A closer look at project life cycles

by Dr. Solomon Bhunu, Corporate GIS, Cape Town

What is a project? Literature presents a plethora of project definitions. Nicholas [1] defines it as such: "a project is a system of people, equipment, materials and facilities organised and managed to achieve a goal". One has to note the absence of the temporary nature phenomenon of a project in this definition.

The Project Management Institute-PMI [2] simply states that, "a project is a temporary endeavour undertaken to create a unique product, service, or result".

A different view to a project is offered according to systems theory (see Jackson [3]). The project follows the systems view of breaking it down into phases. Each phase can possibly be further broken down into lower processes with specific objectives. A systems approach to a project allows us not only to break down the systems (i.e. the project) into subsystems, but provides the techniques and tools to take into account the environment which will be affected, or which will affect the system (assuming it is an open system) - hence considering inputs, outputs and constraints.

Generally, after an analysis of all the literature, one would then summarise a project as a temporary effort of work that meets the following criteria:

- has a start date and an end date
- has a schedule, cost and quality constraints
- is a unique endeavour and contains risk
- has a certain scope.

The project life cycle

A definition of life cycle is stated by Global Knowledge [4]: "Life cycle implies two things: that a process is perpetual and that the sequence of events is obligatory or uni-directionary". However the term "life cycle" is not used in its true sense in projects and systems. Life cycle in this instance is neither a perpetual circle of events nor is the sequence of events rigidly fixed; otherwise this would strongly contract to the project's definition, which assumes a finite period.

Generally, project life cycles have phases, which are sequential, and their ends are punctuated by either technical information transfer or technical component hand over. Project phases vary by project or industry, but some general phases include concept, development, implementation and support.

The general trend depicted by life cycles [2], is that, cost and staffing levels are low at the start, rise during the middle phases, but subside as the project comes to an end (see Fig. 1)

Comparison of different project life cycles

This section provides a comparison of commonly used project life cycles. A brief description of the cycle is given first, then strengths and weaknesses are discussed and lastly a summarised view is provided. The discussions on strengths and weaknesses are based on works by Charvat [5], Business eSolutions [6] and Gardiner [7].

Waterfall

The waterfall model originated in 1970 largely through the efforts of Dr. Winston Royce [5]. The model has gone through a number of changes and revisions ever since. The model gets its name from the analogy of water falling downward (see

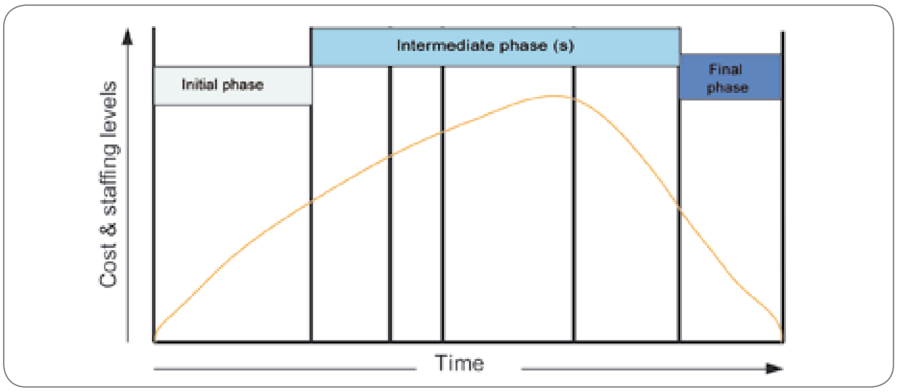

Fig. 1: Project life cycle: Project cost and staffing across the life cycle (adapted from PMBOK [2]).

