Towards a Development of an Operational Process for Software Requirements: Case study application for Renewable Energy Software’s

Khalid T. Al-Sarayreh¹ and Kenza Meridji²

¹Department of Software Engineering, Hashemite University, Zarqa, 13115, Jordan
²Department of Software Engineering, University of Petra, Amman, 11196, Jordan
khalidt@hu.edu.jo, kmeridji@uop.edu.jo

Abstract

This research study introduces the operationalization process for the requirements activity of software engineering discipline to build renewable energy projects. It describes a new development approach composed of three phase’s methodology to undertake the operationalization process combined with the development of an operational model. A theoretical implementation is provided in addition to the development of a practical implementation on a standalone photovoltaic system case study. Moreover, the author’s expect that this will help in a better understanding of the principles of software engineering, in the implementation of these principles for the development of renewable energy projects as well as to contribute to build the research knowledge of software engineering foundations.

Keywords: Software engineering, Renewable energy (RE), Candidates software engineering principles, Operational model, Operationalizations process, operational goal

1. Introduction

Energy engineering has become a field of interests at all society levels, individual, industrial and government levels. It is new and rapidly growing area. It does include broad areas of engineering such as bio-engineering, civil-engineering, electrical-engineering, chemistry-engineering etc.

This new discipline also needs software to be able to conceive and to build high quality systems for energy modeling for renewable energy calculators that are reliable, secure and with high performance. On the other hand software engineering is described by IEEE 610.12-1990 (2001) as “The application of a systematic, disciplined, quantitative approach to the development, operation and maintenance of software, the application of engineering to software”.

Software is not the product of nature, in other long-established engineering disciplines engineers study the laws of nature by observation and seek to comprehend them. Nevertheless, in software engineering software projects are studied and observed. As a result, it is very challenging to discover the set of fundamental principles for software engineering due to software nature.

Energy engineering (En E) is a new emerging multi-disciplinary field of engineering. This area is expected to know an enormous growth. Energy field has become a national priority for government and thus becoming an area of a big concern and interest.

This new energy engineering discipline is concerned with the cycle of sourcing, assessing, designing, building, operating and improving the methods and systems. Subsequently, this will be used for the generation and the transmission of sustainable and renewable energy sources.
The main focus of energy engineering systems is energy efficiency issues, increasing relative systems performance as well as energy reliability, safety and security.

Energy engineering is a discipline that needs software for energy modeling, energy performance buildings and energy simulation etc. These software’s needs to be built with methodologies, techniques and tools that already exist in the new emerging engineering discipline that is software engineering. However, software engineering lacks the definition of its foundation as stated by Meridji, (2010). A research was conducted to define the fundamental principles for software engineering (FP’s) that conforms to engineering criteria; this research used the principles found in the literature as input, as well as a four phase’s methodology defined by Meridji & Abran, (2010). The output of this research was a definition of principles that conforms to engineering (FP’s). However, there was no description on how to operate them.

Currently, there is no development approach for renewable energy software’s using the selected nine fundamental principles form an engineering perspective and their operationalization.

For the purpose of this research, this set of nine fundamental principles from an engineering perspective have been selected and will be used in our development approach in order to be able to build high quality renewable energy software’s in the development process.

This paper is structured as follow: Section 2 presents an overview of the related work. Objectives of this research are described in Section 3. Section 4 presents the development approach for the renewable energy projects using software engineering principles. Section 5 presents the operational model for requirements engineering process. Theoretical implementation is described in Section 6. A practical implementation is illustrated in Section 7 through a case study. Conclusions and future research directions are discussed in Section 7.

2. An Overview of the Related Work

A significant amount of the work carried out to date on software engineering principles has been based on expert opinions, with a few exceptions where defined research methodologies have been used.

This section presents an overview of the related work done for software principles from an engineering perspective.

