PUC

ISSN 0103-9741

Monografias em Ciência da Computação
n° 45/08

Software Process Based on Heuristics

Mariela Inés Cortés
Viviane Almeida Santos

Departamento de Informática

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO
RUA MARQUÊS DE SÃO VICENTE, 225 - CEP 22451-900
RIO DE JANEIRO – BRASIL

Monografias em Ciência da Computação, No. 45/08 ISSN: 0103-9741
Editor: Prof. Carlos José Pereira de Lucena October, 2008

Software Process Based on Heuristics

Mariela Inés Cortés *

Viviane Almeida Santos

Computer Science Department. State University of Ceará, Ceará, Brazil

{mariela, viviane.almeida}@larces.uece.br

Abstract. Software process reuse involves different aspects of the knowledge obtained from generic process models and previous successful projects. The benefit of reuse is reached by the definition of an effective and systematic process to specify, produce, classify, retrieve and adapt software artifacts for utilization in another context. In this work we present a formal approach for software process reuse to assist the definition and improvement of the organization's standard process. The Case-Based Reasoning (CBR) technology is used to manage the collective knowledge of the organization.

Keywords: Case-based reasoning, Software process reuse, Process improvement, Similarity measurement.

Resumo. O reuso de software envolve diferentes aspectos do conhecimento obtido a partir de processos e modelos e informações de projeto anteriores realizados com sucesso. Os benefícios do reuso são alcançados pela definição de um processo efetivo e sistemático para especificar, produzir, classificar, recuperar e adaptar artefatos de software para sua utilização em outros contextos. No presente trabalho é apresentada uma abordagem formal para reuso de processo de software de forma a assistir na definição e melhoria do processo padrão da organização. A tecnologia de Raciocínio Baseado em Casos (RBC) é utilizada para gerenciar o conhecimento coletivo da organização.

Palavras-chave: Raciocínio Baseado em Casos, Reuso de processo de software, Melhoria de processo, Calculo de similaridade.

* Visiting researcher at Departamento de Informática, PUC-Rio.

In charge of publications:

Rosane Teles Lins Castilho
Assessoria de Biblioteca, Documentação e Informação
PUC-Rio Departamento de Informática
Rua Marquês de São Vicente, 225 - Gávea
22451-900 Rio de Janeiro RJ Brasil
Tel. +55 21 3527-1516 Fax: +55 21 3527-1530
E-mail: bib-di@inf.puc-rio.br
Web site: http://bib-di.inf.puc-rio.br/techreports/

1 Introduction

Considering the forward dependency between the development process quality and the product quality, the deep knowledge of the activities involved in the process and their management are critical factors for the organizational success.

In high level, the software development process defines a formal sequence of activities for planning software development. This knowledge captures the guidelines to drive software development in a specific domain and/or context.

The definition of a process for software development is a complex task since it requires experience and combines the knowledge of diverse technological and social aspects. The utilization of standards for the process definition (ISO, 1996) (ISO 2006) (Chrissis et al., 2003) (Paulk, 1993) (Softex, 2006) is recommended in norms, processes and maturity models. However, the process model must be adapted to fit the organization characteristics.

Software process models describe the organization knowledge and, thus, models that enhance successful experiences must be disseminated and recommended for reutilization across the organization. This best practice is appointed by the standards (ISO, 2006) (PMI, 2004) that promote the establishment of reuse processes for historic data and organizational process's assets. The process consolidation is achieved through the systematic reuse in specific projects and the incremental capture of relevant experiences, looking for the continue improving.

The purpose of the process reuse technology is to support the process definition and improving on the basis of standard processes, according to norms and quality models, and learned experiences (Perry, 1996). Dynamic and context-depending aspects of the knowledge in software development turn the Case-Based Reasoning approach (CBR) (Kolodner, 1993) useful as it provides a broad support for the dynamic management of the organizational knowledge and continuous incremental learning necessary for the definition and improving of software development. The knowledge management involves classification, retrieval and reuse of process assets, acquired by the previous experience.