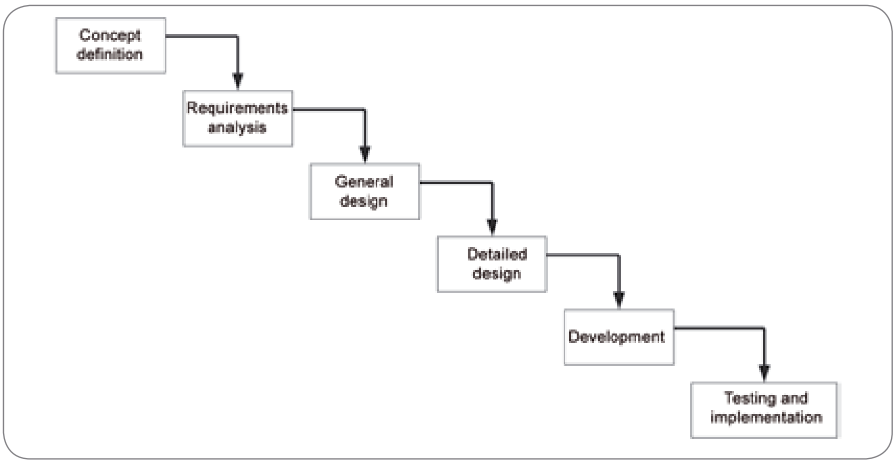

Fig. 2: Water fall project life cycle (adapted from Charvat [5]).

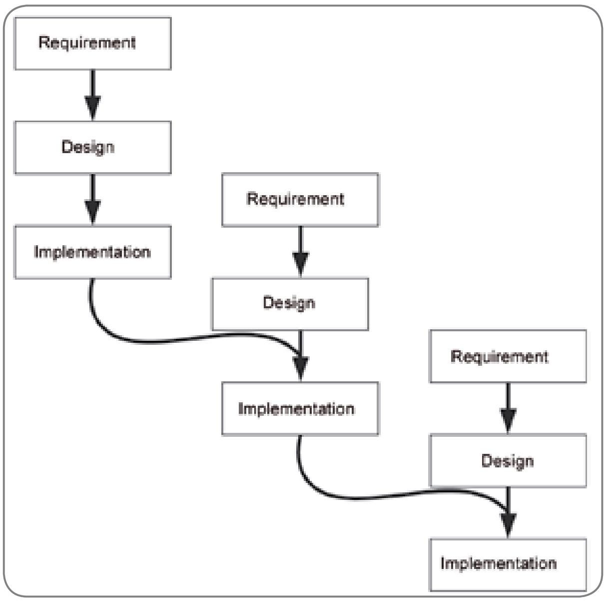

Fig. 3: Incremental project life cycle (adapted from Gardiner [7]).

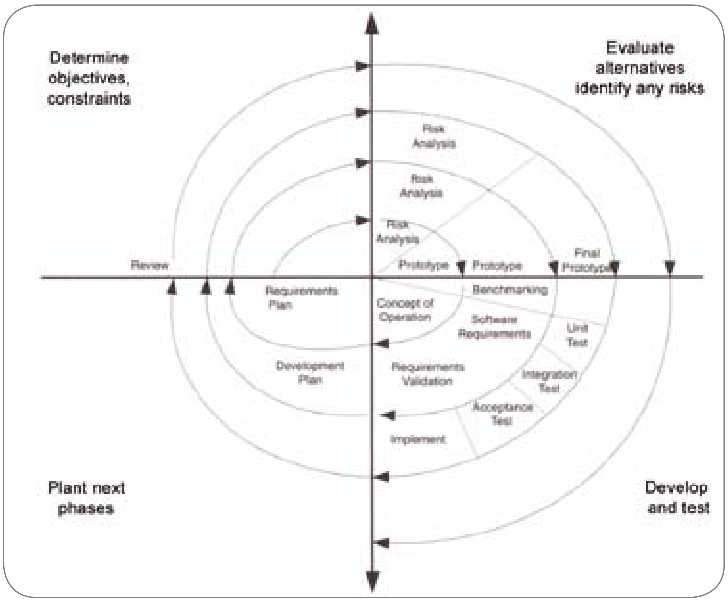

Fig. 4: Spiral model of project life cycle (adapted from Charvat [5]).

Fig. 2). The number of phases varies depending on the project, but there is a clear transition from one phase to the next.

The pure waterfall performs well for products with clearly understood requirements or when working with well understood technical tools, architectures and infrastructures. Its weaknesses frequently make it inadvisable when rapid development is needed. In those cases, modified models may be more effective.

Incremental life cycle

The concept behind the incremental project life cycle is that of aiming to produce usable products at the end of each section. The model (see Fig. 3) uses the waterfall approach in overlapping sections. In reality, the whole cycle will be led by a series of objectives, which are assigned to each section. General objectives are defined at the beginning of the project.

