
Farhad Khosrojerdi*, Stéphane Gagnon*, Raul Valverde**

* Université du Québec en Outaouais (UQO), Gatineau (Québec), Canada,
** Concordia University, Montreal (Québec), Canada,

Abstract - This paper proposes a framework for managing Project Quality Management (PQM) processes of software development projects related to photovoltaic (PV) system design. The International Organization for Standardization (ISO) quality evaluation model, Software Product Quality Requirements and Evaluation (SQuaRE) standard, is used to determine quality characteristics and quality metrics of the software. This work presents the following contributions: I) defining quality characteristics associated with a PV design software using the SQuaRE standard model, II) adding the proposed framework as a tool and technique which is used by practitioners following the global standard book for project managers, A Guide to the Project Management Body of Knowledge (PMBOK), and III) Identifying quality measures and sub-characteristics of a PV design software. The presented model can be employed for simulation-based and/or model-based software products in various technical fields and engineering.

Keywords— software quality evaluation; quality model; software development projects; renewable energy projects; PMBOK; SQuaRE standard;

I. Introduction

Globally, the contribution of PV systems to energy generation was approximately 14,000 MW in 2010 and is expected to be 70,000 MW in 2020 [1]. In Canada, the domestic market has been
growing on average at about 26% per year since 1993 and about 48% since 2000 [2]. Due to growing usage of solar energy and PV system installations, there are several online software products that provide designing, simulation, analysis, and calculation of the power produced. Residential and commercial applications of PV systems have motivated information technology (IT) and utility companies to create PV design software packages. These customized software products or applications estimate the energy production and the cost for the PV system installation.

The goal for software engineers and project managers in this field is to develop high quality PV design software products.

PV design software users can rely on output reports of a PV design program that is produced based on state-of-the-art technical approaches related to PV model and simulation. Some of these applications, such as PV-LIB, are open source and allow for an open assessment of software quality [3-6]. This helps prevent excessive cost to overcome poor software quality during development [7].

Usually, software development projects are initiated to modify one or different characteristic(s) of a software and improve its quality. For a project management team, it is essential to be able to describe various characteristics of a project deliverable(s). Quality characteristics and quality metrics of a software help the project team to plan, execute, and monitor quality of performing a software development project. Only in this matter, a product that is delivered at the end of a project meets quality requirements defined by stakeholders. The project management global standard’s guidebook (PMBOK) [8] presents guidelines for managing projects regardless of their context. PMBOK defines processes in each of eight knowledge areas in project management. These processes include inputs, outputs, and tools and techniques that enable a project team to manage activities correlated to each phase of the project. In this study, we focus on the activities that identify and guarantee the quality of a software development project and the project deliverable, a PV design and simulation software product.
A quality tool or model can assure best quality evaluation outcomes [9] and provides a framework to validate quality of the software’s outputs. Use of quality models enables project managers to outline quality characteristics of a product and manage activities required for the software quality assurance. Quality model is defined as “a set of characteristics, and of relationships between them, which provides a framework for specifying quality requirements and evaluating quality” [10]. Software quality models have been developed to be used for evaluating a software. Numerous quality models have been developed over the last 40 years. Ambiguities in the evaluation methods [11], uncertainty and vagueness about selected criteria [12], and difficulties in defining inputs, outputs, and metrics [13] cause complications applying these models. Therefore, managing software development projects, herein PV designing, would be problematic even more.

To overcome these problems, we present a quality model defining quality characteristics of a PV design software. Two important technical characteristics of a PV system model are identified as quality characteristics of a PV design related software. In addition, the quality sub-characteristics and the quality metrics based on SQuaRE standard are defined. Obtaining quality measures assist a project management team to utilize them for: I) the process of identifying quality requirements in planning phase, II) the process of implementing quality management plan in the execution phase, and III) the process of controlling and monitoring the quality in the executing phase of the project life cycle. In addition to specifying quality measures, we propose the use of an Adaptive Neuro-Fuzzy Inference System (ANFIS) methodology to analyze linguistic reviews and expert’s judgement of the quality evaluations. The ANFIS technique constitutes a fast and appropriate systematic solution to the PV system even with inaccurate quality measurement data.

