Diversification in Software Engineering: Concept, Applications, Challenges

Manas Kumar Yogi 1 N. Padmavathi 2
1 Assistant Professor 2 B.Tech
Pragati Engineering College, Surampalem Kakinada, A.P., India

Abstract—This paper examines factors that have advocated the diversification of software process models. The purpose is to understand more clearly the problem-solving process in software engineering and to identify criteria that can be used to evaluate alternative software-driven problem-solving strategies for differing project requirements. A review of software process modeling is given first, followed by a discussion of process evaluation techniques. A taxonomy for categorizing process models, based on establishing decision criteria, is identified that can guide selecting the appropriate model from a set of alternatives on the basis of model characteristics and software project needs. These criteria can facilitate adaptability in the software process so that the process can be “altered or adapted to suit a set of special needs or purposes”. The factors that have contributed to the diversification of software process models have often been related to the expansion in goals and capabilities in the software industry.

Key words: TQM, Ignorance, Process, Quality, Diversification

I. INTRODUCTION

The growth in software capability—which reflects the impact of emerging technologies, evolving methodologies, and increasing interdisciplinary impacts—drives solution-related factors. The impact of historical development and problem- and solution-related factors not only helps explain the changes that have occurred in process modeling but can also help project or forecast future developments in process modeling.

The causative forces involve factors ranging from increased software development experience and more interdisciplinary background among software engineers to the degree of problem complexity; organizational goals; availability of technology; and changing cognitive styles in problem solving. Regarding the latter, for example, differing cognitive styles and paradigms of cognitive experience exist in every discipline and practice. For example, in the domain of management information systems, an in-depth understanding of its models, cognitive activities, skills, and knowledge can lead to improved approaches for developing information systems and allow problems to be solved more creatively and efficiently. Bottom-up, reverse engineering of models and cognate approaches may uncover original system design purposes. They can then be helpful when exploring the relationship among models in a specific discipline and can lead to establishing clearer frameworks for understanding.

II. UNDERSTANDING DIVERSIFICATION IN SOFTWARE ENGINEERING

At the problem level, the roots of diversification include:
- Scope and complexity of problems
- Types of requirements and forms of problems
- Need to learn and apply new capabilities
- Challenges of continuous change
- Impact of the consumer economy and interdisciplinary effects
- Development of e-business applications
- Multiplicity of stakeholders, project team skills, background requirements, and business goals

At the solution level, diversity has been driven by variations in:
- Project management approaches
- General standards
- Quality-assurance standards
- Hardware and software tools
- Networking tools
- Data mining and automation tools
- Nature, scope, and domain of applications
- Need for business-driven software engineering
- Secure software engineering
- “Killer” applications
- Mobile or wireless software engineering

