Process Tailoring in Iterative Development

John Murdoch, John A. Mc Dermid
High Integrity Systems Engineering Group
University of York, York, YO10 5DD UK

Philip Wilkinson
Rolls Royce plc, PO Box 31, Derby,
DE24 8BJ UK

This paper proposes a dynamic or variable approach to the tailoring of SE processes which seeks to reconcile the complicated models of SE, which we justifiably develop within the discipline, with the pragmatic needs of system developers. Of particular concern is process definition at product inception and in the innovative parts of well-defined projects, where SE expertise is needed but at low cost and in the absence of specific development experience. We reference current trends in the management literature (Ghoshal & Bartlett 1997), in software development (Kruchten 2000, Weiss & Lai 1999) and in role-based process modelling, building on earlier papers (Murdoch et al. 1998, 1999). The following sections discuss the organisational context in which SE is practised and the typical staged development cycles used by projects. Role-based models are reviewed and then used to describe the proposed approach, using examples. A final section discusses some issues arising from attempts to apply the approach in a practical setting.

ORGANISATIONAL CONTEXT

The organisational context in which systems engineering is practised (or being introduced) has been undergoing significant change over the last decade and more. One trend is a loss of confidence in strategic planning (Mintzberg 1994) and top-down decision-making. Organisations are becoming increasingly concerned with the flexibility of their processes. Fundamental issues of responsiveness to innovation in product concept, system architecture, component technologies and inter-working relationships are of growing importance. Current developments in management thinking and in e-business technologies are also pushing in the direction of flexibility and adaptability.

This paper addresses the tailoring of systems engineering processes to iterative product development, using a role/responsibility-based approach. Two levels of tailoring are discussed: (1) from standards and the systems engineering literature to an organisation's capability resources and (2) from the capability resources to an individual project. The approach is motivated by the wish to apply standard reference models developed by the systems engineering community in a flexible way, especially to link and scale up innovative and entrepreneurial activity into projects. Simple examples are provided to illustrate the approach and some issues involved in a practical application are discussed.

INTRODUCTION

The systems engineering (SE) discipline is well described in texts (e.g. Martin 1997, Lacy 1992, Stevens et al. 1998), courses and standards (IEEE1220, EIA632, ISO15288). Much of the knowledge developed within the discipline is expressed in the form of reference process models, extended to include product and information modelling in the model-based approach (Oliver 1998). It is natural to strive for completeness in such models in terms of tasks to be performed, documents to be written, information models to be populated etc. Repeated application of tasks at several levels of decomposition/aggregation may be needed for large systems. When applied, such reference models are tailored to the needs of the particular organisation and project. In a further development, the Capability Maturity Model approach has been applied to SE, effectively representing the quality engineering values of repeatability and continuous improvement in SE practice.
making hierarchies. For example, (Ghoshal & Bartlett 1997) view an organisation as a set of three kinds of role, namely front-line Entrepreneurs, middle-level Capability Developers & coaches and top-level Institution Builders (Figure 1). Here, the terms front-line, middle and top refer to where the roles might be located in a classic hierarchical organisation. This management approach involves the driving down of responsibility to ‘lower levels’ of the organisation, in an attempt to recover the small-company spirit in large organisations. Although the Entrepreneurs are supported by the Capability Developers, they are autonomous, not merely implementing a strategy imposed from above.

We use this approach as a way of thinking about the application of SE within an organisation (Figure 2). The System Engineer (or System Architect) is viewed as a mainly autonomous agent participating in a product development activity. The System Engineer role would be defined in terms of the responsibilities to a project. An individual enacting this role will have specified knowledge, experience and qualifications and will be supported by the SE capability resources within the company. Several questions arise: who should define the objectives of the System Engineer?; who should conduct the tailoring of reference process models?; and how prescriptive should the organisation be with respect to the activities of the System Engineer?

In the approach proposed here, responsibility for choosing how best to apply SE discipline knowledge and the SE capability of the company is placed in the hands of the assigned System Engineer. This is a concept of capability pull rather than procedural push. One implication is that two levels of tailoring are involved; (a) the tailoring of generic models from the academic and professional SE communities into the company’s capability resources and (b) the tailoring of the capability resources to a particular project. Feedback of experience is involved in each case.

With regard to the prescriptive-ness of SE activity, it seems necessary to have a variable approach. Process models (in the form of task structures) are fundamentally plans of action. The success of a plan is dependent on the predictability of the environment in which it is to be executed. We would expect investment in detailed process models by the Capability Developers to be justifiable for repeat design projects with stable product architecture, technologies and supplier arrangements. Family-based product development is one approach to limiting variability between projects and increasing the return on capability investment. Regulatory constraints may be applicable which require documented and review-able procedures.