2.1 Bourque et al. 2002

The authors defined some fundamental principles in Bourque, et al., (2002), “Fundamental principles of software engineering – a journey”. A research methodology was used; it was composed of two workshops, two Delphi studies and a web based survey. As results, fifteen principles were identified as being candidate’s principles for software engineering.

2.2 Xingu Wang 2007

The author identified in “Software engineering foundations” in Wang (2007) a set of 55 principles of software engineering from the literature. Wang described a framework that contains a set of software engineering principles. The methodology used was based on mapping and on the exclusion of duplicates principles. The output was the identification of 31 principles for software engineering.

2.3 Normand Seguin et al.2010

Seguin, et al., (2010) identified in “software engineering principles: a survey and analysis” from the literature on software engineering principles, principles described in
the work of individual authors such as in (Bohem, 1983; Davis, 1995; Wiegers, 1996) - or in collaborative effort such as the work produced by (Bouque, et al., 2002; Bushmann, et al., 1996; Ghezzi, et al., 2003).

A methodology composed of two verification steps was used by Seguin, et al., (2010) using a set of criteria’s to check if these proposed principles were in fact a candidate fundamental principle (FP’s).

The output of the verification process resulted in the identification of 34 candidates (FP’s). Still these (FP’s) haven't been verified from an engineering perspective.

2.4 Meridji et al., 2010

Published as “Software Engineering Principles: Do They Meet Engineering Criteria?” Where a four phases methodology was designed to analyse and consolidate 34 candidates fundamental principles from an engineering perspective that was proposed by Meridji, (2010). Using both engineering criteria’s derived from the engineering categories of knowledge defined by Vincenti, (1990) and the engineering criteria’s defined by the joint IEEE and ACM software engineering curriculum IEEE & ACM (2004). The output of this research study resulted in the definition of 9 candidates engineering FP’s that were conforming to the engineering criteria’s. In addition, a hierarchy composed of specific instantiations of those candidates (FP’s) was proposed.

3. Research Objective

Software engineering is still an immature discipline when compared to other engineering disciplines as analyzed by (Abran, & Meridji 2006; Abran, et al., 2007). This actual research study is done towards making an operational process, for instance the requirements process of software engineering. This will eventually contribute to the better understanding of the principles of software engineering and to the development of renewable energy software’s as well as to the improvement of software engineering foundations. To achieve this objective a three phase’s methodology approach was developed.


The proposed development approach for renewable energy projects is composed of three phase’s methodology for the operational process as shown in Figure 1. This approach consists first in the identification of the requirements operating goals of the renewable energy project, second on the identification of the operating principles that may be applicable from the list of candidates engineering fundamental principles (FP’s) that were taken as input to this research work conducted by Meridji, & Abran, (2010), third, the description of the operational steps.

Figure 1. The Three Phase’s Development Approach for the Operational Process
4.1 Phase 1: Identification of Requirements Operational Goals

The first phase starts with the identification of the renewable energy requirements operational goals. This identification consists in identifying the requirements operational goals that are taking place during the requirements engineering process. To take place, this process needs four goals defined as follows:

- To elicit requirements;
- To analyse requirements;
- To specify requirements;
- To validate requirements.

4.2 Phase 2: Identification of the Operating Principles

This second phase identifies the set of operating principles that may be present and applicable for each of the requirements operational goals for renewable energy projects. These operating principles are taken from the list of nine candidates engineering fundamental principles done by Meridji, & Abran, (2010). Table 1 describes nine engineering FP’s that will be identified for renewable energy projects. The relation between the list of candidates engineering FP and the requirements operational goals is illustrated in Figure 2.

