8-15-1997

Supply Chain Management and the Software Production

Jean_pierre Kuilober
University of Massachusetts, Kuilboer@umbsky.cc.umb.edu

Supply Chain Management and the Software Production

Jean_pierre Kuilboer
Kuilboer@umbsky.cc.umb.edu
100 Morrissey blvd, Boston, MA 02125
University of Massachusetts, Management Science and Information Systems

Abstract:

One can convincingly argue that non-technical factors account for the majority of the improvement in the software production process. In this complex technological world, defining the scope of a system considered for improvement is important but one should not lose sight of the interconnection with the environment. The general system theorists had introduced the system and subsystem concepts, applicable to the software development process. The environment is difficult to mold to our advantage except for the largest players in the field. The internal development process should be adapted to a supply chain management framework, which can be used to good avail for guiding a Software Process Improvement (SPI) program.

As part of an organization plan to improve productivity, competitive advantage, quality, or just survival, many avenues to Software Process Improvement (SPI) have been suggested. Many organizations, faced with the rising complexity of the demand for system development, and high competitive quality, cost, and time pressures have embarked in SPI programs. The product of this effort has been reported as a mixed bag of successes, failures and a large number of "status quo" where the resulting processes or products are different but neither proved better nor worse.

First of all, a software development entity should focus on its core competencies and leave the peripheral issues to some other party. For example, leading edge software developers such as Microsoft will delegate their own internal logistics to another software provider (i.e., Using SAP for HR, distribution, production). Also what many have painfully found out is that such enterprise resource planning (ERP) tools are better utilized for their intended purpose without extensive modifications in the initial period adapting the organization process to the ERP instead. One of the paradoxs is that the software industry, on the leading edge of developing productivity tools, workflow software, groupware, and project management software facilitating other industries successes in managing their processes have not been well-served themselves.

To remedy such as situation and gain competitive advantage projects are under way to provide a better management of the end to end software production process. For example the leading software supplier for Enterprise Resource Planning (ERP), SAP has a project underway which should change the way large software developers operate. SAP itself provides modules to other industry sector including service industry encompassing functions and process such as financial, production, logistics, human resources, and more. The SAP project, code named "Golden Gate," is under development for the software industry, with current partners such as Informix and Autodesk. The resulting deliverable will be a modular ERP targeted at the Software Production Process.

Faced with a difficult problem of improving on software process quality, under increasing time pressure, firms eternally are looking for a panacea. This phenomenon has been widely reported as the search for a "Silver Bullet" (Brooks, 1987). As expected no such solution exists but many have recognized the benefits of some tools and techniques offering partial solutions. They have been coined with the term "Silver Pellets" (Yourdon, 1992), and the secret of success is the careful integration of a full range of such devices. For an organization, offered so much seemingly free advice (other people success stories incompletely reported in the trade press magazines) and expensive advice (i.e. consultants will always offer a panacea to your problem for a fee) it is more important than ever to choose wisely.

A partial list of the SPI initiatives is potentially informative, offering a platter of methods, techniques, standards, guidelines from the informal common sense to the formal (e.g., clean room approach, formal notations...). Not all techniques are intended to be used concurrently as it could be prohibitively expensive. Some of the suggested alternatives are as follows:

The Capacity Maturity Model (CMM) developed with the process in mind has clear stages and is intended to show a path from Initial, Repeatable, Defined, Managed, onto an Optimized process. While it is theoretically attractive and has been time tested, few organizations have ever reached the fifth optimized stage and trying to skip stages has often been detrimental to the attempting organization. This approach is likely to be widely adopted and chronicled as it is an avenue supported by DOD and the public sector. Applied to SPI, Herb Krasner (1996), Chairman of the SQI Advisory Group has given a 9-point prescription that can help an organization progress through the CMM levels.

The Software Process Improvement and Capability Determination (SPICE) is carried under the auspices of the International Committee on Software Engineering Standards ISO/IEC JTC 1/SC 7, through its Working Group on Software Process Assessment (WG 10). This project is aimed at a first degree at assessing the software process and at the later stage provides some venues for process improvement.

The ISO 9000 series is in its second iteration. They are the most widely used guidelines for process control in a number of indistries and the documentation is supposed to be industry independent. In practice they have been adopted heavily in some sectors where the ISO 9000 certification provides a passport to global trade (i.e. electronics, automobiles) but have not yet received large support from small firms critical of the initial certification costs. As a means for success in the software sector the jury is still out. The non specific nature of the standard (even if one is aimed at software, i.e. ISO 9000-3) has hampered its wide adoption in a complex and fast-moving sector. The main endeavor of the guideline is to provide an assertion that the certified organization has a stable process which is clearly documented and that the established procedures are followed according to the written document. Consulting groups have advocated the adoption of ISO 9000, which has been also broadly adopted by more than 100,000 public and private organizations worldwide. ISO 9000 support has been embedded in some leading edge software, thereby facilitating the adoption process (i.e., ARIS- Toolset for Business Re-engineering from IDS Dr. Scheer).

The path to software quality is based on a sound software process and a feasible set of metrics. More than just in the phase of writing code the full life cycle is to be considered. As in the manufacturing sector some supply chain management should be adopted from the supplier to the firm itself through the distribution channels onto the final customers. Feedback from the customer will be essential to reach a quality process equivalent to the level 5 of the CMM. Concentrating on internal process will not be sufficeint to achieve excellence. More than just a development project, the release of software implies the real coordination of many coincident responsibilities (development, documentation, production, logistics, distribution, shareholder relation and lastly a clear human resource coordination). The four components of supply chain management will be covered in order to reach an improved standing and impress the customer. The four steps will be Plan, Source, Make, Deliver.

