Many software projects fail. In this research project the software development practices of 36 organisations are investigated to determine to what extent standard development methods are being deployed. It is found that, in general, these standard methods are used only with major modification and adaptation.

In all cases of software development, the requirement must be analysed, a design must be developed, implementation must proceed in a managed and controlled fashion, and organised testing must be undertaken. In addition, provision must be made for handling change during all phases, and particularly during the implementation and operational phases of the application. Modern web-based applications extend to a wide range of users, and questions of security and aesthetics are rapidly increasing to critical levels of importance.

Prescriptive process models bring useful structure to software development work and provide a roadmap for software teams. These prescriptive process models, or methods, include:

- The Waterfall model
- Incremental process models
- The RAD model
- The Spiral model
- Concurrent development models
- Other specialised process models.

More recently, organisations adopt what are called agile software development processes. These agile processes are aimed at boosting customer satisfaction and encourage early incremental delivery of software. Project teams tend to be small and highly motivated, while informal methods and minimal software work products are the norm.

Web-based applications are becoming increasingly integrated into business strategies for a wide range of companies, for example e-commerce, and these applications have specific characteristics that must be considered in the development process. Web applications are network intensive, have a high level of concurrency, face an unpredictable load and are usually in a state of continuous evolution. It has been found that the agile process methods can be applied successfully to these applications.

**Prescriptive process models bring useful structure to software development work and provide a roadmap for software teams**

These agile process models include:

- Extreme Programming
- Adaptive Software Development
- The Dynamic Systems Development Method
- Scrum
- Crystal
- Agile Modelling.

An additional feature in the software development landscape today is the growing acceptance of open source software. This software has a new set of support, licensing, training and maintenance characteristics. Are the same development methodologies relevant?

**The research project**

This research investigates software development methods in an organisation today. Each of 36 students in the final year BSc class identified an individual organisation. Letters of introduction including an outline of the research objectives were prepared. Students made contact with the organisation and explored the organisation’s system development methodology through questionnaires, interviews and tours.

Additional information was derived from secondary sources such as reports, internet searches and the media. Several staff members were interviewed to corroborate details where necessary.

The students prepared a report and presentation outlining the result of the research. This paper summarises the results of the investigation.

**Perceived strengths and weaknesses of some methods**

The organisations commented on some of the methods, as follows:

**RAD**

**Strengths:**

- Early functionality, with little delay before the client sees results
- Quick response to business needs
- Facilitates risk control
- Documentation emphasises the end product rather than the development of the product
- Fosters early conversation between client and developer
- Limits exposure to forces of change
- Saves development time.

**Weaknesses:**

- Some decisions traditionally made by management are devolved to the development team
- Can be seen as “quick and dirty”
- Can suffer from bad documentation
- Final system can display poor performance.
**Waterfall**

**Strengths:**
- Disciplined process is easier to manage
- Forces proper initial analysis and design
- Termination possible before costs have escalated
- Well defined phases, well defined inputs
- Recognises the proper sequence of events that should take place.

**Weaknesses:**
- Little room for error correction
- Little knowledge transfer
- The solution does not evolve with technological advances during the course of the project
- No early feedback through prototyping
- Good forecasting ability essential
- Poor consistency across projects
- Difficult to implement
- Limited re-use of intellectual capital
- Idles key team members
- Defers testing to end.

**Iterative**

**Strengths:**
- Allows for error correction
- Encourages prototyping
- Keeps track of emerging technologies
- Feedback improves solution.

**Weaknesses:**
- Difficult to tailor
- Can suffer from “analysis paralysis”
- Perception that there is no need to “get it right the first time”, leading to unnecessary rework
- Encourages moving development targets
- Poor consistency across projects
- Difficult to implement
- Limited re-use of intellectual capital.

**Iterative/incremental**

**Strengths:**
- Allows for error correction
- Encourages prototyping
- Tracks emerging technologies
- Feedback improves solution
- Defined iterations and the system boundary assist “get it right first time” ideal
- Clear development goals.

**Concurrent**

**Strengths:**
- Fast product delivery
- High quality product
- Easy adaptation to changes.

**Weaknesses:**
- Difficult/expensive to manage
- Requires strict control to ensure system integrity

**Summary of results**

The 36 organisations were split into three groups:
- Global software developers (7)
- Local software developers (13)
- Organisations that develop their own software (16)

In order to maintain anonymity, the names of organisations are omitted from this paper.

This summary of the major comments made does not necessarily cover all the aspects of every methodology mentioned.

**Group 1: Global software developers**

This group comprises the large, multi-national software organisations that develop widely-used operating systems, databases and major business applications.

Because of the diverse nature of their products, these organisations have developed generalised development methodologies that may be adapted to each application.

Examples are the Java Custom Development Method (JCDM), the Microsoft Solutions Framework (MSF), the Application Implementation Method (AIM) and the Rational Unified Process (RUP). These are defined, disciplined and deliberate approaches to software development based on specific principles, models, concepts, guidelines and proven practices. The assumption is that there is no single structure or process that optimally applies to the requirements and environments of all projects.