![Figure 2. Applicability Check between Operational Goals and the List of Software Engineering FP](image)

**Table 1. Nine Candidates Engineering FP Meridji, & Abran (2010)**

<table>
<thead>
<tr>
<th>ID</th>
<th>List of Software Engineering FP</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Apply and use quantitative measurements in decision making</td>
</tr>
<tr>
<td>2</td>
<td>Build with and for reuse</td>
</tr>
<tr>
<td>3</td>
<td>Grow systems incrementally</td>
</tr>
<tr>
<td>4</td>
<td>Implement a disciplined approach and improve it continuously</td>
</tr>
<tr>
<td>5</td>
<td>Invest in the understanding of the problem</td>
</tr>
<tr>
<td>6</td>
<td>Quality is the top priority; long term productivity is a natural consequence of high quality</td>
</tr>
<tr>
<td>7</td>
<td>Since change is inherent to software, plan for it and manage it</td>
</tr>
<tr>
<td>8</td>
<td>Since tradeoffs are inherent to software engineering, make them explicit and document it</td>
</tr>
<tr>
<td>9</td>
<td>To improve design, study previous solutions to similar problems</td>
</tr>
</tbody>
</table>

Phase 3: Description of the Operational Steps

Phase three describes the operational steps for the definition of the requirements of the renewable energy projects. It is concerned with the description of the different operational steps that are taking place into each of the defined operational goals as well as for each operating principle during the requirements process. Figure 3 describes the list of
operating principles that are taken as input for each of the operational goals for the purpose of operationalization.

![Figure 3. Applicability Check between Operational Goals and the List of Software Engineering FP](image)

5. Operational Model: Requirements Engineering Process

The operational model is composed of four operational goals such as: elicit requirements, analyze requirements, specify requirements and validate requirements. Into each of these operational goals a set of operating principles is applied after identifying their presence. Figure 4 illustrates the operational model. As well as the four corresponding operational goals and the set of operating principles that will be applied into each of the phases.

![Figure 4. Operational Model for Requirements Engineering Process](image)

6. Theoretical Implementation

This section presents the theoretical implementation of the development approach for renewable energy projects as described in Figure 1. A set of operating principles that are applied into each of the operational goals of the requirements process are implemented following the three phases methodology development approach.

6.1 Phase1: Identification of Requirements Operational Goals

The identification of requirements operational goals is the description of what is intended to be done during the requirements engineering process. To accomplish this process that allows the understanding of the requirements of any problem domain. A set of requirements operational goals is needed. In this phase four operational goals are identified and described as follows:

**Operational Goal 1:** elicit requirements

In this operational goal the system purpose is defined. In this process the collection of user’s and customer requirements is done using different elicitation techniques. A list of
requirements is produced at the end of this process. (SWEBOK Guide, 2004; Bruegge, 2010; Sommerville, 2011).

**Operational Goal 2: analyze requirements**

The work product from the previous activity is taken as input into the requirements analysis operational goal. The purpose of this activity is to find out problems; inconsistencies and incompleteness in the requirements. As well as it involves the development of one or more models of the system. These models are called analysis models (Pressman, 2009; Bruegge, 2010).

**Operational Goal 3: specify requirements**

To organize requirements the requirements engineer produces documents as a final artifact. This defines a set of requirements. This document serves as a contract between developers and customers as described by (Pressman, 2009; Bruegge, 2010).

**Operational Goal 4: validate requirements**

In this operational goal all related artifacts are evaluated for quality. These artifacts go through validation and verification test (Pressman, 2009; Bruegge, 2010; SWEBOK Guide, 2004).

### 6.2 Phase 2: Identification of the Operating Principles

This phase consists on the identification of the operating principles that are expected to be applicable for each operational goal of the requirement engineering process. Table 2 describes the list of the software engineering fundamental principles identified by Meridji, & Abran (2010), the four requirements operational goals and the result of the applicability check of the software engineering principles into each of the outlined operational goals.

Consequently, this applicability check outlines eight of the candidates engineering fundamental principles described in Table 1 to be applicable into the requirement operational goals except for principle #9 “To improve design, study previous solutions to similar problems”. Therefore, the candidates engineering fundamental principles that will be applied into the operational goals will be defined as operating principles as illustrated in Figure 2.

