

2008

University of North Carolina Wilmington
Master of Science in
Computer Science and Information Systems
Proceedings

<https://csbapp.uncw.edu/mscsis>

MS CSIS Capstone Project

An Applicable Approach to Signal Analysis and Peak Detection

Greg Huff

University of North Carolina Wilmington
And
GE Hitachi Nuclear Energy Americas
Wilmington, NC

August 25, 2008

Capstone Project Committee

Dr. Thomas Janicki (chair)
Mr. Jerald Head
Dr. Karl Ricanek
Dr. Gene Tagliarini



HITACHI



Table of contents

1. Introduction.....	1
2. Project Statement	5
3. Features and Benefits	8
4. Software Development Methodology	13
System Development Methodology.....	13
Alternative Software Development Methods.....	27
Technologies	28
5. Third Party Software Assessment.....	29
6. Summary of Work Completed.....	35
7. Timeline	49
8. Scope Modifications	50
9. Limitations	52
10. Lessons Learned.....	54
11. Course Work	56
12. Conclusion	57
13. Acknowledgements.....	59
14. References.....	61
Appendix A.....	63
Appendix B.....	64
Appendix C.....	65
Appendix D.....	66
Appendix E.....	67
Appendix F	68
Appendix G.....	69
Appendix H.....	70
Appendix I.....	71

List of tables

Table	Page
3.1 Feature and Benefits of the capstone project	9
4.1 Level 2 and UP comparison matrix	15
8.1 Scope Modifications	50

List of figures

Figure 1-1 Dicrotic Notch [5]	2
Figure 1-2 Nuclear Power Chugging	3
Figure 3-1 Application Framework	10
Figure 4-1 Unified Process (IBM)	14

Nomenclature

ABWR	Advance Boiling Water Reactor
ASCII	American Standard Code for Information Interchange
BWR	Boiling Water Reactor
CAR	Correct Action Request
CPR	Critical Power Ratio
DIVOM	(Delta CPR / ICPR) / (OM)
DSP	Digital Signal Processing
ECG	Electrocardiogram
ESBWR	Economic Simplified Boiling Water Reactor
FFT	Fast Fourier Transform

GEH	GE Hitachi Nuclear Energy Americas
GUI	Graphical User Interface
H/SSS	Hardware and Software System Specifications
ICPR	Initial Critical Power Ratio
MCPR	Minimum Critical Power Ratio
NRC	Nuclear Regulatory Commission
NRRL	Nuclear Reactor Reload Licensing
OM	Oscillation Magnitude
SDD	Software Design Document
SMP	Software Management Plan
SRD	Software Requirements Description
STP	Software Test Plan
STR	Software Test Report
UM	User Manual
UNCW	University of North Carolina Wilmington
UP	Unified Process
XML	Extensible Markup Language

1. Introduction

Analog signals are the result of many applications in the physical world. The complicating issues with the raw analog signals are the addition of noise and other physical phenomena to the underlying signal. Thus, engineering computations using analog signal inputs are subject to an increased degree of error. Identifying the underlying signal, without noise distortion, along with its corresponding properties, including minima and maxima, is a key to improving engineering methods.

Many engineering methods, such as those used in the Biomedical and Nuclear Energy industries rely on the precise detection of periodic peaks of a generated signal. Potential application domains may include medical, communications, defense, energy, and aerospace [1,2]. The source of the signal may complicate this process by adding noise or signal effects related to the physical properties of the source. Therefore, engineers are interested in the ability to analyze the signal including its raw form as well as the underlying frequency of the signal.

The medical industry relies on accurate signal processing in the interpretation of ECG results. These signals contain information, or components, which indicate the well being of an individual. The precise detection of each feature within these signals is crucial to the successful interpretation of the results [3]. The nuclear energy industry relies on the ability to accurately identify relevant peaks within a given signal to determine the stability of the reactor [4]. An interesting

phenomenon occurs in both industries (medical and nuclear) that may cause false results if interpreted incorrectly. The phenomenon is referred to as a “Dicrotic Notch” (Figure 1.1) in an ECG and referred to as “Chugging” (Figure 1.2) in a reactor power signal.

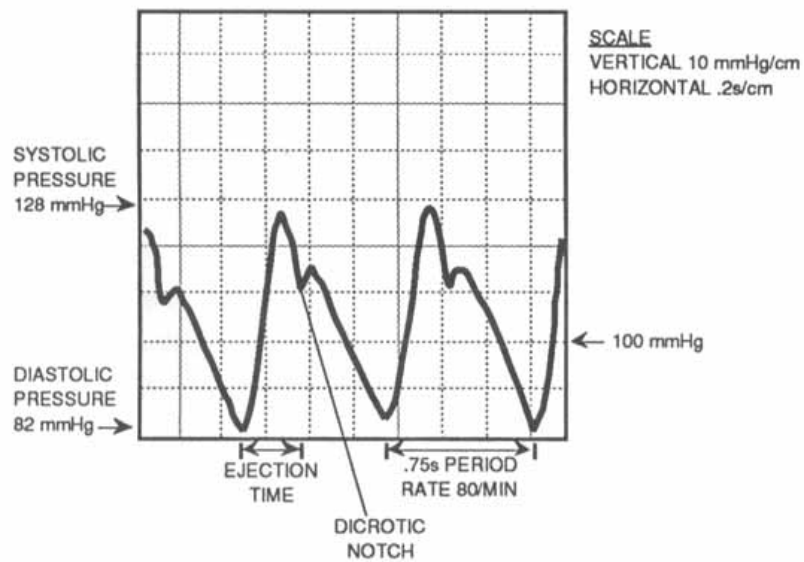
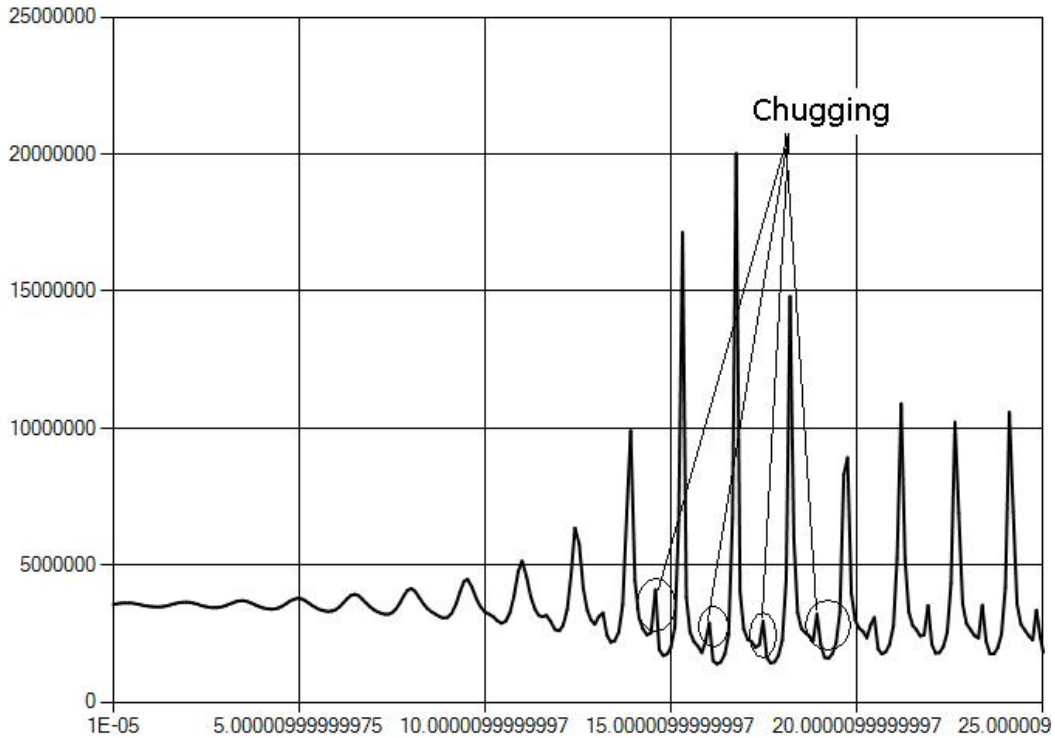


Figure 1.1 Dicrotic Notch [5]



**Figure 1.2 Nuclear Power Chugging
(Simulated Power Data)**

Biomedical research continues to improve the techniques used to identify the components of the ECG. Many techniques have been identified in literature including signal filtering and wavelet transformation to exploit the large slope and high frequency content of the components within the ECG [3].

