

# Light and Provable

Position paper for the OOPSLA 2000 Workshop on "Deploying Lightweight Processes"

Klaus Marquardt

Dräger Medizintechnik GmbH, D-23542 Lütbeck, Germany

Email: marquardt@acm.org

Copyright © by Klaus Marquardt

In systems that are quality centric, like life supporting systems, lightweight processes are often employed to gain efficiency while keeping or improving the quality. However, the introduction of lightweight processes faces two major difficulties that I would like to discuss:

- External process owner;
- Dependencies to configuration management procedures.

## **External process owner**

The product, the development process and the actual adherence to it are subject to audits by regulation offices like the FDA. Thus the development process is typically owned by a quality department (QD) responsible for regulatory affairs. When a project employs a lightweight process, it has to deal with some resistance from the QD and those developers that are used to the QD way of processes.

A successful approach is to define a meta process together with the QD, that merely describes what must be done in a very general way. A focus here are the effective requirements that usually stand behind a heavyweight process. These are ensuring that the right product is build in the right way – and the right way must be proved. Although most process owners assume a V model of development, this is not mandated by regulation offices. As an implicit V model can be a serious hindrance for introduction of lightweight processes, it should be kept out of the meta process.

When the project defines and refines its individual process, this process must be approved by the process owner. So a changing, stabilizing process causes some overhead and forces the developers to continuously reflect and describe their process.

Traditionally, for each developed component in the system a developer must sign responsible, and a review must take place. This prohibits exclusive use of collective ownership or pair programming. A possible solution is that those developers that were involved with that component, sign for the review with one of them being the main author.

## **Dependencies to configuration management procedures**

CM procedures allow to reproduce each version, and also allow to maintain a quality assurance state with each component. In lightweight development processes, changes are the norm rather than the exception. The CM procedures must not prevent changes like renaming, removing and splitting components or changing their relations.

CM can bring its own penalty here, increasing the developers resilience to change components that are under CM control. Where the CM procedures can not be simplified, scripting can become the most valuable task of a coach.

When a quality status is maintained in the CM system, the notion of “what does ready mean” becomes important. In processes based on change, even “ready” components are changed again when further knowledge has been gained. So the CM procedures must allow an automated way to indicate the status of new versions, especially without a formal review for intermediate versions.