The rest of this paper is structured as follows: section II provides a literature review describing quality models. PQM processes and PV technical factors used in this study are described in section III. In section IV, quality measures for a PV design and simulation software is defined. In section V, the ANFIS method is applied on random quantified data to demonstrate how real-world data can be used for quality measurement of a specific characteristic of the software. Finally, a conclusion is made in section VI.
II. Quality Models

A. McCall model
This model defines software product qualities as a hierarchy of factors, criteria, and metrics [14]. Eleven product quality features into three perspectives are presented as: I) product review (maintenance, flexibility, and testing), II) product operation (correct, reliable, efficient, integrity, and usability), and III) product transition (portability, reusability, and interoperability). The problems in the model involve values associate with combining distinctive metrics, as they are based on “Yes” or “No” responses [15, 16].

B. Boehm model
Boehm’s model proposes different identification and classification of quality characteristics and presents metric automated evaluation [7]. Although it introduces a metric providing quantitative measurement of the degree to which a program has the associated characteristics, it is performed for specific program. Besides, the process of definitions of quality attributes are arbitrary [15].

C. Dromey model
This model establishes a measurement for including a software property in a model [15]. It argues that hierarchical models are vague in their definitions of lower levels. The quality evaluation is performed unlikely for each product in a dynamic fashion. However, there is no discussion of how this can be implemented [16].

D. FURPS model
In this model the quality characteristics are divided into functional requirements (RF), defined by the inputs/outputs, and non-functional requirements (NF) including usability (U), reliability (R), performance (P), and product support (S) [17]. The drawback of the model is that major quality characteristics are not identified herein [16].

E. ISO/IEC Standard series
In 1991, the ISO published its first international consensus on the terminology for the quality characteristics for software product evaluation (ISO 9126:1991). From 2001 to 2004, the ISO published an expanded version, containing both the ISO quality models and inventories of proposed measures for these models.

One of the main objectives of (and differences between) the SQuaRE series of standards and the current ISO 9126 series of standards is the coordination and harmonization of its contents with ISO 15939 [21]. The set of goals or quality characteristics can be the same or similar to the one defined in ISO/IEC 9126. Such a definition helps in evaluating the quality of software, but gives no guidance on how to construct a high quality software product.

Table 1 depicts quality measures for a PV design and simulation software. The presented quality metrics may be used for software quality evaluation of a PV software from an end-user’s point of view. The quality characteristics represent various quality criteria that a user expects from a PV design application. From technical perspectives, only Functional Suitability demonstrates technical quality dimensions associated with a PV performance model.

<table>
<thead>
<tr>
<th>Quality characteristic</th>
<th>Quality sub-characteristic</th>
<th>Quality measure name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Functional Suitability</td>
<td>Functional completeness</td>
<td>Functional coverage</td>
</tr>
<tr>
<td></td>
<td>Functional Correctness</td>
<td>Functional correctness</td>
</tr>
<tr>
<td></td>
<td>Functional Appropriateness</td>
<td>Functional appropriateness of usage objective</td>
</tr>
<tr>
<td></td>
<td>Co-existence</td>
<td>Co-existence with other products</td>
</tr>
<tr>
<td>Compatible</td>
<td>Interoperability</td>
<td>Data format exchangeability</td>
</tr>
<tr>
<td></td>
<td></td>
<td>External interface adequacy</td>
</tr>
<tr>
<td>Security</td>
<td>Confidentiality</td>
<td>Access controllability</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------</td>
<td>------------------------</td>
</tr>
<tr>
<td></td>
<td>Integrity</td>
<td>Data integrity</td>
</tr>
<tr>
<td></td>
<td>Accountability</td>
<td>User audit trail completeness</td>
</tr>
<tr>
<td></td>
<td>Authenticity</td>
<td>Authentication mechanisms sufficiency</td>
</tr>
<tr>
<td>Usability</td>
<td>Appropriateness recognisability</td>
<td>Description completeness</td>
</tr>
<tr>
<td></td>
<td>Learnability</td>
<td>User guidance completeness</td>
</tr>
<tr>
<td></td>
<td>Operability</td>
<td>Operational consistency</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Functional customizability</td>
</tr>
<tr>
<td></td>
<td></td>
<td>User interface customizability</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Undo capability</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Understandable categorization of information</td>
</tr>
<tr>
<td></td>
<td>User error protection</td>
<td>Avoidance of user operation error</td>
</tr>
<tr>
<td></td>
<td>User interface aesthetics</td>
<td>Appearance aesthetics of user interface</td>
</tr>
<tr>
<td>Reliability</td>
<td>Maturity</td>
<td>Fault correction</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Test coverage</td>
</tr>
<tr>
<td></td>
<td>Availability</td>
<td>System availability</td>
</tr>
<tr>
<td></td>
<td>Fault tolerance</td>
<td>Failure avoidance</td>
</tr>
<tr>
<td>Maintainability</td>
<td>Modularity</td>
<td>Coupling of components</td>
</tr>
<tr>
<td></td>
<td>Reusability</td>
<td>Reusability of assets</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Coding rules conformity</td>
</tr>
<tr>
<td></td>
<td>Analysability</td>
<td>System log completeness</td>
</tr>
<tr>
<td></td>
<td>Modifiability</td>
<td>Modification efficiency</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Modification correctness</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Modification capability</td>
</tr>
<tr>
<td></td>
<td>Testability</td>
<td>Test function completeness</td>
</tr>
<tr>
<td>Performance efficiency</td>
<td>Time Behaviour</td>
<td>Mean response time</td>
</tr>
<tr>
<td></td>
<td>Resource utilization</td>
<td>Mean processor utilization</td>
</tr>
</tbody>
</table>
III. PQM Processes for a Software Development Project