An existing application named “ANALYZE”, used within the Nuclear Energy industry, has not used these modern classification or analysis methods. The purpose of this project is to replace the existing legacy application used to identify signal characteristics. The current application’s algorithm for peak detection compares the point under observation to its two neighbors. If the neighbor’s values are greater, there exists a minimum peak, or if they are less than the point

value under inspection, there exists a maximum peak. Therefore, any anomaly or signs of initial bifurcation within the signal, including chugging, will produce false peaks. Low pass filtering is applied within the current application, although the results of the filter are typically an attenuated signal with the additional peaks remaining present in the signal.

The objective of this project was to perform the analysis, design the software, seek management approval for the design proposed, and implement a software application that improves the accuracy and reliability of nuclear stability calculations. This application incorporates features from the current application as well as new and improved features. The new application name is “ANLYZ01P,” which is a GEH proprietary application.

The motivating benefit for this new application is to improve efficiency and quality of stability calculation results, which are used in NRRL stability studies conducted by GEH. The reload licensing process, approved by the NRC, requires the performance of specific analyses to develop reactor protection system set points necessary to ensure safe operation of a reactor with a specific core and fuel design. Currently, stability studies performed as part of the confirmatory and set point analyses are cumbersome and complex. This project provides one component among many to facilitate streamlining the stability study process.

2. Project Statement

Currently, the legacy application named, “ANALYZE,” is a proprietary application owned and used by GEH. This application is used by the stability team to execute calculations and assist in the identification of information used for NRRL stability studies. This application provides engineers many calculated results that would take many hours to compute manually. These results are reliable and verifiable as long as the input signals are free of abnormalities.

Several existing issues impacting the overall quality of the data within the application have been noticed; therefore it was decided to redevelop the application. The primary issue with the application is that it was considered an unverified source of data. This implies an engineer must verify the application results every time the application is executed and that verification must be performed by a method external to the application. This results in a very time consuming, inefficient, use of the stability engineer’s time. Another issue that was identified is the consistency with which the application provides complete results. The incomplete results indicate problems with the methods contained within the application.

After an extensive review of the current application’s code, the primary issue impacting results was identified as a faulty peak and valley detection algorithm. The existing algorithm uses a brute force method to determine where peaks and valleys occur within a signal. These peaks and valleys are then used as the basis

of each calculation in the application. Faulty peaks are identified due to the chugging or to changes in the signal frequency due to other systematic properties.

The merit of this project is based on the successful analysis, documentation, development, testing, and implementation of the software package. This project consists of the implementation of an application, “ANLYZ01P,” which will be used to interpret, plot, and report stability information and calculations related to nuclear power.

The completed application improves upon the issues identified with the legacy application and introduces new features, which assist engineers performing stability analysis. These features include:

- a. Preprocessing of signals to determine the quality and usability within the application
- b. A robust peak and valley detection algorithm
- c. A batch and GUI interface including graphical plots and ASCII results
- d. Application output including text, images, and binary reporting

In addition, this project was a catalyst in streamlining the process used by stability engineers, and will increase efficiency and quality. Other software projects, along with this project, will provide the ability to automate the process used to complete stability analysis.

The architecture of the application consists of an application framework, which provides the core functionality of data acquisition, peak detection, calculation engine, result plotting, and reporting. Calculations are considered “pluggable calculations” such that they may be developed, verified, and modified separately from the base application. Stability calculations included in this project are derived from known engineering methods used for NRRL stability studies conducted by GEH. Portions of those methods implemented in ANLYZ01P are considered proprietary to GEH.

3. Features and Benefits

ANLYZ01P provides an application framework, which contains the components responsible for fulfilling the functional requirements. The components of the application framework provide a logical distinction in the functionality and are subsequently described. The application is available to be executed via a GUI or via batch mode, which gives users the flexibility to automate the process by executing the application without user interaction [19].

The batch interface provides the stability engineers the ability to automate their work by executing the application with the appropriate inputs through a configured background process. This project will further improve efficiency by being a verified source of calculated information. A verified application implies a reduction in the time necessary for crosschecking since independent verification of results produced by the application is not necessary.

Quality improvements are derived from this project by the study and implementation of an enhanced application-specific peak and valley detection algorithm. Improvements to the preprocessing of signal data and identification of features within the signal increase the expected quality of application output. The application offers the ability to interpret, plot, and report results of calculations. Additionally, the application has the ability to “plug-in” calculation methods resulting in the computation and reporting of additional calculations, making the

application extensible. Table 3.1 summarizes the features and benefits of the new system.

Table 3.1 Features and Benefits of the capstone project

Feature	Description	Benefit
Data Acquisition	<ul style="list-style-type: none"> Data input for the application consists of different formats. Input data is comprised not only of time series data, but also of any application or user input data required to successfully execute the application. 	<ul style="list-style-type: none"> Reliable data analysis from standardized data formats eliminates manual intervention of entering data.
Peak Detection	<ul style="list-style-type: none"> Identifying the minimum and maximum peaks, including location and value, provides the data necessary for executing stability-specific calculations. The algorithm used to detect peaks is the subject of an analytical comparison within this project. 	<ul style="list-style-type: none"> Advanced algorithms determine necessary and accurate peak and valley values for stability calculations.
Calculation Engine	<ul style="list-style-type: none"> Provides the application interface required to execute engineering calculations that have been developed and verified. 	<ul style="list-style-type: none"> Application extensibility with robust management of computation resources.
Signal Plotting	<ul style="list-style-type: none"> Generating plot graphics to allow engineers to have the ability to evaluate, interpret, and report results. 	<ul style="list-style-type: none"> Concise analysis of data and calculations
Application Reporting	<ul style="list-style-type: none"> Information regarding the execution and output of the application and applicable calculations will be captured and saved. 	<ul style="list-style-type: none"> Easy verification and archival
Pluggable Calculations	<ul style="list-style-type: none"> The application engine offers the ability of using user-defined calculations. These calculations will provide extensibility to the application. Once developed and verified the calculations will “Plug” into the framework and be executable. 	<ul style="list-style-type: none"> Application extensibility and reduced development overhead.

User Friendly Interface	<ul style="list-style-type: none"> • Toolbar and menu driven dialog boxes for analysis. • Tabular interface for quick navigation. 	<ul style="list-style-type: none"> • Familiar look and feel reduces learning curve
Batched Execution Option	<ul style="list-style-type: none"> • Background submission and processing of reactor power data. 	<ul style="list-style-type: none"> • Automation capability reduces overall workflow time.

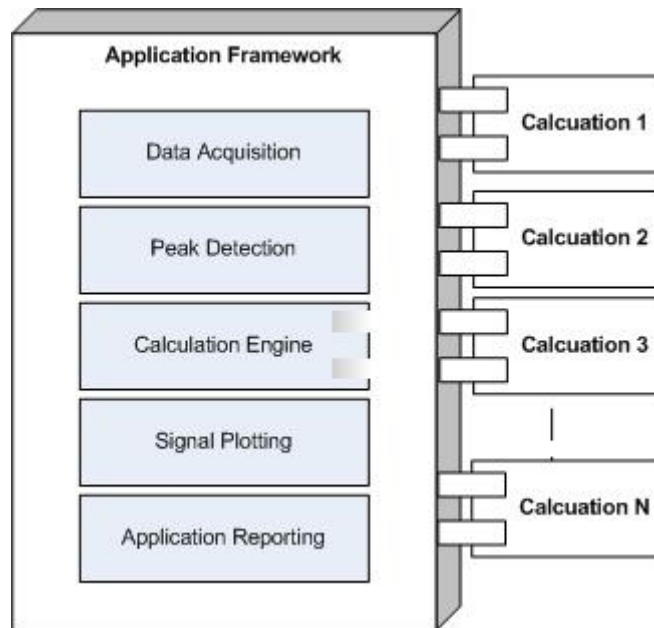


Figure 3.1 Application Framework

The application framework (Figure 3.1) is defined as the container responsible for the fulfilling the functional requirements of the project. An application framework is a modular and extensible object-oriented concept. The application framework contains five main components and is extensible by the implementation of calculations. These components provide a logical distinction in the functionality provided by the application.

