

Designing a process

INTRODUCTION

So far, we have been dealing with the situation where there is a process out there and we want to unearth it, understand it, communicate it, standardise it, analyse it, or improve it. Or several of those. In this chapter we look at how we can use *Riva* in a green-field situation where there are no processes at all, or there are some processes and we want to add one or more new ones. In some extreme cases we may have a process in place but it is so awful we have decided not to even look at it and to replace it from scratch.

For simplicity, we shall assume from here on that we have a truly green-field site. The other situations can be dealt with by appropriate adjustments to the green-field approach.

As with process improvement in Chapter 10, this is not a book about running a process development *programme*, more a book about how *Riva*'s concepts can be exploited in such a programme. So we shall touch on the programme aspects but only to connect *Riva* into them. (For that larger context, you might look at Roger T Burlton's *Business Process Management – Profiting from Process*, SAMS, 2001.)

Here's an overall plan of our process design project as it might appear in the larger process programme:

- 1 Prepare a process architecture.
- 2 For each process we want to design:
 - a. Set the ground-rules for that process.
 - b. Run a process design workshop.
 - c. Validate the design.
- 3 Revisit the process architecture to ensure that the fit we decided on at the outset has worked.

PREPARING THE PROCESS ARCHITECTURE

This should no longer be a surprise: whatever the situation, we shall start with a process architecture. If we have a new organisation – be it a company or department or whatever – we shall go through the procedure described in Chapter 6: characterise the organisation through its EBEs, filter those down to UOWs, establish the dynamic 'generates' relationships between the UOWs, translate the resulting UOW diagram into the first-cut process architecture, and reduce that to give the second-cut process architecture. Initially, there will be no designed UOWs, and hence all the processes – case, case management, and case strategy – will be to

11 – DESIGNING A PROCESS

do with the things that are the essence of the new organisation. As we start to design those processes, we might decide to create new, designed UOWs that will be part of the way we decide to do the business of the new organisation. These will invoke new case, case management, and case strategy processes which we can also design.

GETTING THE PROCESS DESIGN GROUND-RULES IN PLACE

With the process architecture in hand, we can confidently work on each of the processes. Of course, we can't expect to knock them down one by one, sequentially. But in what follows we shall assume that we are addressing one process in isolation.

We must be certain we are headed in the right direction with the right terms of reference. We shall ask the following questions:

- ☞ Are there any *organisational givens*, things we cannot change?
- ☞ What are the *requirements* for the process?
- ☞ What are the *principles* for the process?

To answer these we must question the relevant stakeholders: the process owners, managers, customers and sponsors. How we ask them will be a matter of choice: a workshop, a round-table meeting, or separate interviews, for instance. However we go about it, we must document the ground-rules we discover and get agreement – formal or informal as the occasion demands – from those who matter.

Are there organisational givens?

Are there any bounds to our ability to change things? For example:

- ☞ Are we constrained by the current organisational structure?
Must the process work within the current organisational structure as it is, with the current roles and responsibilities? Are existing roles and responsibilities up for grabs? Are some roles fixed and others open to change? Can we create new organisational units, posts, and job titles?
- ☞ Does our IS or IT infrastructure impose constraints?
Are we restricted in what we can expect in the way of person-to-person communications, support for workflow and job scheduling, availability of information, or production and distribution of documents in all their forms?
- ☞ What constraints do the regulators place on us?
Are there specific requirements for reporting from our processes? Are there required levels of transparency? Which areas of activity must be open to independent audit? Are there prescribed interactions with the regulatory body? Are there things that must be approved, or licensed before we can proceed at certain points?
- ☞ Are there safety, security, or risk issues to be addressed?
Do we need independent roles with responsibility in these areas? What interactions are prescribed for them? Are there prescribed procedures that we must incorporate in our own processes? Are there prescribed processes with which we must interact?

11 – DESIGNING A PROCESS

☞ Are there cultural norms that must be observed?

Are we an empowered or a controlled organisation? What behaviours do we expect from our staff and our managers that might steer us towards certain styles of interaction or relationship?

What are the process requirements?

The process architecture proves itself useful again at this point. Simply by chunking the organisational activity the way it does, it gives us a basic level of chunking of the requirements on the organisation across the processes. If a process is called **Handle a regulatory request**, or **Produce a product batch**, or **Handle a customer**, or **Manage the flow of products**, or **Manage the flow of orders**, we already have a clear focus. Modelling methods that allow arbitrary decomposition leave us with arbitrary chunks and no clear focus.

Pupil: I hesitate to suggest this, but isn't this an ideal place to think in terms of inputs and outputs? Can't we simply express our requirements for the process in terms of what goes in and what goes out?