In order to meet stakeholders’ objectives, the PQM includes the processes for incorporating the required quality regarding planning, managing, and controlling [8]. As defined in PMBOK: “Plan Quality Management is the process of identifying requirements and/or standards for the project and its deliverables and documenting how the project will demonstrate compliance with quality requirements and/or standards. Manage Quality is the process of translating the quality management plan into executable quality activities that incorporate the organization’s quality policies into the project. Control Quality is the process of monitoring and recording the results of executing the quality management activity to assess performance and ensure the project outputs are complete, correct, and meet customer expectations.”

Inputs, outputs, tools and techniques that can be used to manage the processes of quality in a project are shown in PMBOK [8].

Considering the technical characteristics of a PV design software and the PQM processes provided in PMBOK, the inputs, the outputs, the tools and the techniques are defined for the PV software development project (Table 2). As indicated, the SQuaRE-based quality model is added to the techniques that can be applied in plan quality management process and control quality process.

<table>
<thead>
<tr>
<th>PQM processes</th>
<th>Inputs</th>
<th>Tools and Techniques</th>
<th>Outputs</th>
</tr>
</thead>
</table>
| **Plan Quality Management** | 1. Project charter  
2. Project management plan  
4. Organizational process assets | 1. Expert judgement  
2. Data gathering  
3. Data Analysis  
4. Test and inspection planning  
5. SQuaRE-based Quality model | 1. Quality management plan  
2. Quality metrics  
3. Project management plan updates  
4. Project documents updates |
| **Manage Quality**       | 1. Project management plan  
2. Project documents  
3. Organizational process assets | 1. Data gathering  
2. Data analysis  
3. Decision making  
4. Audits  
5. Problem solving | 1. Quality reports  
2. Test and evaluation documents  
3. Change request  
4. Project management plan updates  
5. Project documents updates |
| **Control Quality**      | 1. Project management plan  
2. Project documents  
3. Approved change requests  
4. Deliverables  
5. Work performance data  
6. Organizational process assets | 1. Data gathering  
2. Data analysis  
3. Inspections  
4. Testing/product evaluations  
5. SQuaRE-based Quality model | 1. Quality control measurements  
2. Verified deliverables  
3. Work performance information  
4. Change requests  
5. Project management plan updates  
6. Project documents updates |
This quality measurement framework may be used by software and systems engineers to ensure that PV design applications fulfill all requirements. Whether functional or non-functional, PV design requirements imply close attention to PV mathematical modeling [22, 23], along with various designs of PV integration within smart renewable energy projects [24-26]. These design parameters must also be submitted to optimization techniques, to ensure accurate information output from PV design software, especially for power forecasting tasks and load balancing within grid-connected systems [27-31].

IV. Quality measures for a PV design and simulation software

A. Technical Characteristics of a PV System

A PV cell model, representing its physical behaviors under all operating conditions, is needed to be able to simulate the PV cell [32]. An electrical circuit model could predict variations of I-V and P-V curves to the ambient conditions and environmental factors. The calculation of the model parameters including the values of components and variables of the model describe the physical reactions of the PV cell to the external factors. Figure 1 depicts the overall topology of a PV system and its power conversion concept.