- **Data Acquisition:** Compiles data input for the application, which has varying formats. Input data is comprised not only of time series data, but also any application or user input data required to successfully execute the application. Time series data is generated by GEH proprietary simulation software.
- **Peak Detection:** Identifies the minimum and maximum peaks, including location and value, which provides the data necessary for executing domain-specific calculations. The algorithm used to detect peaks is the subject of an analytical comparison within this project.
- **Calculation Engine:** Provides the application interface required to execute engineering calculations that have been developed and verified.
- **Signal Plotting:** Generates plot graphics to allow engineers to have the ability to evaluate, interpret, and report results.
- **Application Reporting:** Captures and saves information regarding the execution and output of the application and applicable.

As shown in Figure 3.1, the Calculation Engine offers the capability to execute user-defined calculations. These calculations provide extensibility to the application. Since an object-oriented concept is used, once new calculations are developed and verified the calculations will “plug” into the framework and be executable.

Finally, the implications of this work lead to a more robust method of executing calculations depend on the identification of extrema. This application provides a visual interpretation of signal data and robust calculations of stability methods to engineers.

As a result of the work completed by this project, the user have indicated that procedures will be written and/or modified to capture the algorithmic changes made to the existing methodology. Therefore any ambiguity in interpretation of current methodology shall be minimal, if not eliminated, due to the comprehensive detail with which the application is developed. The process owner (end user) of the algorithms and procedures has expressed written appreciation for the level of effort and detail involved in generating comprehensive algorithms (Appendix H).

4. Software Development Methodology

System Development Methodology

The official methodology for developing this software is known as the “Level 2” process. Level 2 is a proprietary process used and owned by GEH. The Level 2 process outlines all of the necessary analysis, documentation, reviews, and approvals required for successful implementation of the project. The Level 2 deliverables are indicated in the Work Completed section (section 6). In practice the Level 2 process most resembles the UP Software Development methodology. This software development process is considered fluid and amenable to change throughout the development cycle using prototypes to define and refine requirements. An example of the UP methodology is embedded in the IBM Rational System Design Software Package [6].

Level 2 projects require a responsible engineer, design team, and review team to analyze, document, review, implement, and test the project. The design team is led by a design chair, who is typically the responsible engineer, and who is responsible for the development of requirements and design of the software. The review team is led by a review chair, which cannot be the responsible engineer, is responsible for the review and approval of all requirements and software design. The design and review teams consist of experts in the field of the application.

As described previously, the Level 2 process can be directly related to a recognized software development process. The Unified Process is based on four

distinct phases as illustrated by Figure 4.1. Each of these phases is described and related to how this project will be developed [6]. The Level 2 development process is comprised of phases similar to the Unified Process. Table 4.1 shows the major activities for each phase in relation to the Unified Process. Iterations of the phases result in modifications to deliverables.

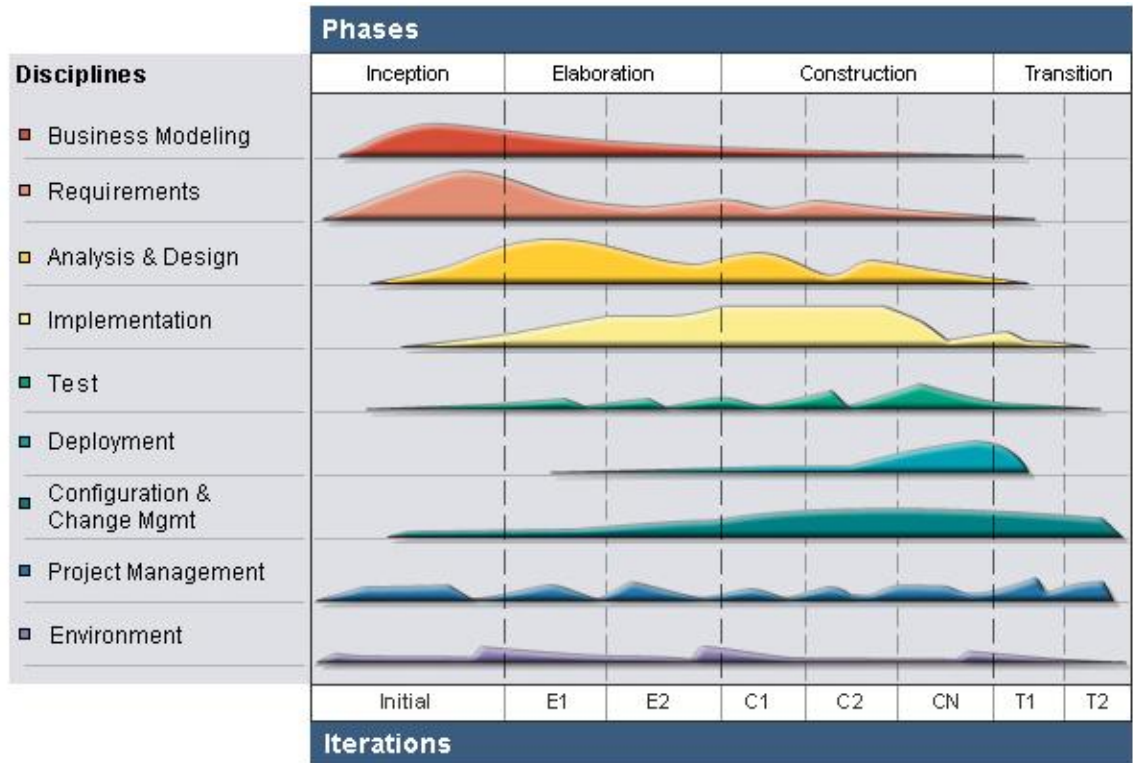


Figure 4.1 Unified Process (IBM) [21]

Table 4.1 Level 2 and UP comparison matrix

GEH Level 2 Process	Unified Process			
	Inception	Elaboration	Construction	Transition
CAR	<ul style="list-style-type: none"> • Business Case • Project Approval 			
Level 1	<ul style="list-style-type: none"> • H/SSS • SMP • Review and Approval 			
Level 2 Phase 1		<ul style="list-style-type: none"> • SRD • STP • Review and Approval 	<ul style="list-style-type: none"> • Development Iterations 	
Level 2 Phase 2			<ul style="list-style-type: none"> • SDD • STR 	<ul style="list-style-type: none"> • UM • Review and Approval

Inception Phase

- **CAR:** Details the domain for the project. Management then approves and assigns resources to the project based on priority.
- **Level 1:** The Level 1 process is best described as an iterative process working closely with the users of the system to complete the initial specifications of the application. Once this Level 1 approval is complete, it is not revisited in subsequent iterations. Level 1 is comprised of the following:
 - *Legacy application understanding and documentation*
 - Understanding of the existing process and application is captured and used to capture specifications.
 - *H/SSS*
 - The H/SSS defines all the functions that the system is to perform, the hardware and software elements of the system,

and the performance and testing requirements associated with these functions and elements. The definitions specifically include the objectives and broad functions to be performed, any significant physical modeling assumptions, the inputs and outputs, the specific range of application, and any special interface testing requirements or other constraints. The testing plan should define the specific data or other bases to be used and the associated acceptance criteria. Significant information in the H/SSS includes:

- System Objective: a general vision of the core project's requirements, key features, and main benefits.
- System Definition:
 - Functional Description: a description of the functionality the application must have.
 - Interface Requirements: a description of how the application will work with other applications or processes already in place.
 - System Diagram: a flow chart of process relationships
 - Initial use-case model (optional): an initial use case describing the overall application. Includes all external entities with which the system will interact

(actors) and defines the nature of this interaction at a high level.

- Existing Resources and Procedures: a description of the current state for the application.
- Hardware and Software Elements: a description of necessary hardware and software tools needed to execute the application.
- Characteristics:
 - Performance Characteristics: a description of expectations for application performance.
 - Maintainability: a description of how the application will be maintainable.
 - Availability: a description of how the application will be made available and any restrictions on who shall have access to the application.
- Modeling and Testing Requirements:
 - Significant Modeling Assumptions: a description of assumptions for software modeling based on known engineering methods, if any.