In this work we describe an approach for elaboration and reuse of the organizational standard process, on the basis of models, standards, quality norms, and relevant previous experience, in accordance with the organizational reality and characteristics. The proposed approach uses a repository of process assets, which can be reused in similar software project and instantiation of the standard process. The CBR technology is used for the management of the repository and the retrieval of assets.

This work is organized as follows: in Section 2 the CBR technology is briefly explained. In Section 3 the process reuse using CBR is presented. In Section 4 a case study is illustrated. Finally, final considerations are presented.

2 Case-Based Reasoning

The CBR technology solves problems in a specific situation (or new case under consulting), through previous similar situations (Pal and Shiu, 2004). A case comprises (Mille, 2006) a pair problem that describes the context of an actual case occurrence, and solution that presents the problem solution. Past cases are used to hint strategies to solve new similar problems, to adapt solutions, to alert for errors and to help in the interpretation of new situations.

A CBR system is composed by 4 basic elements (Kolodner, 1993): knowledge representation, similarity measure, adaptation and learning.

The knowledge representation consists on the description of the relevant information for the cases, in order to assess the reuse. In this sense, diverse formalisms can be used, e.g., graphs, attribute-value representations, objects, etc.

The similarity measure establishes the similarity degree between a base case and a new problem under consulting. This measure is based on a heuristic method (Pal and Shiu, 2004) that allows to identify its utility for a new problem. The utility of a case is an intuitive context sensitive concept, which depends on the specific objectives for the retrieval of cases from the repository. The retrieval process results in a set of ranked cases that can be useful to solve the actual problem, in partial or complete way. The ranking of cases is based on the global similarity measure.

The utility of base case to solve a problem is proportionally related to the effort required to adapt it to fit the specific context (Mille, 2006). When necessary, the adaptation process transforms the recovery solution into an appropriate one for the new problem. This process involves knowledge reuse in problems solutions along the knowledge transference from the previous case to the actual case.

The ability to learn from past experiences is inherent in a CBR system. The effective learning results in a set of methods to extract the relevant knowledge from the previous experience, then index this knowledge to assess their posterior reutilization, and finally integrate these cases in a manageable structure to build an empirical body of organizational knowledge. The continuous learning contributes to increase the system capacity to solve new problems, improving their interpretations, evolving to a powerful problem solver. Feedback about the soundness and effectiveness of the elaborated interpretations is required to improve the learning process of the system.

3 Process Reuse Approach

In the Integrated Platform for REasoning from CAses (INRECA) (Bergmann, 1999), the searching for solutions to solve new problems must be related with solutions for similar problems, that must be indexed into a cases-repository of the system. The CBR enhance this philosophy to solve problem in several knowledge areas. In the context of this work, the CBR technology is applied in reuse of software processes.

The proposed approach for reuse is presented in Figure 1. The main component is the Processes Assets Repository which is designed to store assets models for reuse and their attribute-value representations. This representation involves a set of relevant properties to describe each case, and the values for these properties including numeric, text, pre-defined terms, etc. The utility of a specific case from the repository in the context of a new case under consulting is enabled using this representation.

Considering that process models are abstract, their inclusion in the repository requires the existence of an instance in a specific case. The Search Engine uses CBR technology for retrieve similar cases through the similarity measurement on the basis of process and project features. Attribute-value representations must be defined for the new case, and for the base cases in the repository.

