



# Does Process Stifle Developer Creativity?

## A Layman & Layman White Paper



"It sounded an excellent plan, no doubt, and very neatly and simply arranged.  
The only difficulty was, she had not the smallest idea how to set about it."

-- Lewis Carroll, *Alice in Wonderland*

When an organization undertakes to define its systems development process, it is common to hear this objection: "But..., but..., process is a straight-jacket that will stifle our creativity!" How valid is this complaint?

To consider the validity of the objection, we first have to ask ourselves what 'creativity' means in the context of systems/software development. We often bandy about the word 'creativity' as if it had a simple and obvious meaning, when, in fact, there is no single definitive sense of the word. Certainly, our systems development efforts are creative in the sense that they produce something that is wholly or partly new, but only rarely do they represent a leap of imagination so novel as to represent a breakthrough never before conceived. Craft exists on one plane, innovation on another, and inspiration on yet another. Each is *potentially* creative, but *actually* creative only insofar as something useful is produced. At bottom, creativity is an exercise in problem solving, and in the corporate context, the problem to be solved may have facets other than the brilliance of the end product. So the meaningful measure of creativity – and the gauge of whether or not a particular approach stifles or promotes it – lies in objectively assessing how well we are solving problems.

If process is antithetical to creativity, why is it that we often hear the term "creative process?" For nearly a century now, it has been recognized that creativity, itself, involves a process. And, just as models have been put forth to help us think analytically about what is entailed in the process of systems development, so, too, have models been proposed to represent how creative thinking proceeds and innovation results. Creativity is not due to magic, but to a deliberate effort combining analysis and imagination. Creative effort requires structure and must be purposefully directed if we are to avoid the hazards of faulty cognition and coordination.

A well-known model of the creative process was published in 1926 by Graham Wallas and Richard Smith in *The Art of Thought*. Their five-step model consisted of *preparation, incubation, intimation, illumination, and verification*. These stages are as applicable to systems and software development as they are to any other creative pursuits, and have parallels in activities such as requirements gathering, analysis and design, prototyping, synthesis, change control, and testing. Other activities such as defect tracking provide information we can use analytically to optimize the process over time.

Systems development is a group effort - a complex activity requiring the contribution of many hands and minds. Other examples of group creative activity come readily to mind: advertising and filmmaking, for two examples. These industries rely on process to guide

the work they do. The process is prerequisite, for no complex undertaking can proceed without first establishing a process to structure the approach to be used. If film crews had to reinvent the filmmaking process on every project, few films would ever be released. Instead, filmmakers rely on a process that is the distillation of good practice drawn from broad experience. This process frees the film crew to focus its creativity on the end product, rather than on reinventing or rediscovering the process itself. In the software field, by contrast, many view process with suspicion, believing that it mechanizes the work and introduces pointless overhead.

The misconception that process is restrictive springs from a fundamental misunderstanding of process and process models. Process provides direction, structure, and visibility for our efforts, and serves to align them with corporate objectives. Whatever overhead is entailed in a defined process must be balanced against the cost of the wheel-spinning projects do when they blaze the process trail anew each time.

We can measure the "goodness" of process along various dimensions, but we commonly think of process improvement in terms of reducing error and rework. This leads to another misconception about process: that it cannot allow for inventiveness and experimentation. Work that produces something novel can be expected to involve trial and error, so process *should* be flexible and tolerant of rework in such cases. Process architects must consider the range of projects the organization undertakes and provide a tailorable process that distinguishes between truly groundbreaking efforts and work that is more mundane.

Of course, it is possible for management to "do process" badly, resulting in an oppressive regime that hampers creativity rather than facilitates it. But that is no more an indictment of process than bad drivers are an indictment of the automobile! Properly designing and deploying a good process with the right level of detail, prescription and tailoring options requires an experienced hand. Where an organization is struggling with process, they should hire the necessary expertise or seek the advice of a consultant. Surveys have shown that companies with effective processes have better staff morale than companies that are less process-oriented. Developers can be expected to express fears about the introduction of process, but ultimately they will find process liberating. Far from stifling their creativity, process serves to free developers from project thrashing and the myriad burdens an undisciplined approach adds to their workload.