- Special Interface Testing Requirements: a description of how the testing shall be completed with respect to upstream or downstream applications or processes.
- Quality Assurance:
 - Specific Verification Requirements: a description of how the application shall be verified.
 - Maintenance: a description of any configuration management, (i.e. source code, executable, documentation management).
- *SMP*
 - This work plan defines the documentation that will be generated during the development of this application, the extent of the reviews during the development, and the schedule for the appropriate documentation, testing and Level 2 Design Review. Resources and responsibilities for each of the tasks shall be defined.
- *Level 1 Design Review*
 - The Level 1 Design Review should thoroughly evaluate the development plans, including appropriateness of the planned functions, input and output, documentation, range of application, qualification testing (scope and acceptance

criteria), plans for maintaining application compatibility (upstream and downstream), as well as any NRC approval and Licensing Impact issues. Thorough Level 1 Design Reviews are very important to assure useful, high-quality applications. Inappropriate software development should be redirected at this early stage.

- *Level 1 Approval*
 - Level 1 approval signifies completion of the Level 1 or Inception phase, which includes the initial documentation of core requirements, key features, and main constraints. All documentation to this point is electronically signed and archived. The project may be cancelled or considerably modified if it fails to pass this milestone.

Conceptually, during this phase, the business case for the system is established. This project is considered necessary for process improvement, therefore extensive business case documentation is considered unnecessary for purposes of the development of this application.

The Level 1, or Inception, phase was completed in the fall of 2007. See the Work Completed section (section 6).

Elaboration Phase

- **Level 2 Phase 1:** The purposes of this phase are to analyze the problem domain, establish a sound architectural foundation, develop the project plan, and eliminate the highest risk elements of the project. Architectural decisions are made with an understanding of the whole system: its scope, major functionality, and nonfunctional requirements such as performance requirements.

It is easy to argue that the elaboration phase is the most critical of the four phases. At the end of this phase, the hard “engineering” is considered complete and the project undergoes its most important day of reckoning: the decision on whether or not to commit to the construction and transition phases. For most projects, this also corresponds to the transition from a mobile, light and nimble, low-risk operation to a high-cost, high-risk operation with substantial inertia. While the process must always accommodate changes, the elaboration phase activities ensure that the architecture, requirements, and plans are stable enough, and that the risks are sufficiently mitigated, such that the cost and schedule for the completion of the development can be determined. Conceptually, this level of fidelity would correspond to the level necessary for an organization to commit to a fixed-price construction phase.

The documentation and work to be completed included during Level 2 Phase 1 are as follows:

- *SRD*

- The SRD includes the technical basis description that defines the technical assumptions and mathematical formulation of the application. It also describes special validation requirements, inputs and outputs, and other constraints on the software embodied in the application.

Significant information found in the SRD includes:

- Technical Assumptions: a description of assumptions used in the development of the requirements for this application.
- Mathematical Formulations: a description of the mathematical constructs contained within the application.
- Analytical Documentation: a description of derived algorithms and the basis from which they are formulated.
 - Specific peak analysis definition
- Discussion of Inputs: a description of values, files, and data needed to execute the application.
- Discussions of Outputs: a description of the expected results, logging, and GUI of the application.

- Software Constraints and/or Limitations: a description of the bounds in which the application is expected to operate.
- Validation Requirements: a description of how the application shall be tested and verified.

Other information optionally found in the SRD include:

- User Requirements: requirements for the system, which are directly testable and reportable.
- Use Cases: a description of how the users will execute the application. The use cases shall be specific to functionality contained within the system and representative of workflow using the software.
 - Intermediate Use Cases: detailed information for actions within the use case.
 - *STP*

The STP describes the overall test. The STP provides the detailed description of the software tests including unit, integration, and system tests required to assure that the coding and associated software data library satisfy all requirements as defined in the H/SSS and SRD.

The Level 2 Phase 1 process is defined by both the Elaboration and Construction phases of the Unified Process. Therefore, for purposes of

illustration, the discussion of Level 2 Phase 1 deliverables is noted as Construction Phase.

Construction Phase

- **Level 2 Phase 1 (cont'd):** During the construction portion of Level 2 Phase 1 all remaining components and application features are developed and integrated into the product. Remaining deliverables of Level 2 Phase 1 which directly relate to the Construction Phase include:
 - *SDD*
 - The SDD summarizes how the software is designed to satisfy the H/SSS and SRD requirements. This includes definitions of the software modules, hierarchical charts, data flows, and logic flows at a level appropriate for effective development and maintenance. Documentation of Standards applied, Metrics evaluated, and internal data may be included. Significant information found in the SDD includes:
 - Object Models: descriptions of objects, or classes, which make up the overall application
 - Screen Designs: descriptions of the GUI via mock screen images.

- XML Definitions [7]: a description of the definitions used to constrain inputs for the application.
 - Classes and Methods Definition: a description of necessary classes and the methods or members defined in a given class.
- *Iterative Software Development*
 - The application is developed iteratively, resulting in prototypes, and reviewed by the design team until all requirements have been addressed as defined in the H/SSS and SRD. Depending on the result of the Level 2 Phase 1 Review, the application code may or may not be considered complete. Application Coding must be complete prior to completion of testing and verification in Level 2 Phase 2.
 - *Initial User Manual (UM)*
 - Documentation of user instructions describing the terminology and features of the application.
 - *Level 2 Phase 1 Design Review*
 - The Level 2 Phase 1 Design Review should thoroughly evaluate the SRD, SDD, and STP. The Review team is responsible for crosschecking the completion of requirements and design of the application during this review. Any items found during the review must be

addressed and re-reviewed prior to Level 2 Phase 1 Approval.

- *Level 2 Phase 1 Approval*
 - Review of Level 2 Phase 1 deliverables including SRD and STP by the Review Team. This approval is the final milestone of Level 2 Phase 1.

Level 2 Phase 1, or Elaboration and Construction Phases, was an iterative process that included design team meetings and input as well as application development progress. As a result of the iterative process, the process for establishing requirements became extensive. This phase was more time-consuming and demanding than anticipated. The project design team invested extensive time in the development of calculation algorithms. Each iteration during the algorithm design was supported by the development of a prototype. Each prototype was reviewed, feedback was provided; and algorithm changes were made as results were tested against existing analyses. An important lesson learned was the number of iterations and prototypes required and the length of time required for each iteration. Due to the number of iterations, the original project schedule was not feasible. This phase has been completed with the approval of both the SRD for algorithms and application functionality and the STP.

Transition Phase

- **Level 2 Phase 2:** The goal of this phase is to transition the software product to the user community. Once the product has been given to the end user, issues usually arise that require the development of a new release.

Level 2 Phase 2 officially begins with the approval of Level 2 Phase 1, although work related to this phase may commence prior to the official start. This typically requires that some usable subset of the system has been completed to an acceptable level of quality and that user documentation is available so that the transition to the user will provide positive results for all parties. Deliverables of Level 2 Phase 2 include:

- *STR*
 - The STR documents the results of the unit, integration, and system test identified in the STP.
- *User Manual (UM)*
 - Finalization of the documentation of user instruction describing the terminology and features of the application. Descriptions of user and system inputs and outputs, error messages, and an example user scenario for the application shall be included.
- *Training Material (optional)*

- Documentation of lecture material used to describe the application and how to use it.
- *Level 2 Phase 2 Design Review*
 - The Level 2 Phase 2 Design Review should thoroughly evaluate the results of the STR verifying all results are acceptable prior to the Level 2 Phase 2 approval. The Review team will also review supporting documentation.
- *Level 2 Phase 2 Approval*
 - Review of Level 2 deliverables including extensive scrutiny of the STR, comprehensive UM documentation, and SDD.

Upon completion of the Level 2 Phase 2 approval, the application is released to the users and considered complete. Any additional modifications or error corrections are contained within independent projects with respective software development processes. Metrics should be evaluated once user data and feedback can be collected.