For unprecedented activity, detailed task structures will have to be defined by the project, as product definition unfolds. An organisation can create an environment which fosters creativity and product innovation, but it cannot develop detailed plans to

Figure 2. Organisational context in which SE is practised
make it happen. However, once innovative concepts or entrepreneurial opportunities appear, the organisation can ready itself to develop them into successful projects. The spiral process models of iterative development provide a management approach in such cases.

**STAGED DEVELOPMENT CYCLE MODELS**

The classic, sequential staged development model is appropriate if the product to be developed is of known architecture and the unknown areas of the design are low-level or localised. (Blanchard 1998) has identified SE tasks and associated them with development stages. The SE capability will then provide a staged development model with tasks, documentation, stage gate criteria etc., providing a detailed template for a new project to pick up. The *System Engineer* will be expected to apply the reference model, departures from it being approved at a review gate.

Products or components with significant unknowns use an iterative or spiral development cycle (Figure 3). (Rechtin 1997) and (Stevens et al. 1998) have discussed the spiral model as applied to systems engineering. Iterative models are often used in software development, a recent example being the Rational Unified Process (Kruchten 2000).

With iterative development, processes are designed as the product definition unfolds. The reference process models in the capability resources would be tailored to the needs of the project at each iteration. Such tailoring would reflect the architecture of the product, the technologies involved, the measures of effectiveness which need to be tracked and other issues.

As well as accumulating design data and costs from cycle to cycle, the spiral model also results in an accumulation of *roles*. (Martin 1997) has discussed the identification of Integrated Project Teams (IPTs) with components of the system architecture, mirroring the system parts breakdown tree. The role concept is central to process tailoring discussed below.

**ROLE-BASED PROCESS DEFINITION**

In general, organisational processes involve the following things:

1. *Purpose*, objectives, goals;
2. *Organisational structure*, including roles;
3. *Activities* (human and automated), assigned to roles;
4. *Inputs* and *outputs* (non-data and data) of activities and roles;
5. *Events* which trigger and modify activities;
6. *Resources* required by roles to enact activities.

Data forms the most important part of the inputs, outputs and resources of many processes, particularly of design processes.

In the unprecedented situation, a process model is grown from one iteration of the spiral model to the next. Eventually all of the above aspects of the final development iteration will need to be defined, with the full machinery of project management used. The question addressed here is how to represent processes in the early iterations in order to support scaling up to full project definition (and the linking with known activity structures for familiar parts of systems).

Process tailoring (of the second type mentioned above) is taken to mean the review of a given task structure and its adaptation to meet the needs of a particular project situation. This is carried out by a *System Engineer* or other defined role in the project. It is implied that, before tasks are defined, a project role with defined responsibilities or objectives has been defined. (An objective or responsibility can always be expressed as a top-level task; here we are concerned with task breakdown to achieve the top-level objectives.) Trust is placed in the individual enacting the role, to perform the tailoring and define suitable future activity.

In order to visualise processes without tasks, a process notation which includes the role concept is needed. This paper uses *Role Context Charts* as proposed in a previous paper (Murdoch et al. 1998). A role is drawn as a box with rounded corners and interconnecting arcs represent interactions between

**Figure 3. Spiral model, this version from (Stevens et al. 1998)***
First Iteration

roles. Directed lines may be used to indicate flows (often of information) from one role to another. Roles may be grouped together with dashed contours, indicating teams within which close interaction is expected. Such a chart shows the roles involved in a defined process, or part of a process, over a specified time interval, for example a stage or iteration in a development model. These charts can be mapped to Role Activity Diagrams (Ould 1995) which provide a means of visualising threads of activity enacted by the roles and interactions between them.

(Stevens et al. 1998) have distinguished between Customer and Supplier roles in the application of SE process models and developed a visualisation of the tiers of concurrent development. Other notations which encompass the concept of role are also applicable, for example the Unified Modeling Language (Fowler 1997) used in the description of the Rational Unified Process model (Kruchten 2000), developed in the context of object-oriented software engineering. The outputs would be stipulated, although advisory material and coaching would be available from the Capability Developers. Depending on decisions taken by the System Engineer, this approach may generate activity which is not reviewable and may not exploit or generate any formalised procedural knowledge. The work of the System Engineer would be judged as part of the outcome of the embryonic project as a whole, at the following review gate.

Clearly this approach would be inconsistent with process improvement and organisational learning concepts if adopted throughout a project. There is no provision for feedback of lessons learnt to the Capability Developers, nor support for co-ordination of work, project planning or audit. However, it provides a route for injecting systems engineering expertise directly into a project with minimum 'bureaucracy'. Capability development, reviewability and conformance with organisational reference processes are sacrificed in the interests of ensuring SE presence at critical early stages of product innovation.