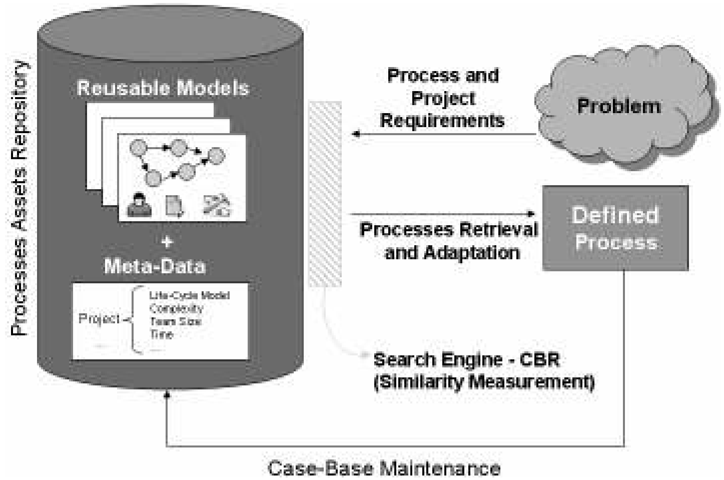

Figure 1. Approach for process reuse

The reutilization involves the adaptation of a previous solution for a similar case, using an appropriate method (Mille, 2006). After its adaptation and execution in the new project, the reused process instance is evaluated in order to examine their effectiveness and capture reuse information.

After that, the new instance of the model can be included into the repository, increasing the attribute-value representation for this model and contributing to the reduction of efforts in the next improving cycles.

3.1 Representation of Organizational Assets in the Repository

The reutilization of cases is enabled whenever the cases will be indexed and stored appropriately in the repository of process assets, in such a way to make possible its efficient retrieval. The suitable representation of the process assets is a critical factor for the success of the method, since the similarity degree for the correct retrieval of the cases is measured on the basis of this representation. The similarity concept consists of establishing an estimate of the utility of a previous case stored in the repository, in the context of the current case under consultation, on the basis of the observed similarity among the representations of both cases (Kolodner, 1993).

The similarity types are restrictions applied to the representation features, to establish its correspondence or co-occurrence among cases (Reis et al., 2001). The similarity types used in this work are described in Table 1.

Table 1. Similarity Types for the features representation

Similarity Types	Short	Possible Values
Numeric	**NUM**	Positive integer or real numbers
Qualitative for Fixed Items	**QFI**	Predefined Terms
Qualitative for Variable Items	**QVI**	Registered terms with possibility of new items

The similarity between cases is based on the comparison of the features in the representation and the corresponding values. In this sense, several studies related to the classification of the process assets for reuse in other contexts can be cited (Perry, 1996) (Reis et al., 2001) (Oliveira et al., 1999) (McManus, 1999) (Oliveira et al., 2006).

The classification proposal for the representation of the assets in the repository in this work is present in Table 2. The features had been organized in agreement to the target in process and project features.

3.2. Retrieval Process

The most appropriate solution for the current problem is retrieved from the repository through similarity measurement. The focus of this measure is to establish the degree of similarity between the current case and the cases in the repository (case-base) (Kolodner, 1993). The greatest value in this measurement indicates greater similarity between the cases.

In CBR, several techniques can be applied for data retrieval. In (Pal and Shiu, 2004) the algorithm to calculate the similarity is based on k-NN technique, where the global similarity (SIM) between two cases (a and b) is defined by the weighted sum of the local similarities (sim_j) for each feature (A_j).

$$SIM(a,b) = \sum_{j=1}^{n} w_j \times sim_j(A_j(a), A_j(b))$$

The weight (w_j) reflects the relevance of a feature (A_j) concerning the similarity of cases. This factor is determined by the user and is measured by the values: High (100), Medium (50) and Low (10). The features considered more important for the problem resolution from the user's viewpoint, possess higher weights.

The base cases with greater similarity measurements are considered as sufficiently similar and proposed to the user as reuse candidates. Note that if the same weight is assigned to all the attributes, the base case that attends the greater number of features must be the suggested one.

The local similarity is calculated in accordance with the similarity type of each feature (Table 2) and considers the computation of distance (d_j) between each feature values in the cases a and b:

$$sim_j = \frac{1}{1 + d_j(a,b)}$$

This measurement must be normalized (Ricci et al., 2002) to avoid over influence of a metric by the great range of values of the attributes. The normalization process uses smallest and greatest values in the repository to linearly produce values between 0 and 1.