Alternative Software Development Methods

Other development process models were considered while drawing parallels to the Level 2 Process used in this project. One was of the Waterfall model, which does not provide the flexibility required for this project so that all the users may see iterations of the project during the implementation process. The Level 2 process

does have at least one similarity to the Waterfall development methodology. They are similar with respect to the approval process of each phase. Once the application has been approved at that phase it is not likely to iterate through that phase until a new revision or error correction occurs. Overall the Waterfall model was considered too rigid for this project because it does not allow for the iterations defined by the Level 2 process [8].

The Agile model was also considered to describe the development process used. Agile software development was considered too flexible, since the development is contained within a specific process, the Level 2 process. Therefore the Unified Process was selected since it most closely matches the GEH methodology for application development.

Technologies

The project development platform is based on the Microsoft .Net 2.0 development platform. Several platforms were considered, including .Net, Java, and Fortran. .Net, and specifically the programming language C Sharp, was selected based on its adaptability into all industries which may have an interest in this application. It provides the GUI-driven interface needed to accomplish the requirements of the project. Java was excluded due to the restrictions in the Sun Microsystems, Inc. Binary Code License Agreement for using Java in Energy specific applications [9]. Fortran was excluded due limitation in graphical presentation and an inability to deliver the graphical requirements of the project.

5. Third Party Software Assessment

A third party product review of software packages providing functionality similar to this application has been completed and is subsequently summarized. Four vendors and five software packages were considered based on relevance of the software they provide to this project. The results of this review conclude that there does not exist a commercially available software package that can economically provide the functionality needed to complete the requirements of this software project.

- SIGVIEW by Signal Lab -- <http://www.sigview.com/index.htm>

Vendor Provided Description

SIGVIEW is a real-time signal analysis software package with wide range of powerful FFT spectral analysis tools, statistics functions and a comprehensive visualization system. SIGVIEW is distributed as shareware – it can be downloaded as a completely functional version and use it for 21 days to find out if it suits your needs. If you decide to use SIGVIEW after that period, you will have to purchase a license. The license key will unlock the time-limited trial version and entitle you to e-mail support, free minor upgrades and possible customization options.

Analysis

SIGVIEW does not have a peak detection algorithm, although it offers customizable applications. Therefore this software would not provide the Peak Detection or the Calculation Engine needed to accomplish the requirements of this project. SIGVIEW is available with a per seat license up to 20 seats, and a site license.

- AutoSignal by SYSTAT -- <http://www.systat.com/products/AutoSignal/>

Vendor Provided Description

AutoSignal is the first and only program that completely automates the process of analyzing signals. Save precious time by eliminating the programming time normally required for performing sophisticated signal analysis. AutoSignal takes full advantage of its graphical user intuitive interface to simplify every aspect of operation, from data import to output of results. Choose your analysis techniques from the menu or toolbar. Select the algorithm and options from the interface. You get immediate visual feedback with 2D or 3D graphs of your signal analysis plus numeric summaries for reports.

AutoSignal gives researchers the power to rapidly find components of complex signals that normally require extensive programming and mathematical routines. AutoSignal provides a vast array of spectral

analysis procedures to help you make intelligent conclusions for any application. Built-in spectral analysis procedures include:

- FFT
- Autoregressive
- Moving Average
- Autoregressive Moving Average
- Complex exponential modeling
- Minimum variance methods
- Eigen analysis frequency estimation and Wavelets

Analysis

This software provides a wide array of analytical tools for signals. The features lacking in this software are the ability to identify peaks and valleys and the extensible calculations for further numerical analysis of the data. AutoSignal is available on a per user license basis.

- PeakFit by SYSTAT-- <http://www.systat.com/products/PeakFit/>

Vendor Provided Description

Peakfit's state - of -the - art nonlinear curve fitting is essential for accurate peak analysis and conclusive findings. PeakFit separates and analyzes nonlinear peak data better, more accurately and more conveniently than

your lab instrument. Nonlinear curve fitting is by far the most accurate way to reduce noise and quantify peaks.

PeakFit helps separate overlapping peaks by statistically fitting numerous peak functions to one data set, which can help you find even the most obscure patterns in your data. The background can be fit as a separate polynomial, exponential, logarithmic, hyperbolic, or power model. This fitted baseline is then subtracted before peak characterization data (such as areas) is calculated, which gives much more accurate results.

Analysis

PeakFit provides the Data Preparation and Peak Detection mechanism, but does not combine that with a Calculation Engine for processing calculations using the peaks detected in the application. PeakFit is available on a per user license basis.

- Matlab and Simulink by Mathworks -- <http://www.mathworks.com/>

Vendor Provided Description

The MATLAB product family provides a high-level programming language, an interactive technical computing environment, and functions for:

- Algorithm development

- Data analysis and visualization
- Numeric computation

MATLAB serves as the foundation for all other MathWorks products (including signal processing).

Signal Processing Toolbox is a collection of industry-standard algorithms for analog and DSP. Signal Processing Toolbox also provides graphical user interfaces for interactive design and analysis and command-line functions for advanced algorithm development.

Analysis

Matlab is a robust, comprehensive software package with extensive Signal Processing toolboxes. Matlab, together with Simulink, provides a platform for algorithm and software development. Matlab will be used during the research and development of algorithms within this project, but project implementation using Matlab is not feasible due to complexity and cost. Matlab is available on a site or per user license basis.

- LabView by National Instruments -- <http://www.ni.com/labview/>

Vendor Provided Description

LabView has vast expertise in the Signal Analysis domain. The software is award winning and recognized in many industries as a standard of excellence in the Signal Analysis field. They offer many tools in addition to the base application, which allow the detection of peaks and valleys using different mechanisms. This application also allows the development of custom integrated software.

Analysis

LabView is a comprehensive application with many features unnecessary for this application. The LabView Software model is very modular and the modules needed to meet the requirements of this project would include: Signal Process, Analysis, Math, and Signal Express. To obtain the necessary components for this application, would require a significant investment on a per user license in addition to training. Therefore LabView is not an acceptable option for this project.

Third party product investigation results in an understanding that products exist in the market with a subset or superset of the functionality needed to meet the requirements of this project. Other than the monetary implications of a Third Party product, the most prevalent drawback is the lack of a User Defined calculation engine. Many products are shipped with a standard set of Analysis tools but lack the extensibility proposed in this project.

6. Summary of Work Completed

Work Completed in Fall 2007

CAR (Inception)

- Business Need identified – The Stability team has identified, documented, and received approval to complete this project in efforts to improve the quality and efficiency of the work completed during each Reload Licensing study.
- Project Team identified – The Responsible Engineer, Design Team and Chair, and Review Team and Chair have been assigned and approved by the business. The project team consists of 9 people.

Phase: Level 1 (Inception)

- *Legacy application understanding and documentation* – An extensive review of the business process, and existing application and methodology, was completed. Example found in Appendix C.
- *H/SSS* – Hardware and software specifications were captured and approved by both design and review teams. Example H/SSS section information in Appendix A. Information in this documentation includes:
 - System Objective
 - Business Benefits
 - System Capabilities
 - System Definition

- Functional Description
 - Interface Requirements
 - System Diagram
 - Existing Resources and Procedures
 - Hardware and Software Elements
- Characteristics
 - Performance Characteristics
 - Maintainability
 - Availability
- Modeling and Testing Requirements
 - Significant Modeling Assumptions
 - Special Interface Testing Requirements
- Quality Assurance
 - Specific Verification Requirements
 - Maintenance
- *SMP* – The SMP has been documented and approved by both Design and Review team. Example SMP information in Appendix B.

Information in this documented includes:

- Tasks and Quality Requirements
- Deliverables
 - Required Documentation
 - Installation Platforms
 - Schedule

- Authorization
 - Configuration Management
- *Level 1 Design Review* – All requirements and documentation have been reviewed and meeting documentation completed.
- *Level 1 Approval* – All Level 1 deliverables have been approved.