Detailed assessment of these generalised methodologies lies outside the scope of this paper, however, they essentially combine elements of the Waterfall, RAD, Concurrent and Iterative methodologies, and also include some agile processes.

It appears that the Waterfall method provides the basic structure, but that the needs of the market, that is, speed of development, flexibility in the face of changing user requirements and the need to incorporate evolving technology have led to the adoption and incorporation of more responsive techniques. The question arises as to whether these incorporations lead to errors and defects in the delivered code.

The Waterfall model tends to be used where the system can be defined clearly at the outset, and little change is anticipated.

RAD development emphasises a very short development cycle (some 60 to 90 days) and is used primarily for information system applications. Teams comprising the project manager, systems analysts, systems developers and the client hold joint application development (JAD) sessions to facilitate client interaction throughout the development of the product. Object-oriented analysis and design techniques are incorporated, and provide the framework for a component-based process model for software engineering. Agile techniques such as Extreme Programming are deployed to provide a solution to a specific problem, or to gain competitive advantage.

Where high quality code needs to be delivered as fast as possible, concurrent development techniques are deployed. These techniques shorten the development period and permit relatively easy adaptation to changing user requirements. However, as tasks are carried out by different project teams and because development activities are mutually dependent, management of these projects is complex.

In RUP, used also by a large engineering organisation, the software lifecycle is split into four sequential phases: inception, elaboration, construction and transition. Each phase is concluded by a major milestone at which an assessment is performed to determine whether the objectives of the phase have been met. Elements of RAD and the Spiral Iterative methodology are clearly apparent. However, the RUP process is an integrated system supported by a full set of development tools.

Current developments in this group are toward an iterative/incremental approach, in which additional functionality is incorporated in each increment/release, and the cycle of design, build and test is repeated until the desired functionality is complete.

**Group 2: Local software developers**

This group comprises organisations that are primarily based in southern Africa and that develop software solutions for clients.

These organisations develop:
- Specialised solutions in C++
- ERP, EIS, GIS, SAP, CRM and enterprise management solutions
- Upgrades and support
- Database development
- Corporate internet connectivity and content; web application development
- e-commerce and e-learning solutions
- High-end business technology solutions
- Biometric solutions
- Continuous business improvement
- Time and attendance, finance, accounting, human resources and payroll systems
With some notable exceptions, the methods used in this group are based on Waterfall. These organisations do, however, face the same issues as any other software developer in the current market, namely, the need for fast development and the fact that client requirements and technology change. Therefore, in many cases the Waterfall approach is adopted with modification. The systems developers use the best tools at hand, and attempt to improve the flexibility of the Waterfall method by incorporating other techniques. Iterations and prototypes are built in to compensate for changing client requirements, and extreme programming is added to accelerate development. A hybrid development methodology results, and the overall effectiveness and efficiency of the development must be questioned.

Often a specific methodology is mentioned, but assessment of the actual procedure reveals a more ad-hoc and eclectic development strategy. Again, the effectiveness of these approaches must be questioned.

The notable exceptions use JAD sessions followed by 21-day RAD development incorporating iterative prototyping. Also, the iterative method in the form of the RUP is applied, indicating that iterative prototyping. Also, the iterative method must be questioned.

A number of these organisations have, however, moved away from Waterfall and now use Spiral and concurrent methods.

Some application of Agile methods was found, primarily as a tool to speed up the development of projects being developed under a prescriptive method. Extreme Programming is used to prototype what are considered “lightweight applications”, and as a tool in some iterative developments. An example of use of the Dynamic System Development method was identified in the construction sector, where the method was found to be cost effective and yield the rapid, relevant results demanded by the user.

The Microsoft Solutions Framework is applied, as are organisation-specific methodologies that formalise a combination of JAD, Spiral and Concurrent methods.

Conclusions

There are a set of large-scale development methodologies, such as JCDM, MSF, AIM and RUP that are used primarily by the major software developers. These are less used by the local software developers and the organisations in South Africa that develop their own software. In general, these methods are formalised and integrated combinations of the traditional approaches, and need to be adapted or tailored for each application.

Overall, the standard techniques appear to be rarely used in their textbook form.

The method that an organisation states or believes it is using is not necessarily the same as the actual method being used.

- A large percentage of the organisations contacted in this survey combine elements of different methods in unique combinations in order to achieve speed of delivery and flexibility.
- There appears to be no difference in the method used for open source software or for proprietary software.

Therefore, the situation is such that:

- A large number of software development projects fail or are abandoned, and
- The techniques taught to IT/IS students are not being applied properly.

Clearly further investigation is required. The possibilities exist that the methods are not being taught properly, or that the inertia and management structure of the organisation prevents newer methods from being properly deployed, or that the methods are simply too theoretical, are not applicable in the real world, and need to be modified.

This points to a range of different interventions: improving student education in the field of systems development, improving management education in terms of getting management to understand the critical importance of effectively deployed systems, and revisiting the standard software development methods in the light of current business practices and requirements.

Given the substantial cost to the economy resulting from expenditure wasted on projects that do not succeed, further investigation and analysis is recommended.

Contact Roger Silberberg, Innovation Africa, Tel (011) 883-1059, rogers@edu.co.za