The distance between two features of numeric similarity type (NUM) is calculated on the basis of a proportionality relation between the values. Thus, the local similarity in this case is expressed as:

$$d_j(a,b) = \left| \left(\frac{A_j(a) - \min(A_j)}{\max(A_j) - \min(A_j)} \right) - \left(\frac{A_j(b) - \min(A_j)}{\max(A_j) - \min(A_j)} \right) \right|$$

For the Qualitative for Fixed Items (QFI) the distance is calculated by establishing a proportion between values through the fixed items: High/Large (9), Medium (6) and Low/Small (3). The expression for the local similarity for QFI features is:

$$d_j(a,b) = \left| \left(\frac{A_j(a) - 3}{9 - 3} \right) - \left(\frac{A_j(b) - 3}{9 - 3} \right) \right|$$

Thus, the expression can be resumed to:

$$d_j(a,b) = \left| \left(\frac{A_j(a) - 3}{6} \right) - \left(\frac{A_j(b) - 3}{6} \right) \right|$$

Finally, to calculate the distance between features of Qualitative for Variable Items (QVI) is used a taxonomy to hierarchically represent the relationships among the terms (Figure A1), where s is the distance (jumps) between Aj(a) and Aj(b) in the taxonomy.

$$d_j(a,b) = \frac{s}{10}$$

The measurement for a new case may require the inclusion of new terms in the taxonomy.

3.3. Adaptation Process

If necessary, the adaptation of the retrieved process for reuse in the new case can be realized following different approaches (Pal and Shiu, 2004). In this work it is used the transformational approach, that applies specific knowledge to manually transform the base-case solution into a new solution to attend the current needs. In this context, the values for local similarities in the different models can be used in order to assist the user in the decision making during the adaptation process. However, the adaptation process is not the focus of the present work.

Table 2. Representation of the assets in the repository

Scope	*j*	Feature	Description	Similarity Type
Project	1	Life-Cycle Model	Project life-cycle model, such as Cascade, Iterative and Incremental, Evolutionary, Spiral, etc.	**QVI**
	2	Complexity	Project complexity: High (the project include critical and advanced functionalities), Medium (the project needs analysis, but has feasible functionalities), Low (the project has simple functionalities).	**QFI**
	3	Size	Project size regarding the functionalities quantity: Large, Medium or Small.	**QFI**
	4	Team Size	Project integrants number.	**NUM**
	5	Time	Project duration in months.	**NUM**
	6	Software Engineering (SE) Knowledge	Knowledge level in Software Engineering: High (theory e practical), Medium (theory only), Low (none knowledge).	**QFI**
	7	Development Paradigm	Project development paradigm, such as Structured, Object Oriented, etc.).	**QVI**
Process	8	Development Model	Software development models, like RUP, XP, SCRUM, etc.	**QVI**
	9	Maturity Model	Maturity model, for example, CMMI, MPS.BR, etc.	**QVI**
	10	Maturity Level	Specific maturity level related to the maturity model specified previously. It can be, for example, 1 to 5 (CMMI and ISO/IEC 15504) or G to A (MPS.BR).	**QVI**
	11	Complexity	Process complexity based on the maturity levels: High (advanced levels), Medium (intermediary levels), Low (low levels).	**QFI**
	12	Process	Specific processes, such as Requirements Management, Project Planning, Quality Assurance, Configuration Management, etc.	**QVI**
	13	Experience on Process Usage	Team's experience on software process usage: High (process used in more than 15 projects), Medium (process used in a range of 5 to 14 projects), Low (0 to 4 projects).	**QFI**
	14	Success Level	This result (1 to 10) represents an indicator of the degree of organizational satisfaction about the adopted process.	**NUM**

3.4. Feedback

The incremental evolution based on feedback is essential for continuously building and improving the process development know-how. The learning process in the CBR system is done through the feedback about the performance of the new process model instance, when the project is closed. At this moment, the effectiveness of the reused process is evaluated by the user before the storage in the repository.

In this sense, the assets representation in the repository includes the process feature Success Level to reflect this reuse information. All satisfactory or not satisfactory new experiences must be added to the repository as organization's learned lessons.

This information can influence the posterior adoption of the process model in other contexts, contributing for the adoption of an organization standard process in the search for the continuous improvement.

4 Case Study

In this section, a case study is presented to illustrate the approach for process reuse. In this context, the description of a new project is detailed assigning values to the wished attributes for process and project. Note that the process for the standard process definition and the instantiation for an already defined process is the same.