Phase: Level 2 Phase 1 (Elaboration and Construction)

- *Initial Design Meeting Completed* – Design Team meeting to discuss Engineering Methodology and Calculations within the application.
- *SRD* (Appendix D)
 - Technical Assumptions:
 - Mathematical Formulations:
 - Data Preprocessing
 - Signal Variance measurement across entire signal
 - Minimum data range restrictions
 - Limiting Channel OM restrictions
 - Engineering Methods Documented – Calculations required for Stability have been identified. Calculations include:
 - DIVOM – The relationship between the fractional change in CPR and power OM.
 - Channel Hydraulic Decay Ratios
 - Single Oscillation Decay Ratios

- Peak Detection: a description of derived algorithms and the basis for which they are formed.
 - Initial Peak Detection Methods Analysis – Historically false peak detection has been the source of inaccurate calculations. Peak discussions have occurred and prototypes and proofs of concepts implemented [10,11,12,13].
 - DSP techniques have been used to generate discussion for accurate peak detection algorithms. Techniques include:
 - Frequency domain analysis via FFT
 - Signal Filtering techniques
 - Zero Padding, Interpolation, and Smoothing algorithms considered [14,15,16,17,18]
 - Filtered Signal Inverse Peak identification
 - Discussions of Inputs
 - Discussions of Outputs
- *SDD* (Appendix E)
 - Screen Design – Initial screen designed.
- *Iterative Software Development* – Sandbox environment developed for method development (Appendix F)

Work Completed in Spring 2008

Phase: Level 2 Phase 1 (Elaboration and Construction)

- *SRD* – The Requirements description was divided into two separate documents such that the approval process could be divided into manageable tasks. See Appendix D for SRD documentation excerpts.
 - **Algorithms** – This document is dedicated to the description of the algorithms used for calculations implemented for this project. Information in this document includes:
 - Flow diagrams for each calculation – Each algorithm is detailed with a high level flow diagram to first explain the algorithm prior to providing the user with details.
 - Description of each task within the flow diagram – Algorithmic details are provided per task illustrated in the flow diagram. Details include description of the task, equation, and appropriate criteria.
 - Example calculation when appropriate – A full example is provided for algorithms that are confusing in nature. The example is a non-trivial example and considered valuable for communicating the principles of the algorithm.
 - Algorithms include: (Appendix D provides a selection of detail from the SRD Algorithms document)
 - Peak Detection

- DIVOM
 - Channel DIVOM
 - Standard DIVOM
 - Composite DIVOM
- Channel Hydraulic Decay Ratio
- Single Oscillation Decay Ratio Calculation
- **Application Functionality** – This document provides the detailed description of what the application is to perform. It contains a summary of inputs, outputs, and application functionality. All calculation information is referenced within the Algorithms document. Information contained in this document includes:
 - Input description – Each potential input into the application is documented along with a description, optional/required, and default value. Criteria are specified for the input values where appropriate.
 - Description of execution methods
 - Batch execution – Requirements are specified for the application to be run via command line and without a GUI.
 - GUI execution – Requirements are specified for the GUI and provide a description of the functionality. Functionality includes toolbars, menus, context

menus, shortcut keys, data presentation, plot presentation, plot data, data logging, user notifications, and input value presentation.

- Data requirements – Description of data used within the application including decimal precision, input units, output units, conversion calculations, and applicable calculations.
- Application Output
 - Application Data – data specific to identifying the application on output generated by the program.

Data includes:

- Application name
- Version of software build
- Time of execution
- Times series data
- User name
- Machine name
- User Input Data – information used for purposes of input into the application, including data provided directly by the user and default data.
- Binary File Input Data – information provided as input to the application originating from the input Cedar file including raw data and graphical plot representation.

- Channel Calculated Data – information calculated from Cedar file input data. This information includes raw data and plots for Normalized, Filtered, Interpolated, and FFT data.
- Analysis Group Calculated Data – information representing the results of calculations performed within the application. Calculations include:
 - DIVOM
 - Channel Hydraulic Decay Ratio
 - Single Perturbation Decay Ratio
- ASCII Output – description of the information expected in the ASCII output. The application contains a flag, which indicates whether or not to include verbose data. This section identifies the inclusion or exclusion of data based on the presence of the verbose flag.
- Image Output – requirements information with respect to saving images. In GUI mode the user has the ability to select which images to save, if any.
- Binary File Output – specified requirements for saving result data in a form accessible to downstream applications.

- Trace Logging – the application shall produce a log identifying pivotal points during the execution of the application, verbose errors, information usable for troubleshooting, and appropriate application statistics.
- Software Constraints and Limitations – Description of information defining the restrictions and applied limitations on data, functionality, and performance.
- Validation Requirements – A basic description of tests needed to validate the functionality listed within the application
- *SDD* – This document describes the design decisions behind the functionality implemented in the project. Sections of this document are found in Appendix E and include:
 - Logical Application Flow
 - Data Flow
 - Software Modules
 - Data Acquisition
 - Peak Detection
 - Calculation Engine
 - Calculations
 - Signal Plotting
 - Application Reporting

- GUI
- Error Handling
- Execution Methods
- *STP* – This document describes the tests needed to test and validate the software. Example tests are documented in Appendix G. The test plan includes the following tests:
 - Unit / Integration tests – The following items will be established through the Unit / Integration testing:
 - Ensure that the code is free of programming errors
 - Verify the code’s ability to execute the desired functions
 - Discover structure / data flow problems under a variety of conditions prior to user testing
 - Validation / System tests – The Validation / System testing will be performed to establish the following:
 - Ensure proper source data is used
 - Ensure generation of requested outputs as required
 - Ensure calculations are correct. Reference the SRD for areas of calculation concern.
- *Software Development*
 - All modules and calculations including:
 - Data Acquisition
 - Peak Detection
 - Calculation Engine

- Plotting
 - Application Reporting
 - Calculations
- Preliminary Software Testing – A total of 9 prototypes have been developed. Each prototype has been subject to numerous tests. The test include:
 - Peak detection accuracy
 - Calculation algorithm accuracy (See Appendix I for example)
 - GUI requirements
 - Application output requirements
- *Level 2 Phase 1 Design Review* – A complete formal review has been conducted including the presentation of:
 - SRD
 - Application Functionality
 - Algorithms
 - STP
- *Level 2 Phase 1 Approval* – As a result of the Level 2 Phase 1 Design Review, all Level 2 requirements have been approved as documented.

Remaining Work

Phase: Level 2 Phase 2 (Transition)

- *STR*

- The software test report is scheduled to be complete by the end of the first week of June 2008. Although informal testing has been completed to date, the software test report will consist of finalized test results for the software application
- *Finalization of SDD*
 - The SDD is scheduled to be complete by the end of the third week of May 2008. The final requirement changes must be documented before the SDD is complete.
- *UM*
 - The user manual document is scheduled to be complete by the first week of June 2008. The user document will be collaboration between the project team members. This deliverable is outside of the scope of this project.
- *Level 2 Phase 2 Design Review*
 - The Level 2 Design Phase 2 Review shall take place once all testing, design, and user manual documentation has been completed. This review is scheduled to take place in the second or third week of June 2008. This deliverable is outside of the scope of this project.
- *Level 2 Phase 2 Approval*
 - Upon the completion of the review, and any modification resulting from the review, the application shall be approved for Level 2 use.

This is scheduled to occur by the end of June 2008. This deliverable is outside of the scope of this project.

Statement of Completion

In the original project proposal it stated that the project should be considered complete when:

- Software is developed to meet the specification of all requirements documented and approved by the Level 2 Phase 1 Approval process
 - *The software has been approved through the Level 2 Phase 1 Approval process.*
 - *Software has been developed according to documented Level 2 Phase 1 approved requirements.*
- Developed Software is successfully tested and verified
 - Preliminary software testing has been completed with respect to the algorithms used to calculate results and the output generated by the application.
- All Project documentation is complete with respect to developed software
 - *Documentation with respect to the Level 2 Phase 1 Approval process has been completed.*
 - *Design Documentation shall be completed in May 2008 with respect to the addition of new functionality added to the application at the end of the Level 2 Phase 1 Review process.*

This project shall not be responsible for any requirements or documentation requested after the Level 2 Phase 1 Approval.