Spiral

The spiral is a risk-reduction oriented model that breaks a software project up into mini-projects, each addressing one or more major risks. The model was developed to counter the problems of the waterfall method. It is cyclic, hence the name spiral as opposed to the linear waterfall model (see Fig. 4). The spiral model has six phases: planning, risk analysis, engineering/development and evaluation. A systems project repeatedly passes through these phases in iterations (called spirals in this model). The baseline spiral, starting in the planning phase, is where requirements are gathered and risk is assessed. Each subsequent spiral builds on the baseline spiral.

Requirements are gathered during the planning phase. In the risk analysis phase, a process is undertaken to identify risk and alternate solutions. A prototype is produced at the end of the risk analysis phase. Software is produced in the engineering phase, along with testing at the end of the phase. The evaluation phase allows the customer to evaluate the output of the project to date before the project continues to the next spiral.

In the spiral model, the angular component represents progress, and the radius of the spiral represents cost.

For projects with risky elements, it is beneficial to run a series of risk-reduction iterations which can be followed by a waterfall or other non-risk-based lifecycle.

Rapid application development (RAD)

Sometimes users just want to see a product they can understand and not have to wait for the development to get off the build line. Traditional software development methodologies usually follow a sequence of steps with formal sign-offs normally at the end of each project phase. This process can be time consuming, but RAD shortens the approach (see Fig. 5).

With RAD, each iterative development cycle delivers a functional version of the proposed solution. The approach is almost cyclic in nature. However, many organisations don't have the time to spend lengthy periods on development; they want to see immediate results. By following a RAD model, clients are able to use a "building block approach" to see the results. Using RAD, results are almost immediate and the product starts becoming tangible. The developers have the advantage of building on their solution and gradually improving it until it reflects what the client requires.

Compared with many traditional methodologies, RAD compresses the analysis, design, development, and test phases into a dynamic series of short iterative development cycles. RAD uses shorter project phases, which means that benefits are realised much more quickly. RAD prototyping can reduce costs. It allows the project

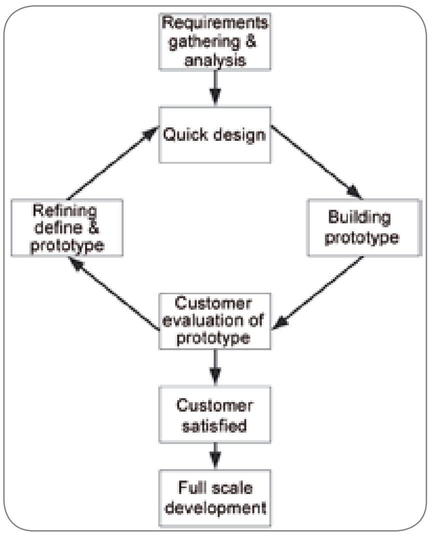

Fig. 5: Rapid application development life cycle (adapted from Gardiner [7]).

manager and the team to identify risks early during the project life cycle. The approach overlaps project phases. However, one of the biggest concerns found with RAD is with quality assurance during the development of the project. Project managers, therefore, must ensure that quality assurance is built into the project.

Evolutionary (also known as iterative)

The evolutionary cycle also maintains a series of phases that are distinct and cascading in nature. As in the other cycles, each phase is dependent on the preceding phase before it can begin and requires a defined set of inputs from the prior phase. As Fig. 6 portrays, the evolutionary model is similar to the incremental in that during the design phase development is broken into a distinct increment or subset of requirements. However, only this limited set of requirements is constructed through to implementation. The process then repeats itself with the remaining requirements becoming an input to a new requirements phase. The "left over" requirements are given consideration for development along with any new functionality or changes. Another iteration of the process is accomplished through implementation with the result being an "evolved" form of the same software product. This cycle continues with the full functionality "evolving" overtime as multiple iterations are completed.

Agile/extreme

Agile or extreme programming is just emerging as a potentially viable systems development life cycle (SDLC) methodology. The basic tenets in philosophy are consistent with other valid SDLC methodologies and the basis in processes is founded on the waterfall methodology as are other SDLC methodologies. However, rather than focusing on documenting requirements, agile/extreme seeks to compress time in the "scope through design" phases by utilising stories, scenarios, and use cases as replacements for formal documentation. Likewise development and test take place inside of a pre-defined "time box" as an additional means of time compression (see Fig. 7).