Diversity is a prevalent characteristic of the software process modeling literature. This reflects the evolution in software development in response to changes in business requirements, technological capabilities, methodologies, and developer experience. Process diversity also reflects the changing dimensions of project requirements, with process models maturing over time in their ability to address evolving project requirements. Diversification is also driven by the increasing importance of interdisciplinary views in modeling software processes. The temporal parameter is correlated with greater demands and changes that require ongoing adaptation and increased complexity. Time also introduces greater capabilities that afford better problem analysis and resources. There are numerous examples of measures that change through the development of increased capabilities, including methodology, technology, experience, and interdisciplinary impact. Technological capabilities seem to have the most influence on process modeling in terms of their impact on process automation, visualization, and degree of process control. Thus, although early process models were manual and sequential in structure, this changed with the introduction of fourth generation techniques and languages. Process technology enabled the support of the rapid application development needed for iterative approaches with their greater emphasis on risk minimization and user satisfaction. Time also increases the accumulated pool of experience in process modeling development. The movement from the traditional waterfall to the V-shaped model, or from the conventional spiral to the win–win spiral model over the decades are examples of the effect of accumulated experience on process modeling structure.
and definition. This capability measure is also a function of problem-related factors, with increases in problem complexity and business requirements affecting the collective pool of experience and altering how problems were solved. The type of methodology adopted also has Considered an impact on process modeling evolution. Considered, an object-oriented methodology supports the architecture-centric approach in rational unified process models in terms of structure, automation, and visualization, as distinguished from process-oriented methodologies. Although these two methodologies exhibit generic conceptual similarities in the earlier phases of the process model, they become more differentiated as implementation related factors are Considered. The SOFL model presents an integrated approach that adopts structured methodologies in the requirements phases and object-oriented methodologies in the design and implementation phases. The adopted methodology can be driven by quality assurance and associated with the evaluation of software systems. Gradual improvement approaches such as TQM view problems differently than highly dynamic approaches such as BPR (business resource planning). For gradual improvement, SEI-CMM; the Kaizen approach; QIP; and the BUTD approach have been introduced with significant effects on structuring and automating the development process. The software field originated with little attention paid to human factors. The importance of social context disciplines was only later appreciated, driven particularly by the increasingly widespread awareness of the high failure rate of software projects and its relation, at least in part, to social science-related factors. At that point, human factors began to be accommodated more seriously—for example, through the use of systems dynamics modeling and greater attention to cognitive effects and behavioral models. This reflected a more interdisciplinary understanding of software problem solving. Economic considerations were more systematically addressed by incorporating risk management in the prototyping, spiral, and other iterative process models; they were manifested in the development process with increased attention to feasibility assessment, cost estimation, risk assessment, productivity, and control. Industrial engineering and operations research are examples of other interdisciplinary influences affecting the evolution of process modeling. The application of quality-assurance standards to business processes is one example. Software modeling, in terms of development structure and process visualization, has also been affected by the increasing impact of customers on business. Thus, iterative structures substantially escalate user involvement and customer–developer communication becomes more effective with greater visualization. Thus, customer Considerations have significantly affected process evolution. However, it is worth noting that working with small systems entails a different experience than working with large systems because modularization is not reliable without tailored approaches. Indeed, most of the influential drivers in process modeling evolution are time dependent, although time is inadequate to explain all the variation. Time can be thought of as a necessary requirement for problem- and solution-related drivers, acting as a trigger and a constraint. Although problem-related factors have been essential to precipitating changes, the availability of resources and capabilities (solution-related drivers) have had even greater impact on this evolution. This can be attributed to the impact of capabilities on problem-related factors. Thus, problem- and solution-related factors are not mutually exclusive, but depend on one other. The degree of automation, control, and integration and the extent to which changes in process structure take place can be used as measures of the evolution of software process modeling. Another Consideration has been the increasing degree of visualization provided for process models. Initial models, like the Waterfall, Evolutionary, and Spiral models, had a static view of the software development process, but later behavioral models explicitly portrayed the dynamic character of real-world software development processes. Indeed, with process improvement models and state-of-the-art advances in CASE tool technology, one is now able to monitor the development process in a multidimensional view, including full simulation of the dynamic behavior of the process. This advances the goal of efficiently controlling the software process.

III. THE HIDDEN VALUE OF DIFFERENCES

Paradoxically, diversity can be acquired through inheritance as well as by overriding the presuppositions that derive from inheritance. Cultural differences are examples of inherited characteristics that affect the degree of diversification in an environment. Scientific, social, political, psychological, philosophical, experiential, and other differences modulate acquired diversity through exposure to values, education, involvement, and interaction. Amid such diversity, commonly shared human needs play a unifying role. Conventional problem solving addresses problems by trying to eliminate the sources of contradiction; integrative problem-solving approaches try to capitalize on differences to obtain optimized solutions. Conventional problem solving eliminates or minimizes the other (the difference) in favor of specialization; cutting-edge problem solving incorporates (integrates) the other by inclusion. Obviously, not every kind of difference can or should become a factor in a problem-solving strategy. Some differences may reflect contradictory facts or disputes about fundamentals, some of which may be irrelevant to the issue at hand. However, the idea of integrating differences rather than removing them is worthwhile if the legitimacy and relevance of the differences have been established. The key to distinguishing between negative differences (which ought to be excluded) and positive differences (which ought to be integrated) is to determine whether the differences are valuable and relevant. If they are, they should be utilized, not ignored or eliminated. Many modalities affect the status or interpretation of differences, for example: The simultaneity factor. Some differences can appear contradictory when they occur simultaneously, but are actually complementary when placed in sequential order on a timeline. For example, Considered a false dichotomy such as whether analysis or design, process or architecture is more important in software development. Of course, when analysis and design are viewed as phases in a unified life cycle, each one is as important as the other. A business firm needs to diagnose a
problem before providing architecture for its solution, and an architecture needs to be tailored to a particular case. On the other hand, a good analysis is worthless if it is followed by a poor design. The unique answer factor. Differences can appear contradictory if only one element of a situation is taken as representative of the entire situation. This leaves no room for other contributing factors and no way to find relationships between diverse differences. For example, is a problem a technical or a business problem? Recognizing that a situation may arise from business as well as technical errors is totally different from understanding the issue from only a single perspective. Different elements can contribute to a complete picture and they may interact with or complement each other. Thus, a technical problem may affect business factors and business factors may create technical problems. The failure of a commercial Website to generate revenue may have been caused by inadequate technical support, which led to frustrated customers. A lack of appropriate budgeting may in turn have been responsible for the shortfall in technical support.

