continue.

REFERENCES

- Boehm, B., Egyed, A., Kwan, J., Port, D., Shah, A. and Madachy, R. (ed) (1998) "Using the WinWin Spiral Model: A Case Study", *Computer*, IEEE Computer Society, New York, Vol 31, Issue 7, pp. 33-44

4 2006 IRMA International Conference

- Dennis, A. and Wixom, B.H. (2000) *Systems Analysis and Design: An Applied Approach*, John Wiley & Sons, New York, pp 9-16, 72-74, 120-124.
- Hoffer, J.A., George, J.F. And Valacich, J.S. (1999) *Modern Systems Analysis and Design (2nd Ed.)*, Addison-Wesley, Reading, pp 24-34, 486-498.
- MacCormack, A. (ed.) (2001) "Product-development Practices That Work: How Internet Companies Build Software", *MIT Sloan Management Review*, MIT Sloan School of Management, Cambridge, Vol 42, Issue 2, pp. 75-84
- Pfleeger, S.L. (1998) *Software Engineering: Theory and Practice*, Prentice Hall, New Jersey, p. 126
- Pressman, R. (2000) *Software Engineering: A Practitioner's Approach*, McGraw-Hill, New York, pp 29-47
- Satzinger, J.W., Jackson, R.B., Burd, S.D. (2004) *Systems Analysis and Design in a Changing World (3rd Edition)*, Course Technology, Boston, p. 42
- Siegel, D. (1997) *Secrets of Successful Web Sites: Project Management for the World Wide Web*, Hayden Books, Indianapolis, pp 160-168, 206-282.
- Sommerville, I. (2001) *Software Engineering (6th Edition)*, Addison Wesley, Essex, pp. 53-55
- Vaughan, T. (1998) *Multimedia: Making it Work (4th Edition)*, Osborne/McGraw-Hill, Berkeley, pp 428-434

0 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

www.igi-global.com/proceeding-paper/methodologies-developing-multimedia-systems/32696

Related Content

Challenges in the Digital Transformation Processes in Higher Education Institutions and Universities

Marco A. Coraland Augusto E. Bernuy (2022). *International Journal of Information Technologies and Systems Approach* (pp. 1-14).

www.irma-international.org/article/challenges-in-the-digital-transformation-processes-in-higher-education-institutions-and-universities/290002

Presidential Elections Web 2.0

Ramona Sue McNealand Lisa Dotterweich Bryan (2018). *Encyclopedia of Information Science and Technology, Fourth Edition* (pp. 3612-3620).

www.irma-international.org/chapter/presidential-elections-web-20/184071

Data Recognition for Multi-Source Heterogeneous Experimental Detection in Cloud Edge Collaboratives

Yang Yubo, Meng Jing, Duan Xiaomeng, Bai Jingfenand Jin Yang (2023). *International Journal of Information Technologies and Systems Approach* (pp. 1-19).

www.irma-international.org/article/data-recognition-for-multi-source-heterogeneous-experimental-detection-in-cloud-edge-collaboratives/330986

Deployment of Enterprise Architecture From the Activity Theory Perspective

Tiko Iyamuand Irja Naambo Shaanika (2018). *Encyclopedia of Information Science and Technology, Fourth Edition* (pp. 2943-2952).

www.irma-international.org/chapter/deployment-of-enterprise-architecture-from-the-activity-theory-perspective/184006

Recommender Systems Review of Types, Techniques, and Applications

George A. Sielis, Aimilia Tzanavariand George A. Papadopoulos (2015). *Encyclopedia of Information Science and Technology, Third Edition* (pp. 7260-7270).

www.irma-international.org/chapter/recommender-systems-review-of-types-techniques-and-applications/112423