7. Timeline

Tasks to be performed for this project are:

Project Task	Planned	Actual
CAR		
Business need	Jan 2006	Jan 2006
Project Approval	Feb 2006	Feb 2006
Level 1		
Problem Research	Sept 2007 Wk 3-4	Sept 2007 Wk 3-4
H/SSS	Sept 2007 Wk 4	Sept 2007 Wk 4
SMP	Oct 2007 Wk 1	Oct 2007 Wk 1
Graduate Project Proposal	Dec 2007 Wk 2	Dec 2007 Wk 2
Level 2 Phase 1		
SRD	Oct – Nov 2007	Nov 2007 – Apr 2008
SDD	Dec 2007 – Jan 2008	Feb 2008 – May 2008
STP	Jan 2007	Apr 2008
Software Development	Dec 2007 – Feb 2008	Dec 2007 – Apr 2008
Data Acquisition	Dec 2007	Dec 2007 – Jan 2008
Peak Detection	Dec 2007 – Jan 2008	Jan 2008 – Apr 2008
Calculation Engine	Jan 2008	Jan 2008
Result Plotting	Dec 2007 – Jan 2008	Dec 2007 – Apr 2008
Application Reporting	Jan 2008	Jan 2008 – Apr 2008
Calculations	Jan 2008 – Feb 2008	Dec 2007 – Apr 2008
Level 2 Phase 1 Design Review	Jan 2008	Feb 2008 – Apr 2008
Level 2 Phase 1 Approval	Jan 2008	Apr 2008
Level 2 Phase 2		
STR	Feb 2008 – Mar 2008	TBC
UM	Feb 2008 – Mar 2008	TBC
Level 2 Phase 2 Design Review	Feb 2008 – Mar 2008	TBC
Level 2 Phase 2 Approval	Mar 2008	TBC
Capstone Project Closure		
Final Project Documentation	Apr 2008 Week 1	Apr 2008 Week 4
Final Project Defense	Apr 2008 Week 2	May 2008 Week 1
Project Modifications (as needed)	Apr 2008 Week 3&4	May 2008 Week 1
Project Completion	May 2008	May 2008

8. Scope Modifications

Modifications have been made to the scope of work due to the iterations of prototypes and refinement of requirements. The following modifications have been made:

Table 8.1 Scope Modifications

Disposition	Feature	Description / Benefit	Reason
Discarded	Ad-hoc Plotting	<ul style="list-style-type: none"> User selected data set plotting (GUI only) 	<ul style="list-style-type: none"> Tools exist for Ad-hoc plotting.
Discarded	Application BWR EDB Output	<ul style="list-style-type: none"> Output verified data to a verified repository for downstream use 	<ul style="list-style-type: none"> Output format unknown at this time.
Modification	Normalization Calculation	<ul style="list-style-type: none"> Normalize data with respect to either the first value in the dataset, or a specified value, for use in calculations or plotting 	<ul style="list-style-type: none"> Modified the calculation from stand-alone calculation to embedded calculation
Addition	Single Oscillation Decay Ratio Calculation	<ul style="list-style-type: none"> Calculate Decay Ratio for reactors that do not oscillate 	<ul style="list-style-type: none"> NRC approved stability calculations for new reactors
Addition	Application Trace Log Output	<ul style="list-style-type: none"> Generate ASCII data representing progress and errors during execution for debug process 	<ul style="list-style-type: none"> Easily trace errors that may occur in application
Addition	Plotting Requirements	<ul style="list-style-type: none"> Plots shall have a footer containing specific application information for verification purposes 	<ul style="list-style-type: none"> Application output shall be identifiable with respect to the source that generated the output
Addition	Plotting Requirements	<ul style="list-style-type: none"> Plots shall have standardized colors across all plots for channels represented on the plots 	<ul style="list-style-type: none"> Reduce human performance issues in interpreting plots
Addition	Calculations	<ul style="list-style-type: none"> Reactors shall have 	<ul style="list-style-type: none"> Provides the

		<p>“Analysis Groups” which will define the data grouping for calculations. The application shall report on the comparison of the Analysis Groups</p>	<p>flexibility to execute calculations based on sub-groups of channels within the reactor</p>
--	--	--	---

9. Limitations

This project has the limitation of domain-specific datasets. Data used within the application shall be derived from a relatively noiseless simulation of real world nuclear reactors. This domain provides the bounds in which the application is to be developed. Therefore, the data contained within sample datasets consist of minimal noise, but significant phenomena within the data, which shall be taken into account during peak detection. There exist a significant number of datasets, including problematic and steady state data, to complete comprehensive analysis of the peak detection algorithm. Other limitations have been documented in the SRD document as shown in Appendix D and include:

- Input Data – ANLYZ01P executes calculations based on Time domain data. Data is limited to a single dimension and indexed by time. The user input file shall be consistent with defined user options in this SRD. Deviation from the definition may result in an incomplete or unexecuted portion of the code.
- Reactor Characteristics – ANLYZ01P expects simulated channel signals that are oscillating to execute the peak detection and calculations. Stable signals or effectively Steady State signals are not supported by ANLYZ01P.

- Channel Data – ANLYZ01P expects reactor channel data to execute calculations. Channel data shall exist and be oscillating in order to provide results.
- Reactor Types - ANLYZ01P supports BWR and ABWR for DIVOM calculations. ANLYZ01P supports ESBWR Decay Ratio calculations. Any simulated reactor channel is supported for Channel Hydraulic Decay Ratio calculations.
- Batch Submission Limitations – ANLYZ01P shall be available to execute on the Microsoft Windows server requested for Batch execution.
- Performance Limitations – ANLYZ01P calculates many internal data points used to derive the results of requested calculations. Therefore input data with a large time range is subject to reduced performance in calculation time and memory. When possible the time range in the input data shall be kept to a reasonable range. A reasonable range shall be defined as less than or equal to 1500 time steps.

10. Lessons Learned

As a result of this project there have been many lessons learned. The first lesson is that existing application algorithms are not always optimal. The first prototype delivered to the users implemented the DIVOM calculation based on existing, or legacy, code. As a result, I learned the existing code was not in exact alignment with the written procedures in place at the time because the existing code had not evolved with associated process changes.

The second lesson learned was a result of the second iteration prototype, which was based on existing procedures. The discrepancies arose when individuals interpreted the procedure differently. As a result I learned that procedures can be ambiguous and therefore not a reliable source for requirements.

The third lesson learned was a result of correcting the issues from the second lesson. Therefore the third prototype was delivered based on joint discussions of interpretations of the existing procedure and individual experiences from the engineers. As a result I learned that experience varies and individuals would handle similar cases differently. This was especially true with cases, which exhibited extreme behavior in some way. Therefore, the final algorithm was developed based on current procedures, procedure modifications, experience, and known exceptions and challenging cases from the end users. This algorithm took many iterations, brainstorming, and examples to determine details that could be implemented in code. Perhaps one of the most challenging details of the final

algorithm was the definition of the fuzzy logic used. The fuzzy logic built into the algorithm is designed to optimize the result provided by the application and must be quantified to be used.

The fourth lesson learned is to fully understand the tools that are available for use. A setback in the development of this application occurred because the corporate standard for the plotting tool was used in early prototypes. The assumption was made to follow corporate standards and not investigate feasibility of using a standard third party plotting tool. As a result of this false assumption, the standardized plotting tool had to be removed and replaced with open source software. The result of this effort included unexpected time to obtain corporate approval to use the open source software, code refactoring time, and time to implement functionality in the open source software that was needed to satisfy project requirements.

Finally, as a result of the process to establish requirements and refactor the code for plotting, I learned project planning is an art. As evidenced by this project, the schedule is only as good as the definition of the requirements. Therefore, early assumptions made with respect to schedule were simply assumptions without a valid basis. The lesson learned is to adequately account for uncertainty while planning a project, especially for projects with incomplete or unknown requirements.

11. Course Work

This project is a culmination of classroom and project experiences gained at UNCW and GEH. Many courses have contributed to my foundation of knowledge enabling the completion of this project. The following courses have played a significant role during the project:

- Analysis, Modeling, and Design – This class provided helpful tools to use during the processing of requirements discovery and documentation. The Unified Process methodology was the foundation for this class and used as a basis in this project.
- Software Engineering – (Undergraduate and Graduate) This class provided the fundamentals used every day in my professional career. Fundamentals include software methodologies, risk management, analysis, design, and change control.
- Digital Image / Signal Processing – This class provided the fundamental knowledge of Signal Analysis. Techniques learned in this class have been applied to this project.
- Project Management – This class was helpful with the management of the overall project.

