**SOFTWARE PROCESS**

-Alfonso Fuggetta , Elisabetta Di Nitto

There is no aspect of our daily life that is not being affected by software. Hence it is very important to understand, evaluate, and improve the procedures, techniques used during software development. The main aim or motive of this paper software process research is to discuss the issues and contribute to tackle the problems that arise during a software process. Given the advent of concepts like open source development, an agile method etc., there has been a lot of transformation in the software development.

It looked into:

* Reconsidering the past approaches and into improving the way people dealt with these issues.
* Main trends in the software process research over the past decade.
* Key challenges encountered.
* Proposing some directions and suggestions for the future research.

**SOFTWARE PROCESS IN YEAR 2000:**

The questions the researchers asked themselves are:

* Were the results really able to influence and address the issues and problems experienced by software developers?
* Were they too often an intellectual exercise unable to concretely affect the way software is conceived, developed, deployed, used, and managed in real and practical scenarios?

The major areas of research were:

* Process modeling and support: involved generation of Process Modeling Languages and environments to describe and model these processes. These models were precise enough to analyze the process.
* Process Improvement: as the software development activities were continuously evolving, it aimed at analyzing the maturity of the process and finding ways to improve it.
* Metrics and empirical studies: as the software development activity is continuously evolving, researchers used empirical studies to analyze and determine the performance of the software.
* “Real” Process: Finally, based on the experiences and achievements of the previous decades, several “concrete” processes were conceived and applied

The FOSE 2000 paper analyzed these attempts and elaborated a number of considerations and criticisms. For example, “The purpose of process languages and technology must be rethought”. As many languages developed during the 90s were too complex and rigid. The dramatic and incredibly fast development of many enabling technologies and practices has further reinforced the idea that software process research definitely needed a significant change in scope and approach.

**SOFTWARE PROCESS IN THE PAST DECADE:**

Software process performance is heavily influenced by and structured around the role and behavior of individuals and organizations. To characterize the shift in focus, it provides an overview of the most relevant directions in software process.

* Social Aspects: Improvement is the result of proper interplay between human-centered aspects (social aspects) and technology.
* Agile process: A drift from the Traditional models(waterfall models). agile methods lack structure and organization, and therefore are difficult to apply to mature and large software development processes.
* Global software engineering: the experiences of different authors at different organizations suggest that success and failure are driven primarily by the ability to establish effective mutual relations and trust among software developers.
* Role of empirical Software engineering: rigorous empirical studies’ goal is to show the advantages of some new approach/practice, or to demonstrate the validity of a research hypothesis.
* Model-Driven engineering: The main reference for model-driven development is the OMG specification named Model-Driven Architecture (MDA) which identifies three different viewpoints used to describe a system: the Computation Independent Model (CIM), the Platform Independent Model (PIM), and the Platform Specific Model (PSM).
* ALM: not explicitly connected to Software Process research, they offer mechanisms for automating some tasks, and for connecting managerial tasks with software development activities. Aims at taking care also of application operation procedures and managerial governance.
* Importance of automation: the most important application of automation in software processes is in supporting the final phases of software development. Automation is possible and exploited in all the activities and processes related to building, deploying, and operating software systems.

Configuration management:

git, software forges.

Quality Assurance:

Staf framework.

Software building :

Private clouds, introduction of virtualization and remote control mechanisms.

**MAJOR TRENDS AND CHALLENGES:**

* The Internet is the Development Environment: the Internet is the development environment, i.e., any development activity is carried out over the Internet. Software is rarely developed in isolation, as it is more and more the result of interaction, integration, and cooperation among developers and between developers and end-users.
* The Internet is the Architectural and Execution Infrastructure: any software is directly or indirectly operating or integrated over the Internet. Internet of things, smart services etc. emphasize this trend. Cloud computing is another example.
* Users are Mobile, Nomadic, and "Always On": designing software for mobile devices is not just a variation of classical development processes it requires new and specific techniques, policies, and methods able to effectively address the new challenges introduced by this change.
* The Internet is the Basic Distribution and Business Infrastructure: the challenges are, devices are permanently connected to the Internet, and software can be easily distributed, installed, and configured over the Internet which generated important innovation-app store.

**RESEARCH ISSUES AND DIRECTIONS:**

The problems facing are extremely complex and rapidly evolving. It is essential to continuously reassess and redirect the focus of the research work.

* The smart convergence is much more important than rigid compliance (development in agile processes).
* The classical distinction among design, implementation, and operation tends to disappear or to be radically redefined.
* It is essential to identify and develop new methods to assess, represent, and communicate the state of a software development process.
* Software is increasingly pervasive and ubiquitous. The issues related to security, privacy and trust become more and more critical.

Software is used in extremely different settings and is characterized by a variety of requirements and needs. There is no single approach or even characterization of problems and issues that can be uniquely used to drive and inspire the research work of the community. It is counterproductive and pointless not to reuse, recombine, and exploit the knowledge and expertise developed in other communities both in Computer Science and in other scientific domains.

* Y PREETHI PRAVALIKA

(IS201401052)

* S SAI HIMA

(IS201401038)