The emphasis is on reaching implementation as rapidly as possible. The goal is often to reach implementation in days rather than months or years. Additionally, as Fig. 7 depicts, multiple time boxes can be employed in an incremental fashion to achieve additional functionality. As in incremental, these can be sequential or in parallel. Likewise, evolutionary techniques are employed to achieve additional software functionality shown via the dashed lines. Development and test are iterated to generate rapid evolution of product functionality and "refactoring" used to redesign or re-scope the effort.

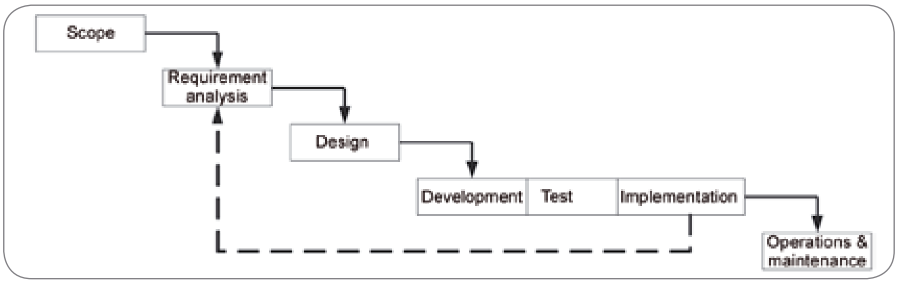

Fig. 6: Evolutionary project life cycle (adapted from PM Solutions [8]).

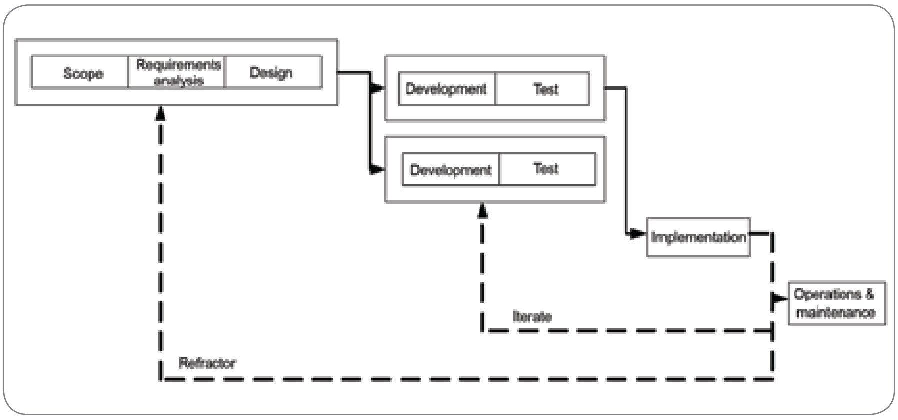

Fig. 7: Agile or extreme programming.

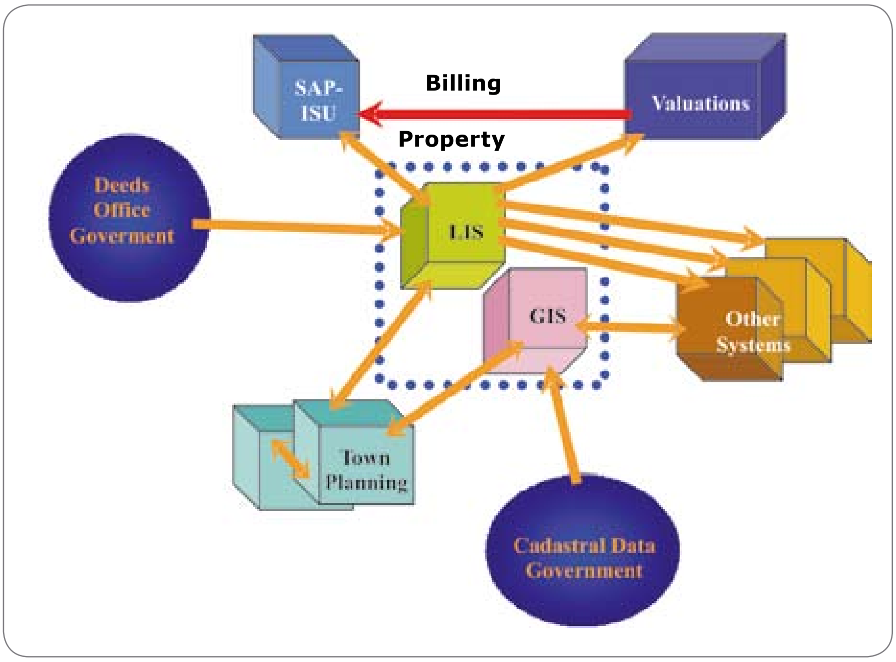

Fig. 8: Illustration of the current LIS and GIS systems interfacing with business.

