December 1998

The Systems Development Lifecycle in Practice

Lynley Hocking
University of Tasmania

The Systems Development Lifecycle in Practice

Lynley Hocking
University of Tasmania

Numerous writers have identified problems with the commonly used systems development lifecycle. Yet it is still commonly used in practice. Here it will be illustrated that, even if developers formally use the model, the activities associated a project do not seem to align particularly well with this popular model.

The Systems Development Lifecycle Model

Change is often described in a linear fashion where the process is illustrated by a series of identifiable steps and with systems development, such linear change models have become known as the systems development life cycle (SDLC). As Friedman (1989) describes it, the lifecycle is "an ordered set of activities which combine to make up the conception, development, use and eventual replacement of new computer-based systems" (p 175). Derived from operations management, systems and management science literature, these models show a linear sequence of events to be followed, the main advantage being that the process can be standardised and well-defined (Eliason 1990: p 175). Here the systems development lifecycle model is described, critically evaluated and its use in practice analysed.

The lifecycle model does have strengths. It allows the overall task to be planned and subdivided so that the process is more controllable and, as it is predetermined, everyone understands the sequence of events so communication is improved (Nandakumar 1993; Avison and Fitzgerald 1995). Avison and Fitzgerald (1995) argue there is intrinsically nothing wrong with the lifecycle model and much depends on the way it is used, commenting it needs to be sufficiently resourced and any deviations noted and controlled early. It should not be seen as a rigid process, but a flexible and iterative one.

Despite its wide use and the proposed advantages, the lifecycle model is recognised to have some serious and fundamental problems (Eason 1988, Frenzel 1996, Krogstie 1995, Lewis 1994, Nandakumar 1993, Siddiqi and Shekaran 1996, Truex 1993). However, the systems development lifecycle is almost generally accepted as the norm for considering the systems development process. Most systems development texts employ it in some form or another.[1] The model is used by Avison and Fitzgerald (1995) as a framework by which to judge methodologies. The ISO 9001 standards for developing and supplying software systems views the process as a broadly linear one, with the process beginning with the definition of functional requirements, programming, inspection and testing, and delivery and installation. Such international standards are used as a benchmark for assessing commercial methodologies and the process of systems development as it unfolds in organisations. Most "standard" methodologies, such as SSADM, are based on basic variations of this waterfall model (Krogstie 1995).

The Lifecycle in Practice: ACase Study

Reflecting common practices, the systems developers in the project described here used the SDLC to conceptually describe their working processes. The commercial systems development methodology they used was expressly based on the approach, though it only claimed to provide tools which could be used throughout the lifecycle of systems development, and the systems developers used it in this way. The project management metho-dology was also based on the lifecycle approach. Formally, the project team adhered to these models, a review stating the pro-ject management methodology was employed to guide team struc-ture, stakeholder identification and management and the manage-ment of suppliers while the systems development methodology was used in early and later stages of the project. The detailed design stages, outsourced to the technical systems contractors, was also covered by a similar methodology, also based on the SDLC. The project leader's generic conception of the process is illustrated in Figure 1 and is reflective of the SDLC model. This was only one of many process models articulated, but most of them reflected the SDLC in some form.

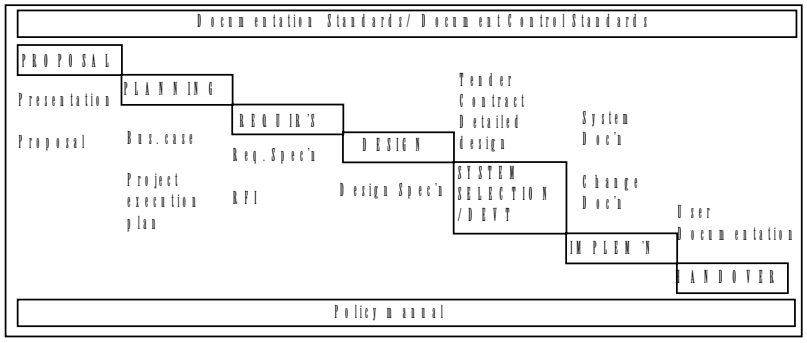

Figure 1. Generic Model of Process by Case Study Project Leader

Using only available documentation to gain an insight into the process proved problematic. To gain a deeper understanding, topics that were discussed during regular meetings

