BLACK CIRCLE PTY LTD

ENABLING AGILITY

If your team of developers is stressing and challenged by timelines, consider what you can do to give them a sound and supportive structure to work within.

The Dilemma

In 1993 I was appointed to head up the unit providing development and support for core applications for a Commonwealth agency. New developments and enhancements were behind schedule, and the support backlog was the stuff of local legend. Several dozen people were stressing over constant interruptions, instead of focusing on delivering high quality support and systems to the client areas.

The problem comes down to the fundamental incompatability between support processes and project delivery. What is the time driver on production system support—or any support task, for that matter? NOW. Fix the bug now, provide the help now, because the business is losing money all the time the problem continues.... The bottom line is productivity,now. Yet what is the time driver on project-based work? The negotiated schedule, whenever that is. It may be fixed or negotiable, but it is for a specific point in time, and the bottom line is flexibility in response to an unpredictable future. And therein lies the problem, because responding to the NOW compromises the schedule, and sticking to the schedule compromises the life of the organisation.

A decade and a half later, this is still being reported to me as one of the top dozen barriers to successful project delivery.

> *The solution to any problem always lies at the next level up.*

Like any systemic problem, the solution does not lie with the individuals confronted by the problem daily. Exhorting them to solve it on a case-by-case basis does not deal with the root cause. The solution lies in re-organising the work, to separate the two incompatible streams—and only the IT manager can do that.

Looking Inside: Consulting with Staff...

Implementing this solution did not take place overnight. The proposal was raised and discussed over several weeks amongst the team leaders. "Nobody will want to do only support work", said one. Not so: it turned out that staff were split in their preference, but there was a sizable number that preferred support work, and many enjoyed both. What about job security for project team members? What about career planning and gaining new experience? What about people getting bored? What about managing variation in project workload? These and many more objections were used as input to develop a robust new structure:

- Two organisational units were set up: one for Support tasks and one for projects
- All staff were attached permanently to the Support unit
- People interested in project work were transferred on a temporary basis as required to a temporary team set up for that specific project in the Project unit.
- Support tasks were streamed through the various teams in the Support unit
- Endorsed projects were actioned by the relevant temporary team in the Project unit

...and Outside: Communicating with Clients

Of course, planning the new structure was only a start. There was the significant matter of stakeholder engagement and education. The various business areas had developed relationships over years, and those relationships were about to change. To ensure that the work would be streamed appropriately, a formal definition of Production Support was required. The IT Steering Committee discussed and adopted the following definition:

> *" maintaining of a production system at the status quo, in terms of data, function, procedure and education."*

For the purpose of management of the production support process it was useful to categorise incidents on a causal basis. This approach enables more rapid analysis and resolution of structural problems identified through routine reports. The following categorisation provides for unambiguous, broad, cause-oriented analysis[1].

1. *Defect Repair*

   This category covers areas where errors in the coding, the data design, or the procedures interferes with the system functioning. "Bug-fixing" falls in this category, as does correction of errors of fact in manuals.

2. *Perfective Maintenance*

   Where a requirement has not been supplied in the system, then it is necessary to provide it, so as to enable the system to deliver the functionality assumed in the organisation's business plans. Common sources of this type of maintenance are variations in a core procedure that were not specified.

3. *Performance Tuning*

   When a system is modified to provide greater efficiency then no increase in functionality has been provided: the system still does the same things; it simply does them with fewer resources. This may be necessary in order to

[1] The basic definition and categorisation are from Rob Thomsett: however, the discussion and specific examples are mine. Thomsett's model differentiates between "environment and architectural support" and "upgrades and conversions".The two were grouped together in this implementation.

enable the system to continue to function as specified—for example, with response rates at the specified level.

4. *Adaptive Maintenance*

The business world changes--often rapidly, and if benefits are to be delivered by the computer system consistently then changes in the environment will necessitate changes in the system. Again, no increase in functionality is involved, but loss of functionality is avoided. An example of such a maintenance requirement is changes needed to a personnel system to implement changes in income tax legislation.