Background and description of the project

Currently the Municipality of Cape Town has two main property information systems (see Fig. 8).

The geographical information system (GIS): This system is spatially enabled

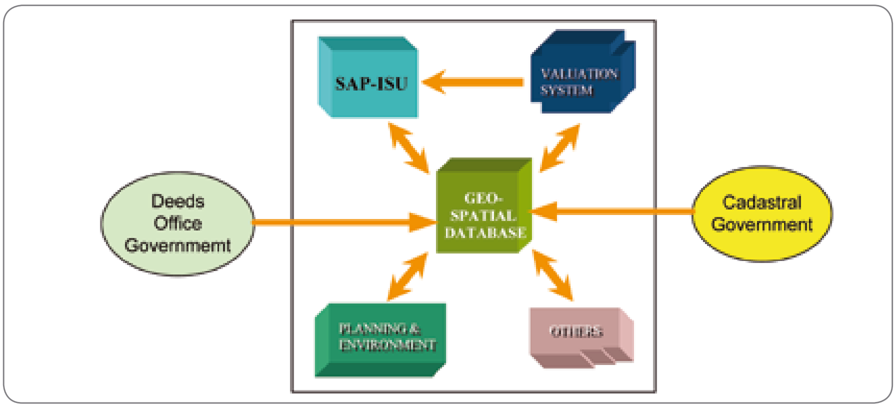

Fig. 9: Proposed integrated property system

i.e. it allows for graphical display of properties and showing the true spatial relationships.

Land information system (LIS): This system captures the property information again but it is basically tabular based, with no ability to spatially display these properties. It is a legacy system for the municipality, dating back ten years or more. It contains most of the attribute data (i.e. registration data, ownership details, etc) of the ±650 000 properties under the municipality's jurisdiction

All the other business systems to the municipality are linked to these two systems. Since the LIS is an old system, most of the systems link to it and on the other hand, the most recent systems link mainly to the GIS. Data transfer across the two systems is done through a combination of manual means and semi-automated processes (rarely). The most interesting challenge faced by the municipality is that of linking property information to an Enterprise Resource Planning system known as SAP (i.e. Systems, Applications and Products), ERP-SAP. This link will allow for the easy billing and evaluation processes being linked to the property data.

Rationale for the project

The two systems, LIS and GIS, are the main source of property information and as such, property related business systems are linked to them. Since there is no direct link between the LIS and GIS, the data relating to the same property entities is not synchronised. The data management, maintenance and quality assurance processes are not standardised at all. Consequently it presents huge challenges in trying to put in place an investigative process to determine the properties that are not being billed. When business draws data from these two systems, the inevitable daunting task of doing data matching has to be done before use. This has meant that the municipality loses a lot in the implementation of these extra time consuming processes around data.

Proposed solution

The proposed solution involves the development of a single property system (i.e. geospatial database system) which will be interfaced with all the municipality's business systems (see Fig. 9).

Choosing the right project life cycle model

Formal project nomenclature equates; better to quality, cheaper to cost, and faster to schedule. Such a model provides the business with a mechanism to strategically choose the fitting project life cycle and management tools. These three measures, although informal, become the value proposition of the business. Such a proposition is demonstrated visually in Fig. 10.

Secondly, a structure for selecting an appropriate life cycle model can be operationalised through a graphical framework [8]. The D3 Cube (Fig. 11), a decision cube, is a framework that provides a common approach to selecting one or more life cycle models. There are three selection criteria each of which map to a specific element of business value and, when combined, form the basis for selecting a cycle to optimise value to the business. As shown in Fig. 11, the three key