Tutor: Firstly, let's remind ourselves that we are talking about designing case processes and case management processes. A neat end-to-end process like a case process might suggest an input-output approach. But you must admit that it is less likely to make sense for a multi-threaded thing like a case management process?

Pupil: Well, possibly. I'll suspend my disbelief for a moment – I'm happy to concentrate first on the case process – the argument will be stronger there.

Tutor: OK. Let's take **Handle a regulatory request** as an example. It's pretty clear what the input is. What about **Produce a product batch**? We know from its name what the output is. If you remember, we based our naming convention that way. But I can imagine a whole pile of other things that we shall require of a process that we would be hard-pressed to express in terms of inputs or outputs. Here's a few to be getting on with: *Customer has been notified that their complaint has been rejected, Management have been informed of price, Batch is in regulated store correctly labelled, Marketing are satisfied with the timing.*

Pupil: I see. It's often not just the thing we want to identify – complaint rejection, price information, management, batch, and so on – but its state.

Tutor: Exactly. By working with final states, we can be more general and we're not forced to contort everything into the strait-jacket of outputs. We can make the same case for working with initial states instead of inputs.

There is another danger we need to be careful of: that of writing process design, when we should be writing process requirements. For instance, if we write 'The customer must receive a letter of apology' we might have made a design decision: the apology will take the form of a letter.

Pupil: You're saying in effect that we need to work with an abstract model?

11 – DESIGNING A PROCESS

Tutor: Yes, I am.

The requirements on a process

When we draw up the requirements for a process, we shall be looking at it from several angles:

⇨ **The starting state.**

What will be the state of affairs when the process starts? There might be several possible starting places – triggers – and we shall need to cover each; this is especially true of a case management process. We can think in terms of how we would write the state description at the beginning of the thread on the RAD, whether for a case process or a case management process:

Approved plan in hand.

Customer call is switched to the appropriate desk.

Specification for batch has been approved but resources not yet allocated.

Project definition available in Project Inception Document.

A request has been received for a new clinical trial supply to be made.

A project has reported an over-run in budget.

⇨ **Any actions required in the process.**

Perhaps certain checks are required, or certain quality controls must be carried out:

All intermediate products must be tested for specified purity.

The unit must be shown to have achieved greater than 99.9% reliability.

All materials leaving the plant must be appropriately labelled.

⇨ **Any business rules governing the process.**

Perhaps certain financial oversight is required, or certain people must be involved in certain decisions or actions. The relationship between a case process and its case management process will appear here. And some rules are what we earlier called 'steady-state goals'. For instance:

Product Quality Assurance must approve all test plans.

Divisional Management must be kept apprised of progress.

Resources will be allocated by the Production Planning Team.

Marketing need to be satisfied with the timing.

All pre-customer checks must be carried out by an independent testing group.

Any type 5 expenditure must be approved by the Divisional Director.

All materials supporting the decision are to be retained.

11 – DESIGNING A PROCESS

The Customer must be kept up-to-date at major milestones in the progress of their application.

☞ The required outcomes of the process.

For a case management process, we can ask what the possible outcomes are for each of the identified triggers. For a case process, we can ask what state we want things to be in when the case has been handled, i.e. the process has completed. We might recognise several alternatives: success or failure; approved or not approved or returned for rework, and so on. We might describe those alternative states in terms of things that have been produced or changed – but only if those things are necessary and of the essence. We can think in terms of how we would write the state description somewhere on the RAD:

Approved plan in hand, management informed of content, and projects database updated accordingly.

Customer complaint rejected.

Batch of chemical in regulated store correctly labelled.

Customer has agreed that their call has been sufficiently answered.

What are the process principles?

The requirements we were looking at in the preceding section were quite functional: the process must *do* this, that, and the other. But there can also be requirements on the process that are more about what the process *should be like*, how it should feel to use, how it should feel culturally, what organisational norms it must conform to, and so on. These sorts of requirements are also typical of an improvement situation: we know what the process should do – we now want to change its feel.

Let's take an example. A large organisation, with many thousands of computer users, had a large infrastructure services group, itself numbering hundreds of staff. Running a company's networks and computers is a thankless task at the best of times: people are rarely satisfied with the PC on the desk in front of them and the facilities behind it. This group was going through a particularly bad patch and relationships with the user community had reached a low. The perception was that the constant battering from the user community had left the group with a defensiveness that was not in the business's interest, but that had become built into its processes. If the relationship was to be improved, the processes had to be redesigned to meet the expectations of customers better.