5. *Environmental and Architectural Support*

Computer applications run on machines—computing environments—which have maintenance and development requirements of their own. Changes to the computing environment (databases, the development languages, the operating systems and so on) can necessitate modifications to the applications to ensure that they continue to function as specified.

Upgrades and conversions of applications supplied by third parties may be required: for example, upgrading to enable continuation of support agreements, etc, when the vendor determines not to continue support of a previous level of a system. Other considerations may also recommend upgrade or conversion of a system.

6. *Consultation & Education*

A significant amount of time may be involved in providing advice and assistance to users of the system. Examples of such tasks include: advice on procedural matters to operators; advice on derivation and presentation of data to management; assistance in problem resolution where the problem is due to incorrect processing, etc. These activities do not involve coding or other obvious computing tasks, but they are necessary to keep the total system running

Sub-categorisation of these categories is feasible and could be found valuable on large sites with very significant production support overheads. It was found more cost-effective to sub-categorise only when targeting a specific category: the overheads of routine sub-categorisation are large.

A distinction was drawn between different types of activity undertaken after a system has been developed, implemented, and signed off. Enhancements were clearly differentiated from, and excluded from, production support through definition as: *changing a production system to increase the functionality of the system*. Accordingly, the only way to get these implemented was to treat them as a project

> *"In analysing the operational budget, the manager asks:*
>
> *What is the minimum that needs to be done in this area to prevent damage?*[2]"

[2] Drucker, Peter, "Management", Heinemann, London, 1974

What happened?

The staff responsible for production support were able to modify their workflow to implement the model within a few weeks. Reporting to system administrators and Steering Committees on their specific system in detail commenced the following quarter. The non-technical language in which the model was presented allowed it to be disseminated to technical and non-technical staff with equal rapidity. Within weeks the non-technical system administrators were able to discuss the scope of their authority to initiate service requests in meaningful terms with their system operators, their managers, and with technical staff.