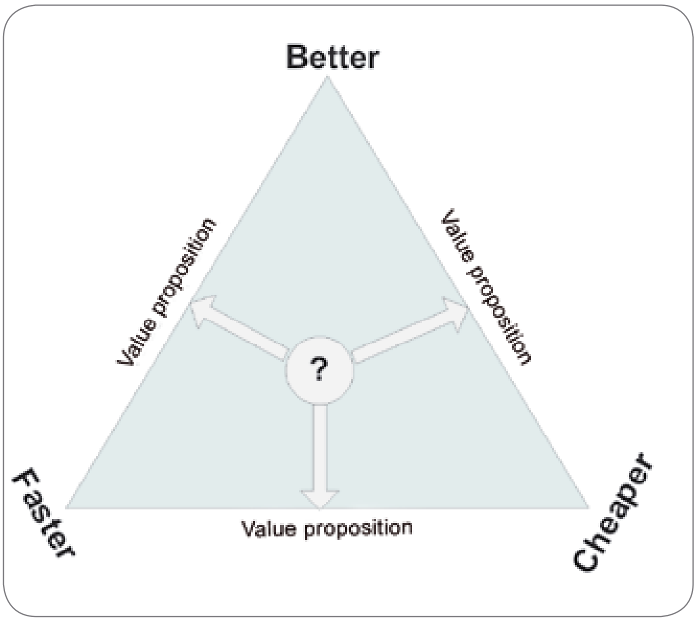

Fig. 10: Business value proposition (adapted from PM solutions [8]).

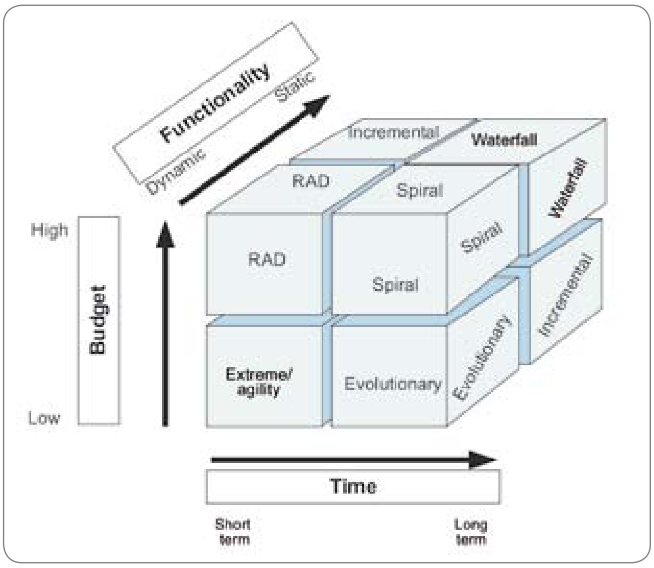

Fig. 11: The D3 cube (adapted from PM Solutions [8]).

Methodology & Criteria	Advantages	Disadvantages
Waterfall		
Budget: *high* Time: *long term* Functionality: *static*	Clearly defined stages. Assures delivery of initial requirements. Well documented process and results.	Lack of measurable progress within stages. Cannot accommodate changing requirements. Resistant to time and/or budget compression.
Incremental		
Budget: *high* Time: *short term* Functionality: *static* or Budget: *low* Time: *long term* Functionality: *static*	Early and periodic results. Measurable progress. Supports parallel development efforts.	Demands increased management attention. Can increase resource requirements. No support for changing requirements.
Evolutionary		
Budget: *low* Time: *long term* Functionality: *dynamic*	Supports changing requirements. Minimises time to initial operating capability (IOC). Achieves economies of scale for enhancements.	Increases management complexity. IOC only partially satisfies requirements and does not have complete functionality. Risk of not knowing when to end the project.
Spiral		
Budget: *high* Time: *long term* Functionality: *dynamic*	Supports changing requirements. Allows for extensive use of prototypes. More accurately captures requirements. Increased management complexity.	Defers production capability to end of the SDLC. Risk of not knowing when to end the project.
RAD (rapid application development)		
Budget: *high* Time: *short term* Functionality: *dynamic*	Minimizes time to delivery. Accommodates changing requirements. Measurable progress.	Increases management complexity. Drives costs forward in the SDLC. Can increase resource requirements.
Extreme/agile development		
Budget: *low* Time: *short term* Functionality: *dynamic* or Budget: *low* Time: *short term* Functionality: *static*	Rapid demonstrable functionality. Minimal resource requirements. Supports fixed or changing requirements.	Not conducive to handling complex dependencies. Creates quality assurance (QA) risks. Increased risk of sustainability, maintainability, and extensibility.

Table 1: Summary of D3 cube inferences and project life cycles

criteria are functionality, budget, and time. Since my focus is on systems development, the cube mainly contains the life cycles that fall into the category of SDLC.