In power conversion in a PV system, the most important part is operating at maximum power point in different environmental and ambient conditions. Partial shading conditions and performing under various temperature require a control system for the PV system, known as maximum power point tracking (MPPT) controller [33, 34]. There are various MPPT approaches and algorithms used for a MPPT controller to generate maximum output power [35].
Hence, technical characteristics of a PV simulation rely on considering the two important technical factors, I) PV model used and II) MPPT technique.

A. **Quality metrics for PV design software evaluation**

The quality of a PV design software is the degree to which it satisfies the implied needs of users. These stated needs are presented in the SQuaRE standard by quality models [10]. Based on the standard, the measurable quality-related technical features of a PV design software are called properties to quantify and can be associated with quality measures. According to ISO/IEC 25023, “A measurement method is a logical sequence of operations used to quantify properties with respect to a special scale.” Applying a measurement method results in obtaining quality measure elements. A measurement function, which is capable of combining quality measure elements, can quantify quality characteristics and sub-characteristics. The outcome of applying a measurement function is called quality measure [10].

Table 3 demonstrates quality measures for a PV design and simulation software. The quality characteristics represent various quality criteria that a user expects from a PV design application. From technical perspective, the quality characteristic of Functional Suitability demonstrates accurate functionality of the power calculated by the PV design software. It means that the estimated output power for a system in certain time and location can be reliable.

These quality measures are identified through the following steps:

1) The information and data are gathered from the manual, the technical reference, and the website.

2) The PV system performance models are validated based on expert’s judgement.
3) The thresholds for the quality metrics are defined.

4) The quality of the metrics is measured based on the thresholds.

Based on ISO/IEC 25023, quality measures defined as: functional suitability, performance efficiency, compatibility, usability, reliability, security, maintainability, and portability. For our purposes concerning quality of the software in the term of its accuracy in designing PV systems and calculating the power produced, three quality characteristics are chosen: I) functional suitability, II) compatibility, and III) usability.

Table 3 Quality measures used for a PV design and simulation software/application

<table>
<thead>
<tr>
<th>Quality characteristic</th>
<th>Quality sub-characteristic</th>
<th>Quality measure name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Functional Suitability</td>
<td>Functional completeness</td>
<td>Functional coverage</td>
<td>What proportion of the specified functions has been implemented?</td>
</tr>
<tr>
<td></td>
<td>Functional Correctness</td>
<td>Functional correctness</td>
<td>What proportion of functions provides the correct results?</td>
</tr>
<tr>
<td></td>
<td>Functional Appropriateness</td>
<td>Functional appropriateness of usage objective</td>
<td>What proportion of the functions required by the user provides appropriate outcome to achieve a specific usage objective?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Functional appropriateness of system</td>
<td>What proportion of the functions required by the users to achieve their objectives provides appropriate outcome?</td>
</tr>
</tbody>
</table>

Applying quantitative analysis for the three quality sub-characteristics ultimately will evaluate quality of the functional suitability factor of the PV design application. The four quality metrics: I) functional coverage, II) functional correctness, III) functional appropriateness, and VI) functional appropriateness of system can be used as measures to quantify the functional suitability characteristics of the software. For simplicity, the functional appropriateness sub-characteristic is not measured in the proposed methodology.
Thus, the following questions describing the two quality measures, functional coverage and functional correctness, should be clarified and then be answered:

1. What proportion of the specified functions has been implemented?
2. What proportion of functions provides the correct results?

The type of PV performance model used and application of MPPT approaches validate the quality of the simulation used for the PV design software. In fact, these properties indicate the level of accuracy and functionality expected from a PV design software. The proportion and the degree of which the software deliver its functionality can be measured by these two metrics. Table 4 presents the final quality measures used for the ANFIS methodology. Ultimately, the following questions describing functional suitability of the software can be answered by the software designers and used by the project team:

1. What is the type of PV model used in the simulation?
2. Is a MPPT method applied?

<table>
<thead>
<tr>
<th>Quality characteristic</th>
<th>Quality sub-characteristic</th>
<th>Quality measure name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Functional Suitability</td>
<td>Functional completeness</td>
<td>Functional coverage</td>
<td>What is the type of PV model used in the simulation?</td>
</tr>
<tr>
<td></td>
<td>Functional Correctness</td>
<td>Functional correctness</td>
<td>Is a MPPT method applied?</td>
</tr>
</tbody>
</table>