<table>
<thead>
<tr>
<th>ID</th>
<th>List of Software Engineering FP</th>
<th>Elct</th>
<th>Anal</th>
<th>Spec IF</th>
<th>Vali</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Apply and use quantitative measurements in decision making</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>2</td>
<td>Build with and for reuse</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Grow systems incrementally</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Implement a disciplined approach and improve it continuously</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>5</td>
<td>Invest in the understanding of the problem</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Quality is the top priority; long term productivity is a natural consequence of high quality</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>7</td>
<td>Since change is inherent to software, plan for it and manage it</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Since tradeoffs are inherent to software engineering, make them explicit and document it</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>To improve design, study previous solutions to similar problems</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 6.3 Phase 3: Description of the Operational Steps

The following section describes an illustration of the theoretical implementation of phase 3. It consists of the description of the operational steps into each of the
requirements operational goals for renewable energy software's for a selected set of operating principles, this includes:

**Principle # 5: “Invest in the understanding of the problem”**

The following requirements operational goals such as: elicit and analyze requirements apply principle #5.

The problem must be understood in the early stages of the software development lifecycle to better tackle the solution. The operational process for this principle includes the following operational goals followed by the operational steps:

**Operational Goal 1:** elicit requirements
- Sort out the requirements into the following sources: system high level goals, stakeholders, domain knowledge, and the organizational and the operational environment (Sommerville, & Sawyer 1997; Kotonya, Sommerville, 1998; SWEBOK Guide, 2004).
- Gather requirements using diverse elicitation techniques such as interviews with stakeholders, scenarios and use cases, prototypes, observations and open or closed group meeting (Sommerville, & Sawyer 1997; Kotonya, Sommerville, 1998; SWEBOK Guide, 2004; Pressman, 2009).
- Identify use cases and their relationships.

**Operational Goal 2:** analyze requirements
- Organize and classify the set of requirements to functional (a set of services that will be provided by the system), non functional (defines the attributes of the overall system), organizational requirements, environmental requirements, product requirements. (Sommerville, & Sawyer 1997; Kotonya, Sommerville, 1998; SWEBOK Guide, 2004; Pressman, 2009).
- Develop an analysis model that will include the static and dynamic models. The former consist of conceptual model and the functional model. (Bruegge, 2010; SWEBOK Guide, 2004; Pressman, 2009).
- Model the high level Architecture.
- Allocate requirements to components,
- Prioritize and negotiate requirements.

**Principle # 6: “Quality is the top priority; long term productivity is a natural consequence of high quality”**

The following software requirements operational goals: elicit, analyze, specify and validation requirements apply principle #6. There is a need to check quality of the final work products in the software requirements process for renewable energy system to assure and to improve the quality. The operational process for principle #6 includes the following operational goals as well as the operational steps:

**Operational goal 1:** elicit requirements
- Use prototyping to validate requirements.

**Operational goal 2:** analyze requirements
- Check the elicited requirements for inconsistencies, incompleteness and realism using a problem check list.
- Detects and resolves conflicts.
- Validate the quality of models by performing, the validation checks of the analysis models for instance; the conceptual model could be validated against the dynamic model as defined by Meridji & Ormandjeiva (2003).

**Operational goal 3:** specify requirements
Evaluate the quality of the documentations for instance for completeness and correctness to make it conforms to quality standards.

- Document standards
- Document quality

Operational goal 4: validate requirements
- Check the quality of the requirements process using quality standards
- Perform reviews,
- Verify the quality of the models developed during analysis,
- Define and design acceptance test that will be used to validate the conformance of the requirements expressed by the customer with the finished product.

Principle # 3: “Grow system incrementally”

The following software requirements operational goals: requirements elicitation, analysis, specification and validation apply principle #3.