Using these three criteria a SDLC model is be selected to best fit the needs and direction of the Municipality of Cape Town. To use the cube to simply plot the project style(s) along each of the three criteria and the resultant quadrant is the recommended cycle. However, the key to this approach is correctly quantifying each criterion through its objective measure. This selection is given below:

Functionality

We are making a selection based on the whether the functional requirements of the projects will be dynamic (changing) or static (non-changing) throughout the life cycle. Reality dictates that business, and the software supporting it, will change over time. Therefore, the question is not really "if " the requirements will change but "when" the requirements will change. However, functionality is regarded as static if the requirements may change after the product is released for the first time and dynamic if the requirements change before the product is released the first time. For the municipality, the proposed system functionalities are a function of the business systems which may change over time. This then implies that the functional requirements might change even before the product is released, but these changes are first sanctioned before implementation, hence one could regard them as static or having minimal effect on the system.

Budget

Organisation size, average project size, and return on investment (ROI) requirements are all considerations when quantifying budget for the decision cube. Additionally, budget can be quantified in terms of dollars, hours, resources, etc. depending upon how organisations budget. A good guideline for determining the quantification for low to high budget is to create a range for the municipality's projects and then establish the organisation's risk tolerance figure. Generally, this figure within the City of Cape Town is R199 000. The proposed budget for the Integrated Property System is R10-million, which then classifies the project as a risk.

Time

Time is actually referred to as time horizon because we are interested in knowing the projected completion of the project. In selecting the quantifying measure for short-term versus long-term one looks at the nature of business and selects a timeframe in accordance with what the business needs to succeed. In the instance of

Cycle phase	Part 1	Part 2	Part 3	Integrated system
Requirements Analysis	Workshop papers, interview analysis papers, summary of legacy systems, GAP analysis	Detailed technical requirements papers, workshop papers	Business re-organisation workshop papers, questionnaires	Feasibility paper + Part 1+Part 2 Part 3
Design	Business case outline	System conceptual design, system logical design, system physical design	Business re-organisation paper	Part 1+Part 2 + Part 3
Development	Business case	Business case + integrated property system application	Business case + integrated property system application + business units re-organisation strategy	Integrated system certification paper + Part 1+Part 2 + Part 3

Table 2. Summary of phase outputs.

the selected project, the funding is time constrained, that is, it is only available for a three year period, at the end of which, the funds can be retained by central government if the project is not complete. However, for the municipality, projects are normally run within a financial year and rarely do they run across two years, in which case, they are classified as long term projects.

From the above discussion, the chosen project cycle model which suits the following criteria:

Functionality = Static

Budget = High

Time = Long Term

This will result in the choice of the waterfall.

Table 1 summarises the key attributes of life cycles and the D3 cube inferences.

Addressing choice anomalies

We do not live in a perfect world and there will always be those projects that present themselves as anomalies to the preferred methods. When the criteria of a given project (time, budget, functionality) applied to the Decision Cube results in a methodology recommendation that is not one of the preferred methodologies for your organisation, you can effectively force the recommendation into one of your preferred selections by moving to the preferred methodology that is in the quadrant nearest the recommendation.

In order to select the methodology most effective for your project, you should move no more than one quadrant in any direction. In the Integrated Property System, the

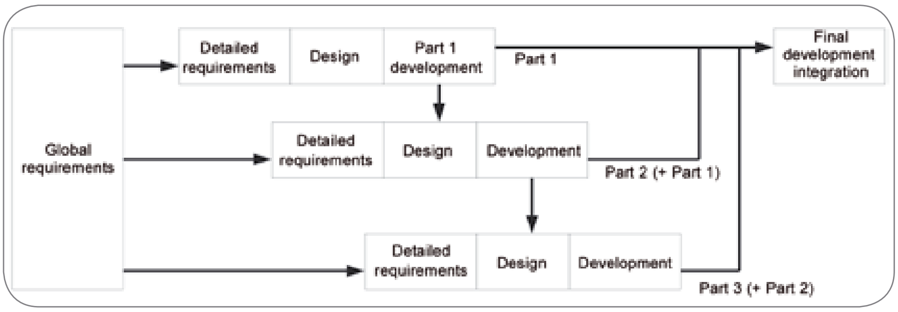

Fig. 12: The proposed project cycle for the Integrated Property System.

preferred life cycle is the incremental model, and is a quadrant away. The implications for such a decision could be far reaching and affect both tactics and strategy of the organisation. In this project, it would mean adjusting the budget. A proposed solution would then be to adopt a hybrid solution and the implementation of this solution is explained in Fig. 12.