[1]Of the 25 texts on systems analysis/development surveyed, 21 used the SDLC as a framework.

throughout the project were compared with the systems developer's Gantt and other charts. Regular staff meetings, project management meetings, steering committee meetings and project directors meetings were held throughout all or part of the project and the topics discussed at such meetings give an indication of activities at that time. Fortnightly progress reports also gave an indication of activities and concerns at the time. Figure 2 maps activities obviously related to phases in Figure 1.

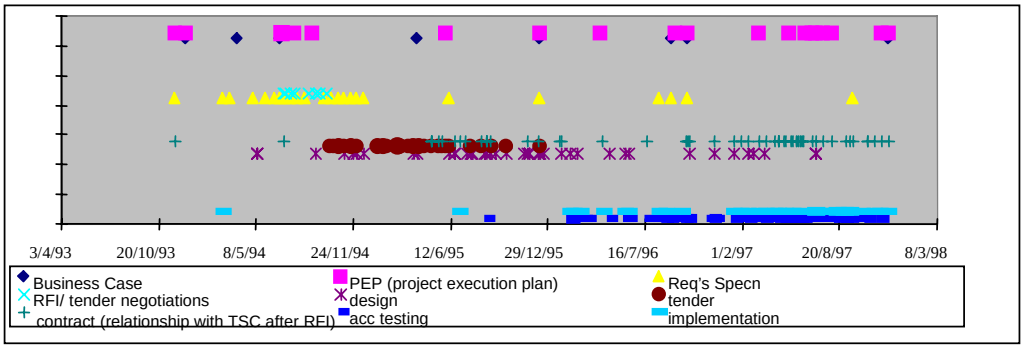

Figure 2. The SDLC in Practice

Broadly, it suggests that the formal model employed do reflect the actions of those involved in the project, but that there was even a greater degree of overlap than suggested in the SDLC model. For example, the scope of the project was discussed on several occasions while the functional requirements were being developed and it, the business case document and the strategy for developing the system were occasionally raised until just before implementation activities were being discussed. Nevertheless, Figure 2 provides a shadow of a SDLC in practice, with the waterfall falling from the top left hand corner forwards. Yet these activities only form no more than 32% of topics under discussion at these regular meetings. Other issues discussed include the acceptance of the project or system by the users, change management (change facilitation), users' lack of knowledge, training the development and implementation of interim processes such as camera-ready processed, relationships with the technical systems developers, delays, related projects and so forth. Many of these topics can be viewed as associated with particular phases of the SDLC model. Planning activities could also include issues of delays, for delays entail that plans have to be changed. Change requests could be considered as part of requirements analysis, as it suggests that requirements are still being determined while "design" could also include the purchase of the technology required. Other tasks which are difficult to identify with phases of the SDLC include issues of standardisation, quality issues, the development and the implementation of interim processes, prototyping and stakeholder involvement. Their inclusion in Figure 3, along with the other tasks, gives a better insight into how the process actually unfolded.