V. Application of the ANFIS methodology

An ANN-based controller relies on the trained data instead of requiring prior knowledge about the PV parameters [36, 37]. The roles of FL and ANN methods in modeling, simulation, forecasting and control of PV systems are significant [38-40]. The ANFIS is a hybrid method combining neural networks and fuzzy logic. Employing the ANFIS method requires to define three important terms known as fuzzification, fuzzy rule base and defuzzification. Unlike the FL, the ANFIS can benefit from being trained from previous operational data. The Sugeno method has been employed in the proposed model in MATLAB simulation to ensure a better compatibility with the conventional ANFIS approach [38, 39, 41].
A. The fuzzification operation

The process of fuzzification and training the ANFIS model could be performed either by using experimental data or by producing an artificial set of data in MATLAB [41-45]. The process includes converting crispy PV performance and simulation values into fuzzy variables reflecting quality sub-characteristics of the PV software in several quality measurements. These operational conditions identify the quality characteristic and sub-characteristics of the PV system and the optimal quality evaluation points. Using experimental data [41, 45] validates the effectiveness of the ANFIS-based quality system and offers a reliable and effective training process for the evaluator.

As indicated in Table 4, the two main quality sub-characteristics, namely the Functional Completeness and the Functional Correctness, are determined as input variables of the control system reflecting Functional Suitability that affect values in the related quality measures. The output of the ANFIS are determined as the Quality Evaluation indicating the functionality of the PV system Fig. 2.

![Input membership functions and the ANFIS designer](image)

Fig. 2 Input membership functions and the ANFIS designer

B. The fuzzy rule base and the defuzzification

The fuzzy rule base describes the relationships and mapping between the output and input membership functions that are generated by the ANFIS. It reflects the previous knowledge obtained from the training data set or from the operational experiences and experts’ judgements to define the relationships between the membership functions. Fig 3 shows the rules identified for the ANFIS, based on the considerations stated based on PV performance systems controlled by appropriate MPPT algorithms. In the last step, the defuzzification, converts the fuzzy sets into the crispy sets.
As it is shown (Fig 4), quality of the software can be evaluated according to the two-quality metrics, Functional Coverage and Functional Correctness. The linguistic rules defining the relationships of the input quality measures to the output heavily depend on the expert’s judgement. By moving the red lines (Fig 4) which are indicating the quality values associated with the quality characteristics, the correlated quality evaluation number is shown in the rule viewer.

MATLAB Toolbox additionally provides a surface viewer illustrating the relationship between the two quality sub-characteristics and the result of quality evaluation (Fig 5); in fact, presents uncertainty around high quality metrics of Functional Correctness. It means that it is more difficult to evaluate a PV design software when the actual system and its related devices are not selected.
VI. Conclusion

To the best of our knowledge, the quality characteristics and sub-characteristics of a PV design software has not been investigated using the ISO/IEC SQuaRE standard. In addition, PQM terminology from PMBOK, can be utilized to help better manage the quality evaluation of PV design software. The ontology-based and fuzzy-reasoning approaches have also been suggested for dealing with a variety of technical software products, which shall be the focus of our future work. Finally, a common vocabulary and definitions introducing different PV design project inputs, tools and techniques, and outputs enable practitioners to employ appropriate applications.

To summarize, this paper presents the followings:

I) creating a tool and technique for PQM processes which is used by practitioners following the PMBOK, the global standard book for project managers.

II) The SQuaRE is used to evaluate quality of an industry-specific software.

III) The framework can be employed for simulation-based and/or model-based software products in different fields of engineering.

The proposed research focus can provide significant benefit to overall PV design software quality. It is argued that uncertainty and vagueness about selected criteria, and difficulties in defining inputs, outputs, and quality metrics as well as ambiguities in various evaluation methods, all make
it difficult to quantify simulation and model-based software products in various industries such as PV.

References


[42] T.Boutabba, S. Drid, and M. Benbouzid, "Maximum power point tracking control for photovoltaic system using adaptive neuro- fuzzy "ANFIS"," presented at the Eighth International Conference and Exhibition on Ecological Vehicles and Renewable Energies (EVER), Monaco, Mar 2013, 2013. Available: [https://hal.archives-ouvertes.fr/hal-00833734](https://hal.archives-ouvertes.fr/hal-00833734)