IV. INTEGRATION—NOT DIFFERENTIATION
What is really needed in solving a problem is to find out whether the relevant differences or diversities can or should be made to work together. The purpose in integrating differences is not only to ensure resolution of contradictory or conflicting factors. Indeed, diverse elements may not even be able to function independently of one another, and eliminating one element in favor of another may introduce other problems. To illustrate the integration of differences, Considered another false dichotomy posed by the following question: “Which is more important: the process or the project?” This is a misguided alternative because it implies differentiation is the only choice and that integration is out of the question. In fact, no process exists without a project and no project can have a successful outcome without the guidance provided by a systematic problem-solving process. Thus, the project and the process must be integrated, combined, or synthesized—not differentiated in an exclusionary sense by sacrificing one element for the other. In problem solving, it is tactically unwise to give priority to differentiation over integration because this tends to predispose developers to ignore or postpone examining the relationships among differences until they are compelled to do so by a roadblock in the solution effort. If differentiation is done first, a roadblock may occur after initial progress has been made in solving a problem when a difficulty related to some defect in the tentative solution is recognized. In this case, the process will be forced to backtrack, retracing its steps to determine what went wrong. By contrast, if one examines the potential benefit of integrating differences before selecting one of the apparent “alternatives,” the risk can be reduced. Thus, a differentiation-first approach is more likely to entail a costly restructuring of an entire effort in order to debug and correct a faulty process, but an integration-first approach may require only a preliminary inquiry and relatively primitive tests to evaluate the potential benefits of integration. Diversity is an organizational asset. It embodies the hidden value of differences: a value that is frequently underestimated, underutilized, or obscured in traditional approaches. Appreciating diversity is the only way in which one can successfully implement interdisciplinary thinking in software engineering. The purpose of investing in diversity is ultimately to exploit and incorporate the interdisciplinary knowledge that it represents into a unified problem-solving framework. Diversity investment leads to a wider understanding of the role of diversity in software engineering and bringing it to bear on issues identified during the problem-solving process. It also implies identifying new, unrecognized, or underutilized areas of knowledge and exploring new aspects of problem definition. One venue for doing this is by incorporating diverse requirements and capabilities into problem solving so that it is tailored to various kinds of business problems and project goals. For example, investment in diversity can be implemented by establishing training programs that prepare employees to think in an interdisciplinary way; to understand diversity; and to learn to incorporate diverse sources and types of knowledge to construct a broad-based approach to problem solving.

V. CHALLENGES
A. Factors That Affect Interdisciplinary Ignorance:
For present purposes, the term ignorance refers to a lack of data or the presence of inaccurate data in a circumstance in which such a lack hinders the proper understanding and definition of business and human problems. Ignorance in this sense includes lack of knowledge about available information as well as about adequate or effective tools. This results in a problem-solving process that may have unreliable or insufficient inputs. Understanding the sources and varieties of ignorance can help reduce the failure rate in problem-solving processes. Just as in the case of domain knowledge, domain or process ignorance is also an interdisciplinary phenomenon; thus, overcoming this kind of ignorance requires an interdisciplinary response. Although a thorough grasp of a problem area and the solution domain results in success, ignorance masks or obscures the real situation and thus broadens the distance between actual problems and their appropriate solutions. The many sources of ignorance include unreliable sources of information; partial knowledge; lack of communication and interorganizational ignorance.