We can think of this as being the first iteration of the spiral model of Figure 3. Perhaps just the roles of Figure 4 would be involved. The review at Gate 1 would include detailed plans for the next iteration and outline plans for subsequent development. The new product idea would be assessed with respect to the level of innovation involved, the degree of similarity to existing or past products etc. Subsequent development cycle decisions would be made e.g. adopt a staged process or iterate further (other variants have been identified by (Martin 1997)).

Following a successful review, the next iteration would be likely to involve more staff and perhaps suppliers and a potential customer. This might include a single System Architect or several individual and team roles to reflect a proposed product architectural breakdown. The process definition unfolds as the product becomes better defined and make/buy decisions are made. At later iterations, more of the reference models etc. in the capability resources will likely be applied. This amounts to a variable tailoring, from a minimum set of constraints at inception, to increasing use of organisational best practice as the project grows.

Figure 5. Role creation following system decomposition

PASTA (Process and Artifact State Transition Abstraction) approach to process modelling, developed to support software product families (Weiss 1999) and the ARIS modelling approach, developed by (Schêer 1998) in the context of industrial processes, include similar notations.

UNFOLDING OF PROCESS DEFINITION

The previous sections have discussed the concept of an iterative process, in which the roles enacting it define the next iteration, informed by the particular engineering design decisions taken and by organisational capability. The tailoring of reference process models is dynamic in the sense that it is different from one iteration to the next. This section sketches the unfolding of a process in order to provide an illustration.

Suppose a new product concept has been proposed and an initial exploratory development agreed. The organisation wishes to implement a SE approach from the first exploratory work. The minimum requirement that an organisation can place on the new product effort is that a System Engineer or Architect be part of the product development team. No specific activities or
Figure 5 shows an example design step in which a system is defined as comprising three Subsystems, with Subsystem A being composed of Units U1 and U2. Roles are defined as having design responsibility for each item at each level. Based on the particular features or risks of the system, additional roles may be identified for safety assessment and for interface definition between U1 and Subsystem C, for example. Specialist roles (for example, Software Engineer) may be introduced into a team if design or manufacture risks indicate that early assessment and planning are required. Roles may be defined as having measures of effectiveness, for example as defined by Oliver (1997). A particular role may participate in one or more teams.

The fan-out decomposition implied by Figure 5 is not always applicable. In many systems, lower-level units or subsystems are shared by several higher-level subsystems. A computer-based system, for example, may provide a range of control functions, an important example of a shared resource. If this indicates a possible approach to role identification and event handling, then the responsibility allocation to roles. As a project grows, the co-ordination of work will become more important, calling on the disciplines of project management, estimation and control etc; the process models developed in the earlier stages/cycles will form the technical input to these activities.

Reference models adapted and recorded by the Capability Developers are used to inform the unfolding process definition. Reference roles and activities will reflect the experience of previous projects, including product architecture and organisation/supplier arrangements. Integration of the innovative activity into larger projects and re-used system architectures would be supported. It may be decided to procure a Unit or Component from a supplier rather than manufacture in-house. A spectrum of possibilities exists, ranging from commercial-off-the-shelf to bespoke procurement.

The allocation of tasks to roles, their ordering, the identification of required interaction between task and controlling events may be defined following on from the responsibility allocation to roles. As a project grows, the co-ordination of work will become more important, calling on the disciplines of project management, estimation and control etc; the process models developed in the earlier stages/cycles will form the technical input to these activities.

Reference models adapted and recorded by the Capability Developers are used to inform the unfolding process definition. Reference roles and activities will reflect the experience of previous projects, including product architecture and organisation/supplier arrangements. Integration of the innovative activity into larger projects and re-used system architectures would be supported. It may be decided to procure a Unit or Component from a supplier rather than manufacture in-house. A spectrum of possibilities exists, ranging from commercial-off-the-shelf to bespoke procurement.

Reference models adapted and recorded by the Capability Developers are used to inform the unfolding process definition. Reference roles and activities will reflect the experience of previous projects, including product architecture and organisation/supplier arrangements. Integration of the innovative activity into larger projects and re-used system architectures would be supported. It may be decided to procure a Unit or Component from a supplier rather than manufacture in-house. A spectrum of possibilities exists, ranging from commercial-off-the-shelf to bespoke procurement.
In later iterations, the flavour of the SE processes will change from early engineering / creative aspects to more integrative technical management issues. Process modelling would spawn roles responsible for, for example, operational concept, maintenance concept development, logistics design etc.

**TRUST AND DEVELOPMENT ASSURANCE**

Fundamental to this approach is that trust be placed in the development team to pull product and process information appropriately from available capability resources. This amounts to a requirement that systems engineering be treated as an area of professional engineering practice, with professional judgement being at the core of technical decision making. Issues of professional certification of individuals and team capabilities are relevant here.