The operational process for principle # 3: “Grow system incrementally” proposes that software should be build starting with few requirements (for instance requirements with high priority). Then, the software analyst increases the requirements slowly. The operational process for this principle includes the following operational goals followed by the operational steps:

Operational goal 1: elicit requirements
The implementation of this operational activity is similar to principle # 5: “Invest in the understanding of the problem”

Operational goal 2: analyze requirements
The implementation of this operational activity is similar to principle # 5: “Invest in the understanding of the problem”

Operational goal 3: specify requirements
In this operational activity many kinds of documents could be produced.
- Produce the software requirements specification document for part of the requirements for simple systems. SWEBOK Guide (2004).
- Produce documents such as: system requirements specification and software requirements specification for complex systems. SWEBOK Guide (2004).

Operational goal 4: validate requirements
The implementation of this operational activity is similar to principle # 6: “Quality is the top priority; long term productivity is a natural consequence of high quality”

Principle # 7: “Since change is inherent to software, plan for it and manage it”.

The following software requirements operational goals: to elicit, to analyze, to specify and to validate requirements apply principle # 7. Change is inherent to software development. Changes can occur in the operational activities of the requirements process for renewable energy software. This implies management of the changes throughout the whole software requirements process. The operational process for principle # 7 includes the following operational goals as well as the operational steps:

Operational goal 1: elicit requirements:
The set of the requirements as well as the sources of requirements are subject to several changes for instance: goals, stakeholders and many others may undergo change.

Operational goal 2: analyze requirements:
The classification and the types of requirements product or process may change as well as requirements priority, analysis modeling.

Operational goal 3: specify requirements
Documenting the requirements is an essential part in the requirements engineering process and in this activity changes are inevitable. Many parts of the documents are
subject to change such as the functional and non-functional requirements, the models or parts of the models.

**Operational goal 4: validate requirements**

In this operational goal the software requirements are checked for completeness, realism and correctness. Errors are certainly found during this process. Then these errors must go through a change process.

For the abovementioned operational goals here are the described steps that may be used in performing the changes defined by Sommerville (2011).

- Define the requirements that may change
- Analyze the changes
- Make the changes
- Verify the changes

**Principle # 4: “Implement a disciplined approach and improve it continuously”**

The following software requirements operational goals: requirements elicitation, analysis, specification and validation apply principle #4. A disciplined approach require following standards, a rigorous planning from cost and time estimation of the projects until the final delivery of the product. In addition, of continuously improving the approach and the processes, such as seeking a continuous process improvement using for instance the capability maturity model (CMMI). The operational process for this principle includes the following operational goals followed the operational steps:

**Operational goal 1: elicit requirements**

The implementation of this operational activity is similar to principle # 5: “Invest in the understanding of the problem”

**Operational goal 2: analyze requirements**

The implementation of this operational activity is similar to principle # 5: “Invest in the understanding of the problem”

**Operational goal 3: specify requirements**

The implementation of this operational activity is similar to principle # 3: “Grow system incrementally”

**Operational goal 4: validate requirements**

The implementation of this operational activity is similar to principle # 6: “Quality is the top priority; long term productivity is a natural consequence of high quality” .

**Principle # 8: “Since tradeoffs are inherent to software engineering, make them explicit and document them”**

Analyze requirements operational goal apply principle #8. Requirements engineers need to examine requirements for consistency, clarity and ambiguity. For instance, stakeholders may need to make tradeoff between functional and non-functional requirements or between requirements and resources as described by Sommerville (2011); this requires resolving conflict by negotiations as defined in the SWEBOK Guide (2004). The operational process for this principle contains the following operational goals as well as the operational steps:

**Operational goal 1: analyze requirements.**

- Identify the conflicting issues;
- Check with stakeholders to discuss a suitable negotiation.
- Trace the decision back to the customer;
- Put the decision into practice.
Principle #2: Build with and for reuse

The following software requirements operational goals: requirements elicitation, analysis and validation apply principle #2. The operational process for principle #2 suggests to reuse knowledge (artifact, pattern) and/or software (template, frameworks, code, components) and also to build software with the purpose of building components, templates ready to be reused for the future software development projects. It’s includes the following operational goals as well as the operational steps:

Operational goal 1: elicit requirements

Reuse can be used during the elicitation activity where a set of requirements could be reused to be analyzed.
- Reuse a set of requirements, scenarios or use cases, domain specific knowledge.