Getting to the root problem of the perception was hard. But a long session with a group of users, using brainstorming and affinity diagramming, allowed us to reduce the many gripes to a handful of failed expectations about the way that service request calls to the group were handled. Turning these around we were able to summarise a set of process principles that would lead to more agreeable experiences for users and that should therefore govern the process redesign. When you read them, you will find them very obvious – but then these things always are:

When I make a service call don't bounce me from person to person.

When we discuss my requirements, set my expectations at a reasonable level.

11 – DESIGNING A PROCESS

Don't say something to please me and, in the event, let me down again.

Be honest with me about timescales.

Get back to me if you say you will.

These straightforward principles captured what was lacking in the processes as they had ended up. We put them on the wall at the subsequent process re-design sessions to guide the design ... but more of that in a moment.

It need not be the customers of the process that want to see improvement – we would expect management to look for it too, of course. As another example, let me quote a large software development group inside another multinational. A cultural move was under way to increase responsibility-taking and pro-active problem-solving and to add a measure of risk-taking. Existing processes were being re-designed. A workshop of stakeholding managers, from within the development group and its customers in the business, produced a set of principles that reflected the changes that were sought:

Commitments and commitment points are clear.

Deliverables are clear.

Responsibilities are clear (including for the quality of all deliverables).

The process acknowledges and attacks risk and makes risk visible.

There are clear and appropriate financial control and responsibilities.

There is clear, fair and unambiguous reporting.

Everything is change managed.

It's a collaborative process.

Quality is central to our approach.

As a third example, I'll say a little about a project with a major charity that was changing its administrative processes to take advantage of a new software system being installed. The new system made possible more streamlined processes, so the processes and system were designed in parallel. The *functional* requirements for the system emerged quite naturally as the design proceeded. As part of the preparatory work, the more abstract requirements on the system and the process were identified in sessions with both users (the administrative staff) and management. They covered the need for *confidentiality* about cases (the system administered grants to individuals in need), *speed* (grantees typically had low life expectancy so a protracted grant approval process could prove useless), and *sensitivity* (ensuring that, for instance, the process responded appropriately if the intended grantee died before approval).

Process principles can – of course – conflict. If we have visited the different stakeholders and asked them for their principles, we may easily end up with conflicting pressures.

However we acquire them, these process principles are important enough to be written in large red letters on sheets that are on the walls when the process design workshop is run. Both during and after the workshop, it's our responsibility – as workshop leaders – to be constantly checking that the design is in line with the principles. The principle 'When I make a service

11 – DESIGNING A PROCESS

call don't bounce me from person to person' had clear implications for the way that the roles involved interacted: the person picking up the original call would not be able to relinquish responsibility for it by passing it off to someone else; they would have to remain in the loop and in control. 'Get back to me if you say you will' meant having reliable prompts on the supporting call management system which were acted on. An estimate of time-to-fix would be given to a caller only after enough investigative work had been done and the estimate had been checked: wild guesses were not enough. If an estimate for completion was likely to be too unreliable, then the response would be staged and an estimate given for completion of just the first stage, which might be a visit to the caller's desk, for instance.

KEY POINTS

The process architecture tells us what processes must be designed for a new (part of) an organisation.

Before we start the design of a process we must

- ☞ list the organisational givens, the constraints on our design freedom;
- ☞ list the principles that are to govern the design;
- ☞ list the requirements that the process must fulfil, reconciling conflicts where they arise between different stakeholders in the process.

Requirements for the process should be expressed in terms of

- ☞ its starting state(s);
- ☞ any activities that must take place;
- ☞ any business rules governing it;
- ☞ the required outcome(s).

THE PROCESS DESIGN PROCESS

We are designing a process from scratch. We know from the process architecture which process we are designing. We have established the constraints, principles, and requirements that apply. It's time to run a design workshop.

Preparing for the process workshop

The process architecture tells us the basic boundary of the process:

- ☞ If it is a case process, the initial state will be some variation on *Case in hand*, and there will be one or more potential outcomes, presumably at least one of which will correspond to 'success'.
- ☞ If it is a case management process, we have a list of candidate triggers from Chapter 5. Each of these will have one or more appropriate outcomes.

11 – DESIGNING A PROCESS

- ☞ If it is a case process, we know which units of work are in turn generated and hence which case management processes must be approached to get cases of those units of work under way.