The Integrated Property System will have three main identifiable products which will be integrated into one:

Part1: The business case: Although there are varied definitions as to what goes into a business case, in this instance the project will be expected to see a detailed analysis of the proposed new business processes and how they take into account old or legacy systems, among other things.

Part 2: Developed application integrating the property system with ERP-SAP and other business systems: this is the software engineering part in which a physical application interfacing with all relevant systems is developed.

Part 3: Proposed business units are reorganised to sustain the new system and business processes. In order to manage the new systems, organisation changes and new papering lines might be created and this has to be part of the project.

Table 2, summarises the products of each phase within the incremental approach adopted.

Conclusion

Significant 'lessons learned'

Managing people is a project management function but as in most cases the project team members will execute different parts of the project life cycle. So it follows that the designation of roles will correlate to the project life cycle. Consequently if there is a lack of understanding on the project life cycle, a manager will have problems in assigning work and addressing the work problems. A good work plan should be based on a sound understanding of the project life cycle. If this is not the case, one is likely to have false estimate and a poor work plan.

Although sound management process groups are clearly defined in literature (see [2]), this alone does not mean general success to the implementation and management of projects. The underlying fact will be a clear understanding of the project life for that specific project.

Using standard life cycles can only help in utilising resources efficiently and in predicting achievable results.

The project phases are extremely useful in planning a project since they provide a framework for budgeting, manpower and resource allocation, and scheduling project milestones and project reviews.

What findings are important and/ or represent a threat to the project success?

Since projects involve people, one should understand that a life cycle alone will not achieve complete success; there is some degree of complexity and uncertainty that can not be easily controlled.

Incorrect, or no, choice of a project life cycle would lead to wrong budget estimates and wrong techniques being applied to activities and inappropriate assumptions being made.

While selecting the right SDLC methodology is challenging, the challenge is not insurmountable. With a clear understanding of the business and a framework for guidance, selecting a fitting SDLC can be readily achieved. The D3 decision cube is an effective mechanism for assisting in the selection of one or more SDLC methodologies. The decision cube assists the business in making a selection based on objective criteria specifically relevant to the business. The decision cube is an objective framework that allows IT and the LOB organisations to work together in the selection of a methodology that brings value to the business.

However, achieving success through this selection means utilising the SDLC methodology, seeking to align this methodology with your PM methodology, and driving to maturity in both. A proper methodology in a maturing environment will enable the business to build software that assists the business in realising its value proposition.

Acknowledgements

The paper acknowledges the work that has been done by Jasper Compion on managing the case project.

References

[1] J M Nicholas: *Project management for business and engineering: principles and practice, 2nd edition*, Elsevier Butterworth-Heinemann, Burlington, USA, pp 51–80, 2004.

[2] Project Management Institute: *A Guide to the Project Management Body of Knowledge* (PMBOK Guide), Third Edition, Four Campus Boulevard, Newtown Square, PA 19073-3299 USA, 2004.

[3] M C Jackson: *Systems Thinking: Creative Holism for Managers*, John Wiley and Sons Ltd, 2003.

[4] Global Knowledge: *An Introduction to PMI's Project Management Life Cycle*, www.globalknowledge.com, 2006.

[5] J Charvat: *Project Management Methodologies: Designing, Implementing, and Supporting Methodologies and Processes for Projects*, John Wiley & Sons, 2003.

[6] Business eSolutions: *Project Lifecycle Models: How They Differ and When to Use Them*, www.business-esolutions.com/, 2006.

[7] P Gardiner: *Project Management: A Strategic Management Approach*, Palgrave Macmillan Basingstoke, 2005.

[8] PM Solutions: *Selecting a Software Development Life Cycle (SDLC) Methodology, A Practical Decision Framework to Maximize Business Value (The D3 Cube)*, Inc. & Park Hill Technologies LLC, 2005.

Contact Dr. Solomon Bhunu, Corporate GIS, Strategic Development Information and GIS Department City of Cape Town, Tel 021 487-2264, solomon.bhunu@capetown.gov.za