

## Position Paper for the SEMAT Initiative :

### *Software Engineering for Software Engineers*

*Ilsoo Ahn*

*Software Engineering Center*

*Korea*

---

I appreciate that the initiative is timely and right on target. It is a rare accomplishment in itself that so many renowned experts of the field have gathered to support a common cause.

First, it is necessary to define the meaning of the term Software Engineering in the context of the SEMAT initiative and also for the public usage. It is confusing what Software Engineering means depending on the context. Software Engineering is *about process-oriented activities (requirements, design, quality assurance, process improvement, and project management)* in many cases, though ACM's 'Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering (SE2004)' points out that such a notion is a *misconception*. Sometimes, Software Engineering seems to be about all the activities of software engineers as Electrical or Mechanical Engineering is about those of electrical or mechanical engineers. Depending on the definition, the scope of the SEMAT activities will vary greatly.

Working for many years as a software engineer and later as a technical manager, I've observed that Software Engineering has not been very effective in enhancing the productivity nor quality of software development. Engineers (developers) are weary of extra tasks implied by Software Engineering. Frictions arise between engineers and process people, and between engineers and managers. Though engineers make an effort to follow the rules, they seldom do have time to do it properly. They

end up going along with the process, but only in syntax and not in semantics. It turns out that various tasks imposed in the name of Software Engineering do not really help software engineers do their job better nor make their lives easier. Software Engineering is regarded by software engineers as a scheme that managers employ to control and manage them, not as a mechanism that can be useful to develop better software. No wonder that Software Engineering cannot be effective in such an environment.

I hope that Software Engineering can make the life of software engineers easier, not harder, by helping them do their job better and quicker. It is ironical that all the recent advances in the computing hardware and software technologies are making life easier and better for everyone, but not for software engineers. They still struggle to meet the unrealistic due date with unstable requirements much the same way as their predecessors used to. In fact, software development consumes so much time and efforts without much reward in many cases that it is losing appeals to young talented people as a profession. This is a serious problem with social implications that deserves a serious attention.

Through the SEMAT initiative, I hope we can make Software Engineering really useful for software engineers in developing software better and quicker. To make Software engineering for software engineers, we need to support the following characteristics:

- Based on trust rather than control or management
- Considerations for concerns of software engineers
- Practical merits rather than formal or theoretical orientations
- Light-weight process
- Essential elements identified for a minimal compliance
- Optional elements defined for further expansion
- Scalable from small to large projects
- Flexible to handle projects with fixed price and firm deadline
- Supported by an integrated set of tools
- Lessons from best practices
- Based on theoretical foundations

There have been more than enough theories, ideas and experiments proposed and practiced over the years. I doubt we have enough time to invent new ones, adding to the already crowded field, validate them, and gain a wide acceptance in the time frame of about a year. Instead, we evaluate existing ideas and practices, and choose a list of promising ones as candidates for further investigations. I believe recent movements towards agile methods are promising, but a large amount of additional work is necessary to make them practical in a variety of situations. We need to:

- Collect metrics from many case studies
- Evaluate the pros and cons of various methods
- Refine them for various types and sizes of projects
- Reconcile them with traditional frameworks such as CMMI and SPICE
- Provide a set of integrated tools
- Provide theoretical bases
- Offer practical guidelines for practitioners
- Make final recommendations for each area of Software Engineering

I believe Software Engineering is in need of and ready for a major paradigm shift. The SEMAT initiative can seize this opportunity to make a great contribution by making Software Engineering for software engineers.