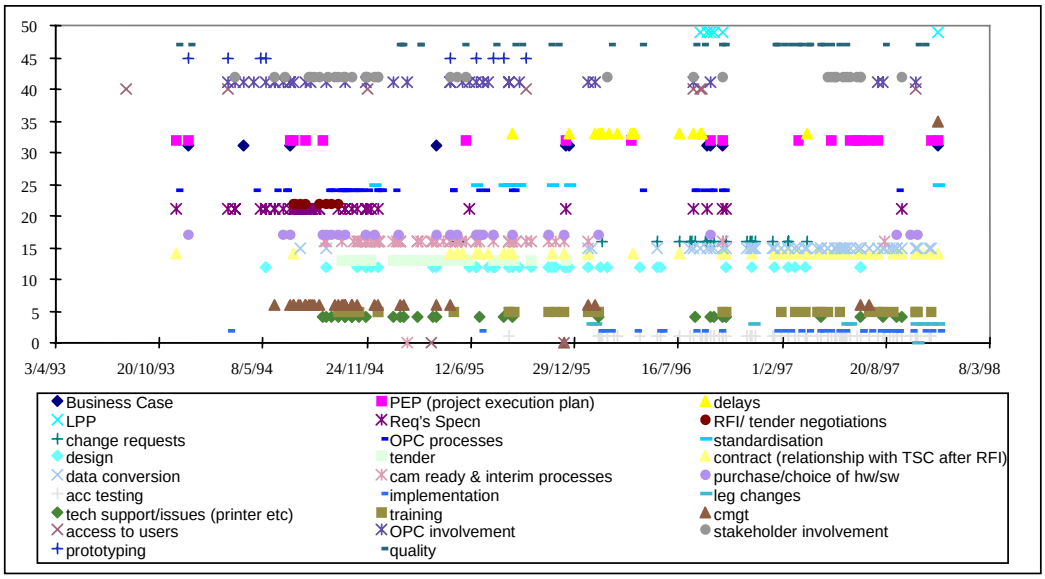

Figure 3. All Topics Discussed in Meetings Throughout the Project Against the SDLC Model Employed

When we consider all these other issues discussed in regular meetings during the project, the overall process becomes more complex, untidy and emergent than either the SDLC or project's Gantt charts would suggest. Implementa-tion issues are being considered while the system is being planned, even if they are not termed as such. Plans must change as circumstances demand it and planning is a continual activity. Generally, the division between planning, design and implementation becomes blurred, while the SDLC aims to keep them distinct and separate.

Discussion

The confusion in the profusion of models used by the project team seems to reflect a balancing act as the systems developers adopted the common broadly linear models of the systems development literature, but had to adapt it to the local circumstances and recognise the emergent and inherently messy nature of planning and implementing change. That change was emergent is

reflected in the many changes to the development strategy and the number of different models used. The variety of models and changes in how the process was formally conceived seems to be a result of a balancing act, as the project team tried to fit the messy, constantly changing reality into formal linear models and a need to reflect a rationalistic managerial approach. Aiming to adhere to recognised best practice, they adopted standard approaches to the process but were not able to strictly adhere to them and operate effectively in their organisational situation.

Westrup (1995) suggests that systems developers can exhibit a much greater degree of organisational awareness than their models and methodologies would imply. Observations of the actions and models used in the case study supports this stance, specifically in relation to how the overall process of systems development is conceived. The normative literature tends to emphasise that phases need to be distinct, and even that they are strictly sequential. Formally, those involved in the project were aligned with these principles broadly, at least to a degree, but when one looks at the issues actually discussed at the time, reality emerges as being far more complex and messy. On the surface, observations of how people involved in the project viewed the process of development were confusing because they were inconsistent and conflicting. However, when one looks at the actual issues and when they were discussed in relation to the models employed, one starts to suspect that the models employed are not adequately describing what is actually occurring and that those involved in such projects are also having trouble aligning their actions to such models and an alternative process model which reflects the emergent nature of the process is needed.

The lifecycle approach is sometimes referred to as the "waterfall" model. Iterations are recognised, in much the same way that salmon ladders are built into dams. Change flows like water down a river, from step to step, from one deliverable to another. Iterations are movements against the flow of change in which we usually drift, but are necessary in order to navigate the right path. The flow down the river is illustrated as a series of arrows from one stage to the next, but how and why we move between them is rarely examined. This paper describes part of a research project aiming to produce such a model of process and consider how it is reflected in normative models.

Conclusions

Despite its deficiencies, the commonly employed lifecycle model is often treated as an objective fact (Lewis 1993). The developers were thus not unusual in initially and formally adopting the SDLC model when thinking about the broad process of development. Yet increasingly in the case study, the model did not match the reality of how the project unfolded. This analysis of the systems development lifecycle in practice suggests that the commonly used process model does not adequately reflect the process as it unfolds in reality. It suggests the process is far more emergent than the lifecycle model suggests and that other process models should be investigated. These implications are pursued in the larger research project and further details can be obtained from the author.

References

References available upon request from the author (lynley.hocking@utas.edu.au).