Operational goal 2: analyze requirements

In the requirements analysis many artifacts could be subject of reuse such as:
- Reuse the different models such as objects models, state models, event traces, user interactions and many others.
- Reuse of components in allocating requirements during architectural design.

Operational goal 3: validate requirements

During the validation activity reuse can be used to define a set of test cases. Figure 5 illustrates some elements that may be reused during the developments of the requirements process.

![Figure 5. Elements of Software Requirements Reuse](image)

7. Practical Implementation: Case Study

A Stand-Alone Photovoltaic Systems (PV) is a case study prepared in the PHOTOVOLTAIC DESIGN ASSISTANCE CENTER Sandia National Laboratories. And it is used for generating electric power by using solar cells to convert energy from the sun light into a flow of direct current electricity, which can be used to power equipment or to recharge a battery. Photovoltaic power systems are generally classified according to their functional and operational requirements, and how the equipment is connected to other power sources and electrical loads. The two principle classifications are grid-connected or utility-interactive systems and stand-alone systems. Photovoltaic systems can be designed to provide DC and/or AC power service, can operate interconnected with or independent of the utility grid, and can be connected with other energy sources and energy storage systems. Stand-alone PV systems are designed to operate independent of the electric utility grid, and are generally designed and sized to supply certain DC and/or AC electrical loads. These types of systems may be powered by a PV array only, or may use wind, an engine-generator or utility power as an auxiliary.
power source in what is called a PV-hybrid system. In many stand-alone PV systems, batteries are used for energy storage.

The practical implementation of this case study will follow the previously mentioned three phases approach. Starting by defining the requirements operational goals, the operational principles followed by the suitable operational steps.

**Operational goal 1: To elicit requirements**

For the purpose of this case study operating principle: “Invest in the understanding of the problem” will be taken as example to demonstrate the applicability of the principles. To elicit requirements there is a need to first identify sources of requirements and second collect requirements using elicitation techniques.

The high level requirement for stand-alone photovoltaic system is described as follow:

- Individuals and/or industrials recognize the need to use renewable energy resources.

**Step 1: Identify sources of requirements**

**System goal** (why the system is being developed)

A high level goal is to increase the capacity of renewable sources energy such as solar, wind and geothermal that will be utilised by individuals, governments and/or industrials.

**Domain Knowledge**

Some specific requirements such as: performance, reliability and security.

**Stakeholders**

Individual users, governments, industrials, operator, maintainer, manager.

**Operational Environment**

It will possibly be constrained by renewable energy equipments, software and hardware.

**Organisational Environment**

These are legal issues that bound the constructor’s of renewable energy projects from development, execution and commercialization of alternative energy resources to the land use and finance issues.

**Step 2: Elicit requirements using one of the elicitation techniques described in phase 3.**

A stand-alone PV system description of a general flow of work is defined as follow:

- A stand-alone PV system receives energy power from a PV array or may use a PV hybrid system as a supplementary second source such as wind, engine generator or utility power.

- The system provide direct current (DC) load by converting energy taken from a PV array into a flow of direct current electricity.

- The inverter converts energy power from batteries into alternating current (AC) electrical loads.
Description of how a system might be used

- A stand-alone PV system is used for generating electric power,
- The electric power can be utilized to power equipment or to recharge a battery.

Description of the necessary system services

- The system provide direct current (DC) load
- The system provide direct current (AC) load

Operational goal 2: To analyse requirements

To analyse requirements a classification of requirements into functional and non functional is necessary in addition to building the appropriate models.