1) Unreliable Sources of Information
This category includes inadequately accountable sources of information. Examples range from unconfirmed, inconsistent, suspicious, or doubtful resources to resources that are untrustworthy or lack qualification. Clearly, determining whether a resource is reliable requires examining the quality and credibility of the data and the data carrier; even computerized systems can be based on incorrect formulas, programming bugs, and inaccurate entries. Interdisciplinary capabilities are needed to eliminate or disqualify unreliable resources and to rate or rank sources, which can be human, digital, or hardcopy sources. For example, one can estimate the reliability of a human source by examining characteristics of subjects such as their skills, psychology, physiological criteria, etc. Technical testing may be required if data is delivered by electronic media. If a source involves specialized information, domain knowledge and expertise in the area may be needed to evaluate its reliability.
2) Partial Knowledge:
This refers to aspects of an issue that have not been revealed (so-called in-breadth ignorance) or information about a specific aspect of an issue that is left incomplete (so-called in-depth ignorance). This type of ignorance may even derive from a complacent or self-satisfied attitude—“what we do not know does not exist.” In-breadth ignorance assumes that information can be gathered using only one or two paths of knowledge, with other aspects of the problem not even considered for relevancy. Failure to recognize all the dimensions of an issue can result in solving the wrong problem and thus leaving the real problem unsolved. For example, although the infamous Y2K problem was at one level a technical problem, it had in fact many managerial aspects. For example, solving the technical dimension of Y2K was arguably easier than finding sufficient staff capable of reviewing systems for relevant bugs. In this situation, because of the intense demand for qualified staff, managing the available human resources became a real challenge. The “technical” problem was indeed interdisciplinary, like most business problems. In-depth ignorance may recognize the relevant aspects of an issue but not study them thoroughly enough to understand them effectively. For example, when the e-business readiness of a certain organization, a company may be deemed well prepared in terms of Web presence, design, and infrastructure, but may have overlooked the need to train and prepare its staff for the demands of e-business. Staff training is a key ingredient of e-business readiness—at least as critical as technical skills, written policies, or strategies. E-business needs to begin with solid technical preparation, but in the long run it requires sufficient staff support, involvement, and understanding. In-depth coverage means that each dimension or component of an issue is studied and analyzed fully.

3) Lack of Communication:
Lack of communication is a major source of ignorance. Communication narrows the distance between the various elements of the problem in question. Lack of communication originates in factors such as failure to contact the stakeholders in a business problem; not using effective communication techniques; or not being able to carry out an efficient communication process. The effects of a lack of communication can be summarized as follows:

- Ignorance of lack of sources. Communication is the primary method for acquiring data from existing or prospective sources. Lack of communication reduces or omits sources of information.

- Extracontextual ignorance. Communication can ease tension between conflicting parties and improve common understanding. This is beneficial when gathering reliable data. Furthermore, the more that data resides outside an organizational context, the more difficult it is to obtain. Communication encourages an amicable and mutually accessible environment in which differences can be viewed as sources of data and knowledge. This also creates opportunities for transferring and exchanging data.

- Ignorance of lack of communication channels. Without appropriate communication channels, it is often difficult to deliver timely or on-time data. Late data delivery can make the problem-solving process less effective. This is especially important in achieving competitive advantage and responding to urgent situations.

- Differentiation ignorance. The current trend in business is to learn from competitors and to seek partnerships to achieve common goals. It is known that integrative approaches facilitate more effective problem solving roles in terms of gathering reliable data, compared to non integrative, differentiating approaches. Communication is the cornerstone for facilitating any integrative process.

4) Interorganizational Ignorance
The value of knowledge stems from its usability and adaptability, not from its mere existence. To be valuable, information or data must add value to an organization and to its problem-solving processes. Otherwise, it is tantamount to a form of double ignorance in which people do not know what they know but assume that they do (or, they do not know that they do not know). This can make knowledge expensive if one is in possession of unused data, or make an organization a victim of knowledge utilization delays that result from a lack of awareness or ignorance of ignorance. Knowledge-based ignorance can hide weakness behind apparent strength and business sickness behind an apparently healthy organization. This source of ignorance has many manifestations and degrees and even low levels can be damaging and costly. Consider, the sales transactions that a department store conducts with its customers on a daily basis. If this accumulated daily data is only stored until the end of the year and then used solely for purposes related to taxes and inventory, the opportunity to apply such critical information may have been permanently lost. For example, applied in a timely fashion, the daily data could have been utilized for a variety of purposes—including tracking inventory in order to avoid going below a repurchase point. If data is not processed on time for such tracking purposes, business sales can suffer because of out-of-stock occurrences on key saleable items, possibly resulting in a loss of strategic clients, alliances, or business partners. Ignorance at the inventory level can block a business from operating, partially or totally in a very short time. Therefore, even though this type of ignorance is associated with a low level of the structured business process, lack of use has a potential for major impact and so represents a serious risk. Studying customer behavior in a manner that measures customer requirements on an accurate, predictive basis is another example of the applicability of such low-level data. Without analyzing daily sales data statistically, it may be impossible to cluster customers, products, or sales points so that the store can prosper and maintain its competitive advantage. Ignorance at the customer satisfaction level may not preclude a business from continuing operation, but it may put such a business at a competitive disadvantage. The level of risk of ignorance in this situation may be moderate, but the long-term effects may be critical. This situation belongs to the branch-level management class of business processes. Conducting ongoing cost-benefit analysis to measure financial performance and to control share profit is also an important issue. Absence of knowledge critical to supporting decision-making processes may prevent an...
organization from effectively supporting strategic management decisions. Such knowledge is of strategic value and can only be derived from daily transactional data. A lack of knowledge of what is happening on a given day may be minimal in terms of risk; however, in the long term, this may mask critical risk factors lurking behind the scene that can lead to business failure. Although many levels of ignorance are linked simply to lack of data, information, or knowledge, some ignorance can be attributed to vague, surface, or unused knowledge. Examples include: on unprocessed data. Data that is not transformed into useful information in the right form, at the right time, and provided to the right people represents unprocessed data. Unprocessed data makes what we know less effective, but still expensive. Many organizations are excellent at gathering data, but fail to relate it to their problems because they do not convert it to other, more meaningful forms of information or knowledge.