What checks should the organisation make that the development team is acting responsibly? The phase or iteration gate reviews should include review of the planning for subsequent development. This is a form of variable development assurance. Instead of checking work against pre-defined task structures, the assurance review would check plans against underlying assurance principles, interpreted flexibly. The level of assurance/audit activity imposed by the organisation would be designed to reflect the investment and risk to other stakeholders arising from the growing project. In the case of safety-critical systems, governmental regulatory audits exist in many industries. Current aerospace guidelines include the concept of a variable Development Assurance Level linked to risk level. For companies operating in such environments, internal reviews can check for regulatory compliance, or at least for plans to comply at some later development cycle.

**INDUSTRIAL APPLICATION**

The conceptual approach described here is being applied in ongoing work with the Controls business unit of Rolls-Royce, Derby UK. The development of aero-engine controllers exhibits many of the characteristics of systems engineering, namely complexity of product and development process, a need for multi-disciplinary skills and a concern with key total system performance attributes (reliability, safety, support of whole engine performance).

Several process improvement initiatives are being pursued by the company, mostly driven by business objectives (reduced non-recurring costs, reduced time-to-market). Much legacy procedural documentation is being discarded, being replaced with intranet-based publishing of new guidelines. Application of the concepts in this paper is influencing these developments. We have found that role/responsibility assignments have added clarity to reference process models in general and have provided a vehicle for discussing the needs for SE roles at different indenture levels. We would argue that SE or System Architect roles embody an engineering design orientation, helping to balance tendencies in system integrator companies to emphasise procurement and requirements management views.

Aeroengine controller architectures are currently fairly stable, justifying the development of reference role structures. Role allocation on a particular project can then focus on areas of variability, for example in detailed function allocation between aeroengine controller and airframe systems.

The flexibility aspects of the approach discussed are mainly applicable to the software components of the control systems, in the particular situation studied. Although this strictly lies outside the systems engineering domain, similar principles apply. The control software is usually developed afresh for each project, with limited re-use, leading to high development and certification costs. Software engineers strive to develop a family-based approach to software development. In an effort to reduce costs, there is a tendency in the software field to emphasise automation in software development, implying investment in re-usable code and specifications. This is quite difficult in the controls domain because of the variability in detailed requirements and the fact that software requirements are largely determined by other design processes. The approach of this paper is being applied to develop a solution which exploits a common software architecture and some re-use of common components but ultimately is based on flexible...
role/responsibility allocation. While modelling and automated tools are part of the organisational capability, emphasis is placed on developing engineering understanding of software requirements and fostering negotiation with processes at higher indenture levels.

CONCLUSIONS

The task of tailoring SE reference process models to the needs of a project has been addressed using a role identification approach. Two levels of tailoring have been identified (1) from standards and the literature to an organisation's capability resources and (2) from the capability resources to a particular project. Tailoring for a project has been placed under the responsibility of the System Engineer or Architect assigned to the project, who uses the capability resources as guidance and support. The concept is one of a project pulling solutions from the organisation, rather than having set procedures imposed. Role identification has been discussed as an unfolding process, driven by the product design. Project management and development assurance machinery is developed from the role responsibilities and the interactions and tasks developed for them, driven by the engineering needs. A simple illustrative example has been discussed and some practical experience reported.

Current developments in management and e-business technologies call for increased flexibility in processes, but with clear responsibility allocation and co-ordination of activities. The approach proposed in this paper seeks to reconcile the elaborate reference models developed in the SE discipline with the increasingly dynamic and distributed organisational contexts in which SE is practised.

REFERENCES


BIOGRAPHIES

Dr John Murdoch, High Integrity Systems Engineering Group, Dept. Computer Science, University of York, York YO10 5DD UK, telephone - (44) 190 443-4728, facsimile - (44) 190 443 2708, e-mail - murdoch@cs.york.ac.uk John Murdoch has worked for over fifteen years in the aerospace industry, in a variety of specialist, systems engineering and R&D roles. Currently, he is with the Rolls-Royce University Technology Centre, University of York UK, with interests in process and product modelling for high integrity systems.

Professor John A McDermaid, High Integrity Systems Engineering Group, Dept. Computer Science, University of York, York YO10 5DD UK, telephone - (44) 190 443-2726, facsimile - (44) 190 443 2708, e-mail - jam@cs.york.ac.uk John McDermid is Professor of Software Engineering and Head of the High Integrity Systems Group within the Computer Science Department of the University of York, UK. He also heads the Rolls-Royce Systems and Software University Technology Centre at York. His research interests are in safety critical systems and software, especially in aerospace applications.

Philip Wilkinson, Rolls-Royce plc, PO Box 31, Derby
Phil Wilkinson has been with Rolls Royce for 15 years. He has worked in a range of specialist safety, design and management roles in Control Systems and is currently Group Manager of the Systems, Safety and Reliability Group.