Plan

To have any chance of success, the organization should establish a plan, a roadmap which will provide some guidance. In the first place there is a need to assess the current situation. In this phase analysis of Strength Weakness Opportunities and Threat will be covered. What are the strengths (in terms of finance, products, employee, management, location..) and what are the perceived weaknesses. One should avoid assessing forever as the environment will change through time and the assessment could become obsolete. Given the result, complementary benchmarking will assess the competitive situation in the environment, benchmarking is suggested. Then a gap analysis can be used for further planning. ISO 9000 and a project to attain certification can be launched when it appears that the organization has a chance to reach a stable process in the medium term (one to two years). The planning process and ISO will provide synergy and incentive is to pursue the improvement effort. If the firm is start-up, in very deep trouble, has complete lack of resources, is rapidly downsizing, or some dramatic change is occurring in the environment then a re-engineering approach could be suggested. The ISO 9000 venture vaguely corresponds to a move from Level 1 to Level 2 on the CMM scale.

Source

In an increasingly complex software world environment, the second step of the supply chain is becoming increasingly important. The organization is now hopefully in control and has a plan of action. The benchmarking has provided the SWOT analysis and decisions have been made on the core competency (current and future). At this stage the firm will have to decide on sourcing and analyze its means of production. In the software industry this could mean strategic alliance, licensing, merger and acquisition, or cooperation/competition. Strategic licensing could facilitate extension of the software in various ways (e.g. The firm could outsource the port of the software to new computing platforms, international versions, and facilitate the integration effort with competitive products through the publication of its product API).

Make

A software development life cycle should be adopted. Through a methodology the requirement can be defined in a logical way, specification can be developed following a set of notation. The documentation of the process, which insures some independence from individual developers (a necessary step given the high turnover in this industry), is increasingly a prerequisite of any successful long term project. Tools can, therafter, be matched to the adopted method and productivity enhanced by synergy between the tools. Developers, provided with strong development tools and suitable workspace environment (e.g. spacious, well lighted, and quiet) will be more productive. The adopted method (from Structure Analysis and Design for traditional system, to the newest Object Oriented techniques such as OMT, Booch, Use Cases, to the Unified Modeling Language) should be matched with the adoption of automated CASE tools supporting the notation and semantics. Similarly the programming and testing process should be integrated through the wise choice of programming language, development platform, testing method and software, and the integration with third party libraries and components. For example an organization developing an accounting software should know which platform they want to support (e.g., AS/400, Windows 95/NT, UNIX) and choose their tool, libraries, and components accordingly. This will entail finding a reporting software or windows/widget library running on the same platform or at least being aware of the tradeoff before the fact, including the side effect of licensing on cost analysis.

Deliver

One of the functions in the software development life cycle is the delivery of the system. The environment is rapidly changing. New technology, mode of distribution, user perception could affect the other steps of the process. As is often the case, delivery is not the forte of most software firms; it is often better to use a combination of distribution mechanisms provided by third party. Three cases could be covered. The first is the internal delivery of software and the second is the delivery of software made for external customers. The third could be the software developed to enable better flow of the supply chain. All will involve some logistics, some coordination of distribution channels, and some administrative recording.

Decisions have to be made as to which distribution channels will be utilized. Current and future alternatives include the reseller software shops, the mail order companies, direct through the Internet, direct sales force through the organization representatives, and through system integrators and consultants.

A potentially explosive growth in distribution channel will be the Internet, providing an instant pipe to the customer computers whence the bandwidth difficulty is solved. The opportunities exist not only for original distribution of software but for improvement in the developing process. With time pressure being a given, software developer can distribute Beta of new version of the software. A creative enticement could generate feedback from the early users, thereby facilitating inclusion of feature and fix of bugs. Once the production release is delivered, electronic feedback could be a medium for speeding the review process.

Feedback

Changes in distribution methods have brought increasing time pressure on the software developer. When the competition has a release cycle of six month it is difficult to advocate a cycle of two years for your own product. This aspect should be critically analyzed and while the firm is under pressure to release a product,

releasing a stable product is still the preferable goal. On the desktop, market leader Microsoft, has recently officially reverted to a somewhat more traditional product cycle, citing institutional customers market resistance to rapid upgrade cycle (i.e. 6 months). The competing philosophy is the currently debated "push" technology where modification to the software would be "forced" to the computer with any modification to available release. Whatever is the adopted philosophy any software developer should better be informed of the implications of any decision. The internet generation offer opportunities for SPI which should be explored. Once the software delivered, it used to be severed from the provider who heard about feedback only through support line complaint. Presently the Internet offer a two-way communication link allowing easy interaction, access to knowledgebase, bug reports, and patches. A deficiency does not have to wait delivery of the next version to see a remedy.

Conclusion

Many avenues for SPI exist. In today environment the problem is not finding a suggested SPI method but to trim down alternative from among the multitude of partial solution. One has to keep an open mind. While most approach lack the theoretical foundation previously sought in a SPI method, pragmatism and common sense should be our guiding principle and better software production methods should prevail in near future. Not addressed in the preceding paper but an equally important concern in the industry is the shortage of skilled IT workers. SPI lead to productivity increase but one out of 10 IT position remains vacant and nationawide about 190,000 positions are unfilled (ACM, 1997). Such a situation should linger in forthcoming years and will without doubt bring more pressure on any SPI offering.

References

ACM, "Workforce study," Communications of the ACM, Vol. 40, No 5, May 1997, p. 9.

Brooks, F. "No Silver Bullets, " IEEE Computer, April 1987.

Krasner, H. "The Prescription: Imagining Life in a Mature Software Organization," Software Quality Matters, Vol. 4, No. 4, Winter 1996.

Yourdon, E. Decline & Fall of the American Programmer, Yourdon Press Computing, Englewood cliffs, New Jersey, 1992.