A Software Process Immaturity Model

Anthony Finkelstein
Imperial College, 180 Queens Gate, London SW7

The software process maturity model (Humphrey, Kitson & Kasse 1989) is now well established, and together with the associated assessment approaches, has been the focus of considerable attention from software producers and acquisition authorities. The model breaks down the software engineering capabilities of organisations into 5 maturity levels from Level 1 - Initial to Level 5 - Optimising. The model is summarised in Table 1 below.

LEVEL	CHARACTERISTIC	KEY PROBLEM AREAS
optimising	improvement fed back into process	automation
managed	(quantitive) measured process	changing technology problem analysis problem prevention
defined	(qualitative) process defined & institutionalised	process measurement process analysis quantitive quality plans
repeatable	(intuitive) process dependent on individuals	training technical practices process focus
initial	(ad hoc/chaotic)	project management project planning configuration management software quality assurance

PRODUCTIVITY & QUALITY

RISK

Table 1: a software process maturity model

Assessment gives the impression that a significant proportion (according to SEI data more than 70%) of these organisations are at Level 1. The framework is, of course, misleading. Many of these organisations lie well below the merely chaotic. They belong to Levels 0 to -2 of the extended software process *immaturity* model we propose. The model is summarised in Table 2 below.

LEVEL	CHARACTERISTIC	KEY PROBLEM AREAS
foolish	(negligent) failure to allow successful development process	software reuse
stupid	(obstructive) counter-productive process imposed	development environments repositories
lunatic	(contemptuous) disregard for good software engineering institutionalised	automatic programming

SENSE

IDIOCY

Table 2: a software process immaturity model

The ad-hoc and chaotic processes followed by organisations at Level 1 can, by dint of exceptional individual and team effort, produce software. Level 0 foolish organisations act in such a way as to prevent this effort bearing any fruit. Where specifications and documentation are produced a Level 0 organisation will lose them. Where a successful software development team is working a Level 0 organisation will change the requirements to ensure it fails. Where a program is written they will so mismanage their configuration control as to ensure that the wrong version is released.

All immature organisations (in contrast to Level 1 organisations) fail to recognise that their management is severely awry. They believe firmly that a technical fix will solve all their problems. For these organisations management issues almost never appear at the top of "key priority issue lists".

Level 0 organisations perceive their primary technical problem to be software reuse. With software reuse in place they are guaranteed, they believe, never to make a mistake worse than they have made in the previous systems they have developed.

Level 0 organisations block effective software development by negligence. Level -1 stupid organisations act positively to subvert software development. These organisations insist on complex processes, involving the use of arcane languages and inappropriate documentation standards. They deploy significant effort and a substantial proportion of their resources in order to impose these. They insist on approaches for which tool support is unavailable, where tool support is available they impose procurement standards which prevent its purchase.

Level 0 organisations perceive their primary technical problem to be software development environments and repositories. With a suitable environment they can, they believe, enforce their policies and processes consistently. They can prescribe standards for and control all documentation. In such an environment they will be able to be able to plug in tools which conform to a complex public tool interface or better still develop their own.

Level -1 organisations while acting in such a way as to prevent software being developed sincerely believe that they are assisting. Level -2 lunatic organisations are contemptuous of advances in software development. They do not care if they produce poor software as they will probably make more money maintaining systems than developing them in the first place. Level -2 organisations have no individuals who know or understand anything about the software development process having dismissed them or promoted them to administrative positions away from software development. Level -2 organisations have a manual describing their software process written many years ago by a software engineer who has long since left the firm. They are proud that nobody ever reads it, should anybody want to read it they would not be able to find it. Level -2 organisations collect vast amounts of quantitative information on software development. They use it to produce voluminous project management reports which cannot be understood. The original data is destroyed. Level -2 organisations reward failure.

Only by a miracle can a Level -2 organisation produce any useable software. As Level -2 organisations rarely get beyond specification they pin their hopes on automatically generating a program from that specification.

No immature organisations sponsor or use the products of research though this does not prevent them from wishing to have a say in the way software engineering research is directed. Immature organisations will not use an approach unless it is tried and tested - they will never try and test an approach themselves.

We are currently developing an extended assessment procedure which will aid in recognition of immature organisations and which we hope to make available to all major software acquisition organisations for a small fee.

References

Humphrey W.S.; Kitson D.H. & Kasse T.C. (1989); The State of Software Engineering Practice: a preliminary report; Proc. IEEE 11th International Conference on Software Engineering; pp 277-288, IEEE CS Press.