In the table below the definition of the desired features for the new case is presented. The Scope and Feature columns represent the feature's classification as presented in Table 2. The Weight and Value columns refer to properties of the new project, about the relevance and value for the feature, respectively, from the user viewpoint.

To illustrate the recovery process, three successful process models and its respective representations are used: ProGer [Rouiller, 2003], RUP for Small Teams (RUP-ST) (Pollice et al., 2004) (Kruchten and Kroll, 2003) and Dynamic CMM (D-CMM) (Orci and Laryd, 2000). It is important to stand out that the repository of process assets must be wide and diversified in order to take care of the most diverse situations.

For each process model, the global similarity is calculated on the basis of their representation in order to determine and retrieve from the repository the most adherent case to fit the new case. The local similarity for Feature is calculated in accordance with the similarity type, as referred to Section 3.2, and is described in the Comparison column. The product of this value times the Weight, presented in Table 3, determines the Local Similarity (LS). Finally, the addition of all local similarities is presented in the column Global Similarity.

Table 3. Feature definition for the case study

Scope	Feature	Weight	Value
Project	Life-Cycle Model	Medium	Spiral
	Complexity	Low	Medium
	Size	Medium	Medium
	Team	Medium	5
	Time	Low	6
	SE Knowledge	Low	Medium
	Development Paradigm	High	O-O
Process	Development Model	Low	-
	Maturity Model	Low	-
	Maturity Level	Low	-
	Complexity	Medium	Low
	Process	Medium	Project Management
	Experience on process usage	Low	Low

Tables 4, 5 and 6 present the values for each feature for projects based on the ProGer model, RUP-ST and D-CMM, respectively. After that, the global similarity about the current case can be calculated. The taxonomies used to calculate the similarities for QVI features are illustrated in Figure A1 in the Appendix.

Table 4. Global similarity about ProGer

Scope	Feature	Case-Base	Comparison	LS
Project	Life-Cycle Model	PMBOK Phases	0,8	40
	Complexity	Medium	1	10
	Size	Medium	1	50
	Team	4	0,85	42,5
	Time	7	0,5	5
	SE Knowledge	Medium	1	10
	Development Paradigm	O-O	1	100
Process	Development Model	PMBOK	0	0
	Maturity Model	-	0	0
	Maturity Level	-	0	0
	Complexity	Low	1	50
	Process	Project Management	1	50
	Experience on process usage	Low	1	10
Global Similarity				367,5

Table 5. Global similarity about RUP-ST

Scope	Feature	Case-Base	Comparison	LS
Project	Life-Cycle Model	Iterative\ Incremental	0,8	40
	Complexity	Medium	1	10
	Size	Medium	1	50
	Team	5	1	50
	Time	5	0,5	5
	SE Knowledge	High	0,5	5
	Development Paradigm	O-O	1	100
Process	Development Model	RUP	0	0
	Maturity Model	-	0	0
	Maturity Level	-	0	0
	Complexity	Medium	0,5	25
	Process	Project Management	1	50
	Experience on process usage	Low	1	10
Global Similarity				345

Table 6. Global similarity about D-CMM

Scope	Feature	Case-Base	Comparison	LS
Project	Life-Cycle Model	Iterative	0,9	45
	Complexity	Medium	1	0
	Size	Medium	1	0
	Team	10	0,55	27,5
	Time	7	0,5	5
	SE Knowledge	High	0,5	5
	Development Paradigm	O-O	1	100
Process	Development Model	-	0	0
	Maturity Model	SW-CMM	0	0
	Maturity Level	2	0	0
	Complexity	Medium	0,5	25
	Process	Project Management	1	50
	Experience on process usage	Medium	0,5	5
Global Similarity				262,5

Although all models are oriented to small organizations, the results for the similarity measurement indicate the Case-Base ProGer= 367,5 as the most adherent case because of the greatest global similarity, involving minor efforts in the adaptation process for the new situation. A further analysis about the local similarity results can be used to guide the user during the adaptation process. In this sense, the composition of feature values in a new model, in order to optimize the global similarity, is an undergoing study.

