The Weakest Link

Abstract: The software industry has established several advanced and sophisticated methods to help organizations with their process improvement programs. Those methods use models, metrics or agility indexes and are generally intense to implement. This article presents an alternative, a simpler method that can get organizations, in particular small settings, to improve through quick-wins.

The context

The software development process is a value chain. Through a succession of value-adding *transformations*, the requirements of a project ultimately end-up in the form of a production software. The identification and removal of the weakest of the transformations of the value chain within the organization constitutes the core of the proposed process improvement method.

The challenge

But let's start the discussion with a quick look at the current process improvement trends. The industry has developed very advanced and sophisticated process improvement methods. Some methods are based on capability and maturity measurement (i.e. CMMI®) and will help an organization benchmark themselves against best-practices and implement well-planned improvement strategies. Others, based more on metrics, scorecards or even near-scientific approaches will measure the organization performance in a purely quantitative way and rationalize an action plan. All those methods have proven to be very successful and efficient; they however all require a time and budget commitment that many organizations cannot sustain.

The solution

The proposed method described in this paper, uses a simple analysis of the software development value chain within the organization, to identify possible quick-wins. The software development process should be seen as an ecosystem, where various people (roles) use processes and tools to transform raw material (the project or product vision statement) into a software solution. For examples, a business analyst will transform a vision into requirements, an analyst will transform use cases into a solution design, a developer will transform a design into source code, a tester will transform non-tested code into tested code... Even related disciplines such as project management and configuration management are *transformation* activities, i.e. from an unplanned set of activities to a planned and resourced development cycle.

Now that the concept of *transformation* activity is defined, we must look closer into the software ecosystem. By doing so we can notice that at least three of its properties should be highly considered for the sake of process improvement:

- Firstly, it is important to acknowledge that that each *transformation* activity is adding value to the *previous transformation* activities in the value chain. The performance of the overall ecosystem is really the sum of all the adding-value *transformation*s, each transformation contributing its own piece to the overall result.

- Secondly, the software development value chain is such that the *transformations* can only deliver high value-add if the previous *transformation activities* in the value chain were also of quality. For example, an architect creating a terrific architecture is of little value if the requirements he was working from were incorrect.

- Thirdly, the value chain is such that any *transformation activity* in the ecosystem could jeopardize the overall effort, if done poorly. If the developers of the organization could not code even to save their lives, then the end-result would still be substandard.

Conclusion

So the conclusion is that to improve the efficiency of the overall ecosystem, one must eliminate the *transformation* activities that are not adding the right value: the weakest links.

The idea behind the proposed process improvement strategy is to improve, one by one, through natural selection, the transformation activities that are not bringing added-value.

By doing so, we enable a better overall result at the end of the value chain. The method consists therefore of focusing on one transformation activity at a time (the weakest link) and improving it until it is not the weakest link anymore. As soon as the weakest link is not the weakest link anymore, then it means a new weakest link should be identified and corrected, and so on, making the value chain stronger every time.

Transformation activities bear more weight on the final result when they are earlier in the value chain. So when two transformation activities are found equally weak, the one happening earlier in the value chain should be tackled first.

The method offers several advantaged over more "industrial" process improvement methods:

• It only requires a small investment at a time, only what is needed to identify and correct the weakest link.

• It enables the quick-win effect: Each win boosts the morale of the team as the impact has immediate value, for the smallest possible effort.

• It is less disruptive of the overall ecosystem.

• The return of investment is the highest at the beginning, delivering immediate boosts in quality, time-to-market and company bottom-line.

117 Lakeshore Road East #403. Mississauga, ON

P: 905-486-0587

F: 905-486-0588

info@gem-up.com

www.gem-up.com

Software process improvement in small settings is a service provided by GEM-UP CONSULTING.

GEM-UP CONSULTING is a management consulting practice focusing on providing expert advices to organizations in the business of developing software.