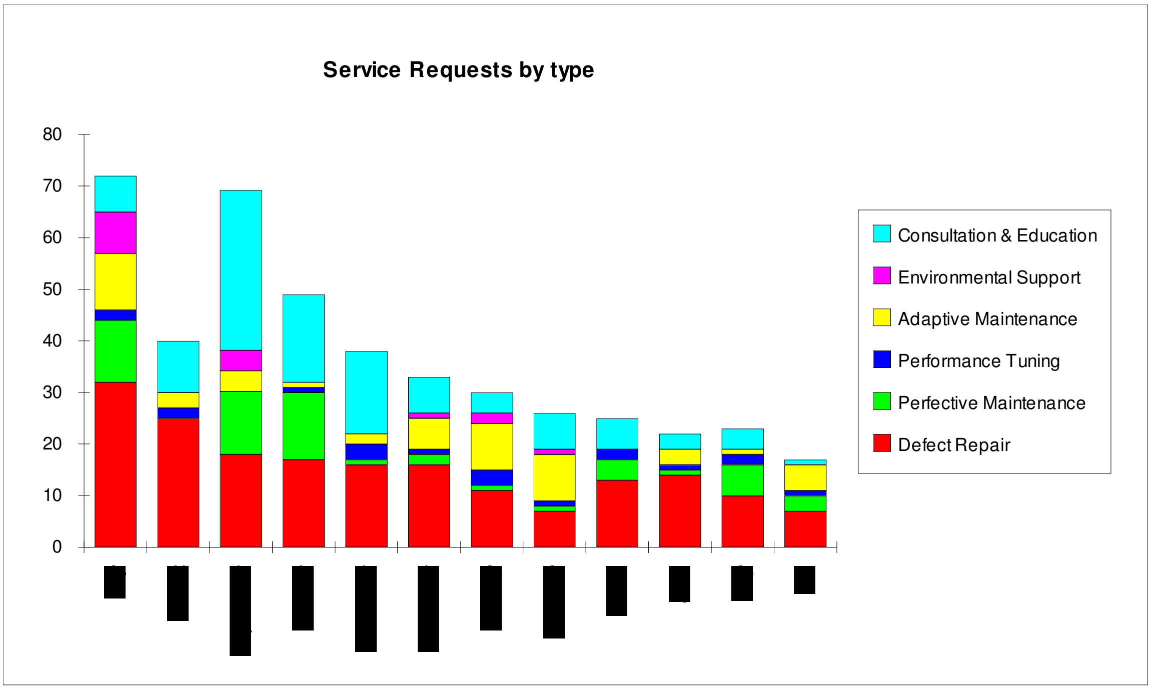

Figure 1: Number of service requests by type over the year following implementation

✓ A key outcome was that the owners of the systems had the information to control the process. They understood where the cost was being expended and why, and they could make informed choices on actions. Thus, for example, if the number of consultation/education requests rose, they could now identify the slowdown in production due to insufficiently trained operators, and take appropriate action. Finally, there was feedback to them on the impact of their decisions. The aggregated data showing service requests over time at Figure 1 demonstrates the rapid impact of managing production support activity.

✓ A further outcome was the impact on overall support cost, which trended downwards as shown at Figure 2.

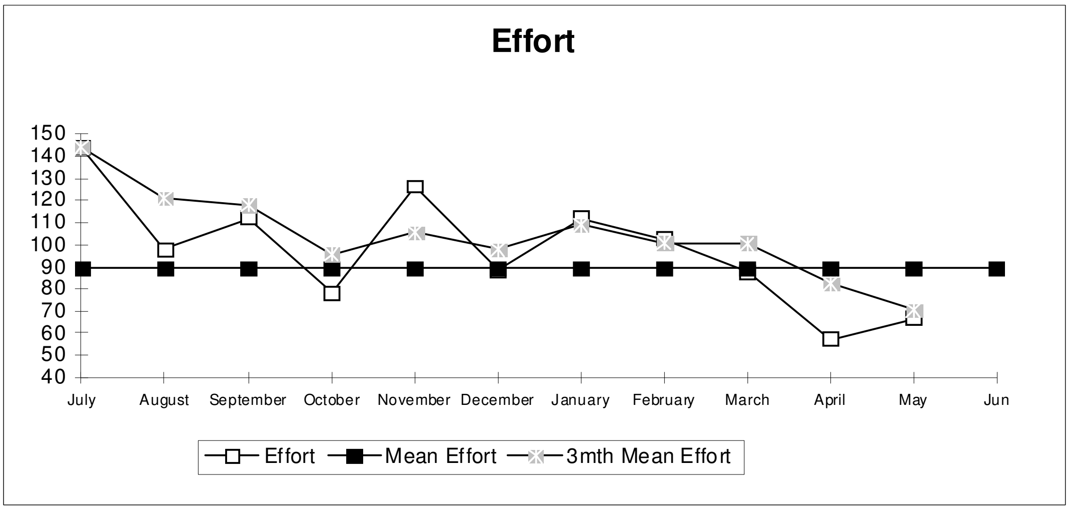

Figure 2: Effort expended on service requests over the year following implementation

✓ In line with these trends, overall expenditure of applications staff on production support dropped by approximately 18 % per annum over three years. This would be paralleled by a drop in non-technical staff costs.

- ✓ ...and to cap it all off, project timelines came back under control within a just few months.

Conclusion

Fast forward to 2007, and I was outlining this experience with a colleague. "Hey, I did that just a couple of years ago, and I got exactly the same results!", he said. Over a decade later, first principles still apply. Highly structured technical teams in large shops now routinely have a "Break Fix" team that is independent of the enhancement and development functions. If yours doesn't, consider what actions you could be taking to reduce costs and build agility.