As there is no existing process, we do not do any prior briefing. It will be enough to get the ground-rules in place as described above. We can now run a process workshop along the lines of the generic one described in Chapter 8 – but with some major adjustments for this green-field situation. The sequence of steps will vary from situation to situation but the following will give some structure to the event:

- ☞ From the outset we have flipchart sheets on the wall with all the key inputs:
 - ☞ the (relevant part of) the process architecture,
 - ☞ the name of the process,
 - ☞ a list of candidate triggers,
 - ☞ one or more potential outcomes,
 - ☞ relationships with other processes,
 - ☞ the organisational givens,
 - ☞ the process requirements,
 - ☞ the process principles.
- ☞ Checking the list of organisational givens on the wall, rather than brainstorming existing roles, we brainstorm the necessary areas of responsibility – abstract roles – corresponding to
 - ☞ mandatory or specialised skill sets,
 - ☞ mandatory approval,
 - ☞ mandatory oversight,
 - ☞ mandatory review,
 - ☞ restrictions imposed by IT,
 - ☞ any activities that must be carried out independently.
- ☞ Checking the list of process principles on the wall, we brainstorm further areas of responsibility – abstract roles – corresponding to
 - ☞ desired skill sets,
 - ☞ desired approval,
 - ☞ desired oversight,
 - ☞ desired review.
- ☞ We brainstorm basic actions within the process, allocating them to abstract roles, and noting any ordering that is immediately apparent.
- ☞ For a case process, we brainstorm the sorts of interactions we expect the process to have with its case management process.
- ☞ We identify business-derived decisions and alternatives.

11 – DESIGNING A PROCESS

- ☞ We gradually lace together the abstract roles with interactions that support the collaborative content. Our first model will be an abstract model.
- ☞ We use the process principles to validate the design.
- ☞ We check that all the requirements have been dealt with.
- ☞ We check that the organisational givens have not been contradicted.
- ☞ If appropriate, we then decide how we want to implement the process in terms of concrete roles, concrete mechanisms for interactions, and so on. In going from the abstract to the concrete – i.e. as we cut up the pizza – we shall find ourselves creating new interactions between the concrete roles.
- ☞ We identify how existing or potential information technology solutions could turn abstract things into concrete. (*Tutor: That sounds dangerous, doesn't it?*)

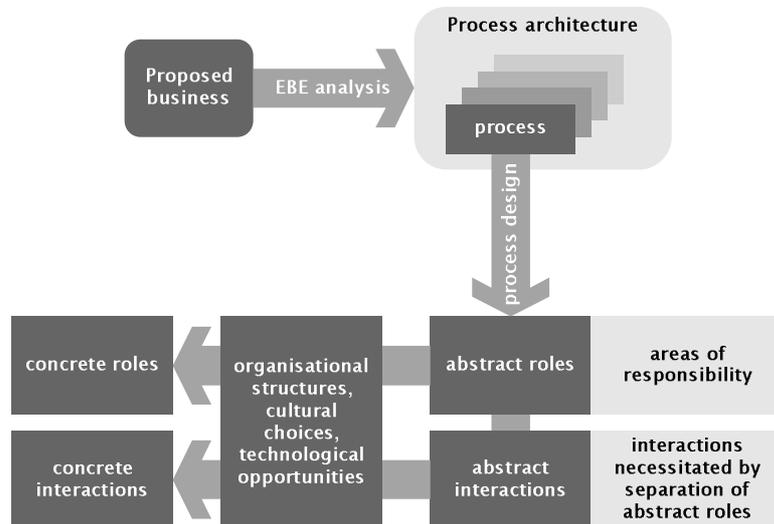
Finally, there is a thread that we left dangling in Chapter 2 and which it is now time to tie off. Suppose we are designing for a steady-state goal such as 'Bank balance is always positive.' We must ask 'What might change the bank balance?' and ensure that only actions that keep the bank balance positive are possible. This might mean, for instance, only allowing spending if a prior check has shown that there are sufficient funds to meet the cost.

Slightly trickier, we might require that 'Marketing are always aware of the status of new product developments.' We perhaps recognise that there will be periods when Marketing haven't heard some detail of a new development and that the best we can do is to minimise those periods. Are there actions or interactions in place to restore that part of the state after process elements that might change it? Where an action can leave Marketing out-of-date, we can ensure there is an immediate interaction to bring them up to date.

11 – DESIGNING A PROCESS

SUMMARY

Figure 11-1 The general scheme for process design



Pupil: You haven't exactly provided a recipe for designing a process.

Tutor: No, and I don't think it's possible to write one. But we have listed what we need to do to generate the information we need to do a design. Design is not a mechanical process – it is all about invention and thinking – it's above all a creative process. The best we can hope to do is to identify what will help us get everything on the table before we start, and check that what we create meets the requirements.

Pupil: I'm also nervous about the fact that we haven't touched on issues such as whether or not the design is acceptable to management, whether it can really be supported by the IT infrastructure, whether it would actually work, whether it would work fast enough.

Tutor: You're right to be concerned about these things, but we did say that all those process programme issues are for other books and other tutors ...