

“Practice Not Process” – Improving the Capability of Software Startups

Gerry Coleman

Department of Computing & Mathematics

Dundalk Institute of Technology

Dundalk

Ireland

Tel.: +353 42 935 2828

e-mail: gerry.coleman@dkit.ie

ABSTRACT

For small software companies, running small software projects, the question of the need for process arises. There is growing evidence that large corporate improvement models such as CMM/SPICE are not suitable in the small and that the overhead involved with process definition is exorbitant.

Even though process can offer advantages to small companies, in the case of the software start-up where no history of software development exists, using the correct practices can be the difference between survival and demise [1].

This places the focus on the requirement for a *pre-process* model which focuses on practices and provides a road map to bring the company to a level where defining process and preparing an ongoing maturity path is an option.

Keywords

Software, process, methods, practices

INTRODUCTION

Whilst all companies have essentially the same business goals, “Faster/Better/Cheaper”, in small companies, these factors, because of very limited resources, may be in direct competition. Also, one of those business goals may take precedence depending on the business environment in which the organisation operates. Finally, because of the cost, particularly in terms of allocating human resource, process definition and documentation take a back seat.

LIGHT METHODS MENU (LMM)

This ongoing study is developing and applying a pre-process model termed the “Light Methods Menu” (LMM) – see figure 1 – which contains the range of software development activities which are the most likely, on a project basis, to meet a software start-up’s business goals and produce business benefit.

Increase Function	Reduce Cost	Reduce Time To Market	Improve Quality
Requirements Management	Project Planning	Requirements Management	Test Management
Iterative Development	Project Monitoring	Iterative Development	Iterative Development
User-Driven Development	Design Patterns	Design Patterns	Test-first Programming
Pair Programming	Earned Value	Pair Programming	Pair Programming
COTS Components	COTS Components	COTS Components	COTS Components
	Continuous Integration	Continuous Integration	Continuous Integration
	Operational Profiling	Operational Profiling	Operational Profiling
.....

Figure 1 – Light Methods Menu (Sample)

The LMM draws heavily on the practices extolled by XP, DSDM, Scrum and other Agile process models [2, 3, 4]. The objective is to determine if these practices work in the target environment and, if so, to establish them and make them part of the company’s everyday software development.

APPLYING THE LMM

Validation of the model will result from trialling it in selected study companies and analysing and evaluating the results.

Other companies, where the LMM will not be used, will function as control companies to determine what business benefit gains in the study companies can be solely attributed to use of the LMM.

The LMM will be based on existing best software practices, which are scaled down for use in the small. The LMM, therefore, will be used at the project level, where in advance of development, companies will identify project-specific goals and choose which activities, contained within the LMM, provide the best option toward meeting those goals in the specific project and the specific company environment.

The LMM includes practices for all phases of software engineering such as project management, requirements management, development methods and models, configuration management, quality assurance, documentation, risk management, reuse, lifecycle tool integration, standards and templates, measurement etc.

The LMM is designed to be sufficiently flexible as not to hinder the company's objectives through excessive formality or bureaucratic overhead.

Ultimately, by adopting the software development practices that provide most business benefit, companies can determine which approaches work in their specific circumstances. By witnessing the real benefit associated with focused practices, companies can then customise their own process improvement effort, specific to their business environment, which can then be defined and documented.

As this is ongoing work the LMM is still being refined. At the workshop this paper will report on:

- The benefits the LMM has brought to software start-ups
- The issues faced by software start-ups in meeting customer expectations
- The applicability of lightweight methods in the start-up situation.

The early results of LMM application will be presented and the issues it raises will be addressed.

This is ongoing PhD research and will provide the workshop with some discussion points on practice Vs process, the real issues faced by software start-ups and if/how Agile methods can support the goals of "Faster/Better/Cheaper" in the start-up environment.

REFERENCES

- [1] Merz, U., "Tactical Project Management at a Startup", Software Test & Quality Engineering, Nov/Dec 2001.
- [2] Beck, K., *Extreme Programming Explained: Embrace Change*, Addison Wesley, 1999.

[3] Stapleton, J. *DSDM: Dynamic Systems Development Method*, Addison Wesley, 1997.

[4] Rising, L & Janoff, N.S., The Scrum Software Development Process for Small Teams, IEEE Software, Jul/Aug 2000.