The process evolution and improvement is realized along its adaptation, reuse, performance evaluation and reincorporation into the repository. Favorable reuse evaluations along diverse projects can guide the adoption of the organization's standard-process.

5 Final Considerations

The purpose of the process reuse is to increase the quality and productivity on the basis of norms and quality models. This approach promoves the reutilization of process assets as a start point for the elaboration of a standard process to meet the organizational needs.

The proposed approach for process reuse presented in this work can be used to assist in the definition and instantiation of software processes. This approach is based on Case-Based Reasoning. It supplies a mechanism for the representation of cases (models of concrete experiences) in the assets repository. The cases were classified according to a set of features to allow an efficient retrieval.

The classification criteria described in this work contribute to the definition of the relevant aspects to be considered in the elaboration or instantiation of standard processes. The retrieval from the repository was described and examples of similarity measurements were presented in a case-study.

This approach foresees the continuous improvement of the process through the permanent feedback to the repository involving the incorporation of learned lessons with the adopted process. The learning capability of CBR systems contribute to the adoption of better and more efficient solutions (avoiding the recurrence of previous errors) and converges to the establishment of the continuous improvement for the organizational processes.

Currently, an approach for the project of a new process in repository is being elaborated on the basis of the combination of desired features, in order to optimize the global similarity. An analog strategy for the process adaptation can be used through the suggestion of process assets using the local similarity for specific features.

The measurement presented in this work was carried through manually; however, a management tool to support the process of reuse approach is currently under development.

References

BECK, K., 2004. Extreme Programming Explained: Embrace Change. Pearson.

BERGMANN, R. et al., 1999. Developing Industrial Case-Based Reasoning Applications. Lecture Notes in Artificial Intelligence, v. 1612. Berlin: Springer-Verlag.

CHRISSIS, M. KONRAD, M., AND SHRUM, S., 2003. CMMI guidelines for process integration and product improvement. Addison-Wesley.

MILLE, A., 2006. From case-based reasoning to traces-based reasoning. Annual Reviews in Control 30(2):223-232. ELSEVIER. ISSN 1367-5788.

KOLODNER, J., 1993. Case-Based Reasoning. Publisher Morgan Kaufmann.

KRUCHTEN P. AND KROLL P., 2003. The Rational Unified Process Made Easy. Addison-Wesley.

MCMANUS, J., 1999. How does Software Quality Assurance Fit In?. Handbook of Software Quality Assurance, 3 ed. Prentice Hall.

OLIVEIRA, S. R. B., VASCONCELOS, A. M. L., 2006. A Continuous Improvement Model in ImPProS. In 30th Annual International Computer Software and Applications Conference. Proceedings on COMPSAC Fast Abstract Session. Chicago, EUA.

OLIVEIRA, K., GALLOTA, C., ROCHA, A. R. et al., 1999. Defining and Building Domain-Oriented Software Development Environments. In ICSSEA'99, 12th International Conference Software & Systems Engineering and their Applications. Paris, France.

ORCI, T. AND LARYD, A., 2000. Dynamic CMM for small organizations. In ASSE’00, 1st Argentine Symposium on Software Engineering. Pages 133 to 149. Argentina.

Pal, S. and Shiu, S., 2004. Foundation of soft case based reasoning. Wiley series in intelligent systems, 5th edition.

PAULK, M. et al, 1993. Capability Maturity Model for Software. Pittsburg: SEI, Carnegie Mellon University, version 1.1. Available in: http://www.sei.cmu.edu/pubs/documents/93.reports/pdf/ 93tr024.pdf edition.

PERRY, D., 1996. Practical Issues in Process Reuse. In ISPW, International Software Process Workshop. IEEE Computer Society Press. France.

PMI Project Management Institute, 2004. A Guide to the Project Management Body of Knowledge: PMBOK Guide. PMI, 3rd. edition.

POLLICE, G., AUGUSTINE, L., LOWE, C., AND MADHUR, J., 2004. Software development for small teams - a rup centric approach. Addison-Wesley.

PRESSMAN, R., 2002. Engenharia de Software, 5th ed. Ed. McGraw-Hill.