Functional requirements of the system description

- The system shall be able to collect data such as battery rate.
- The system shall be developed to provide certain direct current (DC) and/or alternating current (AC) electrical loads.

Non-functional requirements of the system

- The system ensures the performance by performing the services for which it was developed as well as the ability to provide sufficient power to run the load.
- The system ensures reliability of providing a steady source of energy resources in addition to its ability to satisfy the predicted peaks.
- The system ensures energy security for future estimation purpose’s of energy resources that will be consumed for a short or a long period of time.

Conceptual model: Static model

The conceptual models consist of building static and dynamic models. First a class diagram is chosen for a static model.

![Figure 6. A Stand Alone PV System Example of a Static Model: Class Diagram](image)

Identify classes of objects and their associations for the Stand-alone photovoltaic system for instance there is: controller, PV array, Inverter, battery, DC load and AC load.
A stand alone PV system class diagram is described in Figure 6 as well as Figure 7 illustrates an example of a dynamic model depicted in activity diagram for a standalone PV system proving AC current load.

![Activity Diagram - A Stand Alone PV System Proving AC Current Load](image)

**Figure 7. Activity Diagram - A Stand Alone PV System Proving AC Current Load**

**Top level architecture for a mixed AC/DC system**

The different system components that are needed to operate a standalone system are described below. Figure 8 define the top level architecture for a standalone PV system and its components.

**Charge controllers:** In a stand-alone PV system the main function of a charge controller is to maintain the battery energy level from being too much charged as well as from being less charged as described in PHOTOVOLTAIC DESIGN ASSISTANCE CENTER Sandia National Laboratories (1989).

**Battery:** Batteries are responsible for storing electrical energy resources generated by the PV array in periods of sun, as well as whenever energy cannot be provided for instance in cloudy periods or at night the battery will be able to provide energy.

**Inverters:** Inverters convert DC loads into AC loads.

**Photovoltaic array:** PV array receives energy from the sun.

**Interface:** the defined interface that will display information’s such as battery rates.
Figure 8. Top Level Architecture for a Standalone PV System

Negotiate Requirements

The negotiation process allows resolving conflicts by providing feedback to the stakeholders.

Requirements for a standalone PV system are clear unambiguous and therefore can be realized.

This case study was successfully implemented using the operating principle “Invest in the understanding of the problem”.

8. Research Limitations

This research work had some limitations. This study was principally limited by a number of references on the fundamental principles of software engineering; the list of nine candidate fundamental principles is not necessarily exhaustive. A use of a more exhaustive set would have been more appropriate. Furthermore the content of the description of the operational steps is not comprehensive and is limited to the content of several software engineering references used during this study.

9. Conclusion

This research work on the application of a set of candidates engineering fundamental principles has used a three phase’s development approach to be able to operationalize the engineering fundamental principles into the requirements process for renewable energy projects.

This operationalization consisted first on the identification of the requirements operational goals, then the identification of the operating principles and the description of the operational steps for renewable energy projects.
A theoretical implementation of this development approach was successfully done. As a result of this theoretical implementation, eight operating principles were found to be applicable into the requirements operational goals. However, as it can be noticed that there is some overlapping between the theoretical implementation of the operating principles. For instance such that the one noticed between the operating principles “grow system incrementally”, “implement a disciplined approach and improve it continuously” and “invest in the understanding of the problem”.

Subsequently, a practical implementation was successfully implemented for the stand alone photovoltaic system. This implementation consisted of the application of “invest in the understanding of the problem” as operating principle. It has required the use of elicit and analyze operational goals.

The future work for this research study will concentrates on the experimentartion of each of the operating principles.

This research will help to build renewable energy projects using principles of software engineering. Furthermore it will eventually contribute to the improvement and the understanding of the principles of software engineering and probably to the teaching of these principles. This research work will ultimately help to advance the software engineering foundations.

References

[27] PHOTOVOLTAIC DESIGN ASSISTANCE CENTER.