UNUSED DATA. When data is not used to solve problems, it amounts to an absence of data. Unused data, regardless of its level of transformation or meaningfulness, merely represents an added cost created by careless business behavior. If this is related to data that has not been processed, it is a waste of time and money. If it is related to processed data known to be useful, then retaining this data without further examination or resolution is a problem and contributes to wasted time and resources. If data is unused due to lack of managerial commitment and despite the established value of the data, this transcends mere normal ignorance and rises to the level of culpable ignorance.

- Untailored data. Utilizing data effectively requires an accurate problem definition, just as medication makes no sense without a proper prior diagnosis. Thus, understanding the problem and the solution domain is as important as knowledge of the data.

- Vague data. Data may be too low quality to be considered for processing. This is a case of ignorance of the data that one has. Such data may be uncertain, unconfirmed, unclear, or undefined, or need proper translation or adequate clarification. If the data is processed despite its errors or uncertainties, unreliable outcomes and inefficiencies in decision-making result.

- Politically based ignorance. Organizational politics can play a destructive role in obtaining reliable data. If non-scientific, non-rational, or biased motivations are behind the selection of data, this may preclude obtaining critical data. Such politically selected data cannot be considered representative. The excluded data may contain contradicting facts, clarifying statistics, or a more complete picture. Purposely and biased ignorance of this type affects the problem-solving process negatively. There must be a legitimate and objective reason, not based on political or economic interest to justify exclusion of data. Biases are another kind of filter blocking accurate data acquisition; they represent a kind of color-blindness in viewing facts in which the interpretation of the data depends on whether it supports a position held in advance. This attitude inhibits seeing other viewpoints merely because they are the viewpoints of others.

- Technically based ignorance. This refers to the lack of reliable tools that enable us to see, understand, and interpret phenomena correctly. Ignorance is strongly tied to such lack of tool support. One cannot be expected to make sense of data without reliable tools. When tools are unavailable, one should anticipate that the data may not be processed at all; may not be processed on time; may not be processed accurately; may be lost or destroyed due to lack of storage tools; may be used only at lower levels of management; or may not be recognized as enabling decision-making processes.

- Statistically based ignorance. This refers to a failure to establish the right relationship between things in an interconnected environment. Ignorance is often not so much a failure to collect data, as a failure to explore the data or see how data is interconnected. For example, a change in organizational effectiveness that occurs in parallel with a newly adopted style of management may not be coincidental. Viewing data as isolated bits of information without making the effort to observe its correlations or interrelationships is a type of ignorance. The effectiveness of the problem-solving process strongly depends on the ability to observe, discover, or predict relationships between variables in an organizational context.

- Illusion-based ignorance. Data is not always transparent: it may mask deception, illusion, imagination, or tricks that create false impressions. Major national corporations have gone belly-up as the result of this kind of ignorance. In order to distinguish facts from illusions, caution must be exercised when viewing available data.

VI. CONCLUSION

This paper indicates the due concepts involved in diversification of software and the inherent challenges faced by software process modelers. Data for software development is huge task and the degree of ignorance must be tried to minimize for effective software modeling. We present a comprehensive mention of the different factors to be considered while using a framework to support software process development.

REFERENCES