REIS, R. Q., REIS, C.A.L., NUNES, D.J., 2001. Automated Support for Software Process Reuse: Requirements and Early Experiences with the APSEE model. In 7th International Workshop on Groupware. IEEE Computer Society Press. Darmstadt, Germany.

RICCI, F., ARSLAN, B., MIRZADEH, N., VENTURINI, A., 2002. Detailed Descriptions of CBR Methodologies. Information Society Technologies. Available in: http://dietorecs.itc.it/ PubDeliverables/D4.1-V1.pdf.

ROUILLER, A., 2003. Gerenciamento de Projetos de Software para Empresas de Pequeno Porte. Tese de Doutorado. UFPE.

Softex, 2006. Guia Geral MR-MPS (Versão 1.1). Available in: http://www.softex.br/mpsbr/ _guias/MPS.BR_ Guia_Geral_V1.1.pdf.

International Organization for Standardization, 1994. ISO 9001 Quality System – Model for Quality Assurance in Design, Development, Production, Installation, and Servicing, Revised Edition. Geneva, Switzerland.

The International Organization for Standardization and the International Electrotechnical Commission, 1996. Standard for Information Technology—Software Life Cycle Processes. Geneva, Switzerland.

The International Organization for Standardization and the International Electrotechnical Commission, 2006. ISO/IEC 15504 Information Technology Process Assessment Part 5: An exemplar Process Assessment Model.

APPENDIX

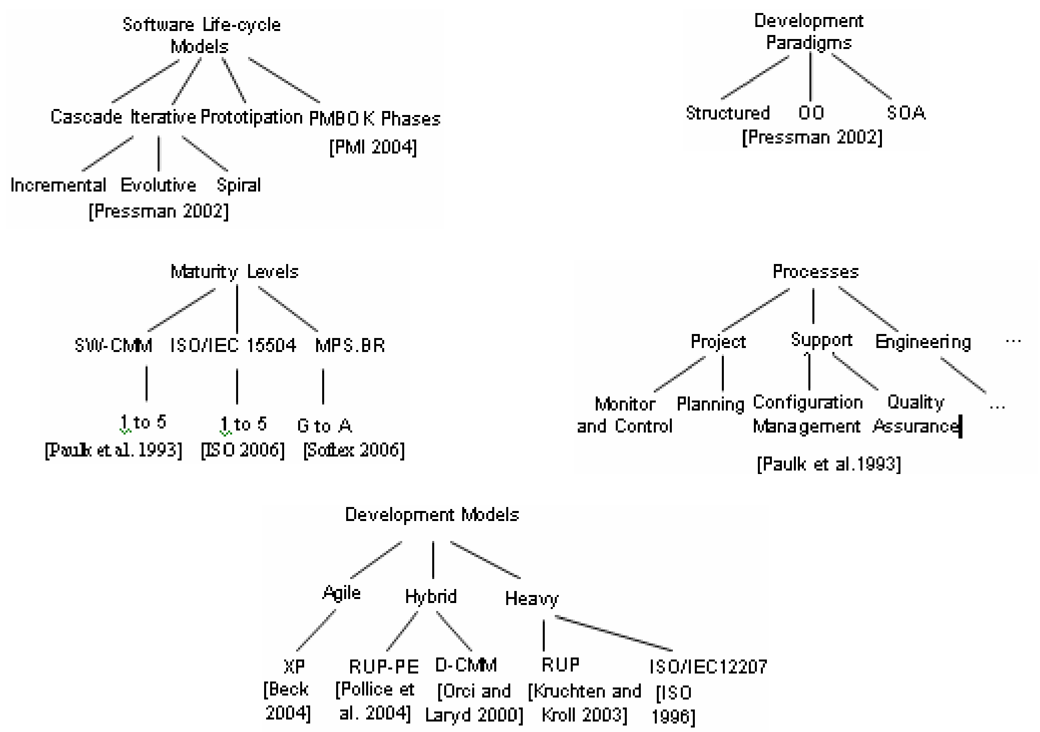

Figure A1. Taxonomies for QVI features