12. Conclusion

Currently this project is in the testing stage and will be implemented at GEH upon the approval of the Level 2 process. Many lessons have been learned as a result of this project for the entire project team. As a result of the diligent work on this project, new procedures will be written, old procedures will be updated, and a uniform process for calculating the results has been adopted. Although the project schedule was not met, the stability group will be able to realize the benefits provided by this application in time for an increased workload due to many upcoming Reload Licensing Analyses in the fall of 2008. Some of these benefits include:

- The reduction in time it takes to document verified results in a usable format
- The time that will be saved because the output will not have to be verified
- The time that will be saved due to the application's ability to run all channels at once, compared to each channel being run separately
- The confidence that can be placed in the verified results
- The flexibility to select either a GUI or batch execution method

Future Work

Future revisions of this software may include functionality that was unable to be specified and delivered as a part of this project. The functionality includes:

- Application Output to a verified data repository – At the time of this revision, the format needed for this output could not be defined. Future work and additional projects will define the output format.

- Additional calculations for future stability analysis – GEH continues to develop the Stability Analysis organization. As this development occurs, future calculations may result which can be integrated into this application.
- Process automation scripting – A key to delivering additional benefit with this application will be the integration of automation scripting. This will allow the application to execute with inputs and outputs defined in a series of other applications to streamline the process engineers use for compiling data for the reload license reports.
- Advanced interactive plots – Plot enhancements include providing an interactive plotting environment, which would allow engineers to set parameters dynamically on the plot and execute the calculations.

13. Acknowledgements

There are many people to acknowledge for their contributions to the success of this project. First, and foremost, I would like to acknowledge the commitment and support of my wife, Christy Huff. Her encouragement, understanding, and patient love enabled me to complete this project.

There have been several key members of the project, who contributed to the design and requirements definition. The design team has provided unfettered access for questions, feedback, criticisms, testing, and review. Members of the GEH design team include Leah Crider, Juswald Vedovi, Necdet Kurul, Steve Shelton, and Christan McElroy. I am grateful for the support each of my team members.

Similarly, I am grateful for the members on the review team. The review team members have made themselves available when needed for long review meetings and approval. The review team members include: David Vreeland, Shawn Lamb, and Alan Chung.

Thank you to Kristen Triplett, who joined the group in February and has been a faithful meeting attendee, note taker, and document reviewer. I recognize making sense of the conversations and ideas was not an easy task.

Tom Janicki, PhD, has been a great advisor. He has provided guidance, feedback, and encouragement throughout the project. Without his support and encouragement, there is no telling when I would graduate. My wife thanks you as well.

Jerry Head has provided leadership and oversight to the project from GEH's perspective. I thank you for your leadership, support, and access to your team.

Gene Tagliarini, PhD, provided technical knowledge and guidance for the project. I am thankful for the fundamentals you have instilled, which allow me to succeed.

Karl Ricanek, PhD, provided technical insight and guidance for the project. Thank you for your contributions to the success of this project.

I appreciate all of those who made it possible to complete this project as a joint effort between GEH and UNCW. I feel this optimized the learning experience and allowed me to work and make valuable contributions. Thank you.

Last but not least, I dedicate this project to my wife, Christy, daughter, Sarah (3yrs old), and son, Will (3 months). I look forward to closing this chapter in our life and starting a brand new one. I love each of you more than I can express.

14. References

1. “The MathWorks – MATLAB and Simulink for Technical Computing.” 1994-2007. The MathWorks, Inc. viewed November 2007. <
<http://www.mathworks.com/industries/>>
2. Saeed V. Vaseghi, *Advanced Signal Processing and Digital Noise Reduction*, Wiley-Teubner, 1996
3. Aboy, M., J. Mcnames, Tran Thong, D. Tsunami, M. S. Ellenby, and B. Goldstein. "An automatic beat detection algorithm for pressure signals." *Biomedical Engineering, IEEE Transactions on* 52 (2005): 1662-1670.
4. Brennen, Christopher Earls (2005) *Fundamentals of Multiphase Flow*. Cambridge University Press. ISBN 13 978-0-521-84804-6
5. “Data Acquisition Peak Detection Analysis Software.” Figure 1. 2007. DATAQ Instruments, Inc. viewed November 2007
<http://www.dataq.com/applicat/articles/an8.htm>
6. Rational Unified Process, Best Practices for Software Development Teams, Rational Software White Paper TP026B, Rev 11/01,
http://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf
7. “XML Schema Tutorial.” 1997-2007. W3Schools. Viewed November 2007. <
<http://www.w3schools.com/schema/default.asp>>
8. Peter Schuh, *Integrating Agile Development in the Real World* (Programming Series), Charles River Media, Inc., Rockland, MA, 2004
9. “Java Technology Licensing.” Sun Microsystems, Inc. viewed September 2007. <
<http://www.java.com/en/download/license.jsp>>
10. “Peak Detector PtByPt (Not in Base Package) – LabView 8.5 Help.” August 2007. National Instruments. viewed November 2007.
<
http://zone.ni.com/reference/en-XX/help/371361D-01/ptbypt/peak_detector_ptbypt/>
11. “Peak Detector (Not in Base Package) – LabView 8.5 Help.” August 2007. National Instruments. viewed November 2007. <
http://zone.ni.com/reference/en-XX/help/371361D-01/lvanls/peak_detector/>

12. "Peak Detection Using LabView and Measurement Studio- Developer Zone – National Instruments." February 1, 2006. National Instruments. viewed November 2007. < <http://zone.ni.com/devzone/cda/tut/p/id/3770>>
13. "Wavelet-Based Peak Detection- Developer Zone – National Instruments." October 9, 2006. National Instruments. viewed November 2007. < <http://zone.ni.com/devzone/cda/tut/p/id/5432>>
14. "Zero Padding Does Not Buy Spectral Resolution- Developer Zone – National Instruments." February 1, 2006. National Instruments. viewed November 2007. <<http://zone.ni.com/devzone/cda/tut/p/id/4880>>
15. "FFT Properties 3.5 The basic tutorial on frequency analysis." Dew Research. viewed November 2007. < <http://www.steema.com/FFTProp/FFTProperties/FFTProperties.htm>>
16. Smith, J.O. "FFT of a Zero-Padded Sinusoid", in Mathematics of the Discrete Fourier Transform (DFT) with Audio Applications, Second Edition, http://ccrma.stanford.edu/~jos/mdft/FFT_Zero_Padded_Sinusoid.html, 2007, online book, accessed November 2007.
17. "FFT Fundamentals – LabView 8.2 Help." August 2006. National Instruments. viewed November 2007. < http://zone.ni.com/reference/en-XX/help/371361B-01/lvanlscconcepts/fft_fundamentals/>
18. "FFT Interactive Tutorial- Developer Zone – National Instruments." February 1, 2006. National Instruments. viewed November 2007. < <http://zone.ni.com/devzone/cda/tut/p/id/4186>>
19. Batch Applications—The Hidden Asset, Mainframe Migration Alliance, June 2006, <http://download.microsoft.com/download/4/1/d/41d2745f-031c-40d7-86ea-4cb3e9a84070/Batch%20The%20Hidden%20Asset.pdf>
20. Ikeda, Hideaki, et al. "BWR Stability Issues in Japan." Science and Technology of Nuclear Installations. Volume 2008. Article 358616. Hindawi Publishing Corporation. October 2007. 11 pages
21. Jackson, Robert B., Stephen D. Burd, and John W. Satzinger. Object-Oriented Analysis and Design : With the Unified Process. Boston, MA: Course Technology, 2004.

Appendix A

Hardware and Software Specifications (H/SSS) Examples

(This appendix has been removed due to potentially confidential / proprietary content.)

Appendix B

Software Management Plan (SMP) Examples

Project Schedule is an initial schedule developed for the SMP. This schedule is then reviewed by the review and design team members and agreed upon. Often the schedule is modified, as requirements are better understood.

Setup: Set Permissions	1 day	24-Sep-2007	24-Sep-2007
Setup: Migration of Source Code	1 day	24-Sep-2007	24-Sep-2007
Setup: Set Check-in Policies	1 day	24-Sep-2007	24-Sep-2007
Setup: Send mail to users for installation and getting started	1 day	24-Sep-2007	24-Sep-2007
Setup: Create Project Structure	1 day	24-Sep-2007	24-Sep-2007
Create Project Plan	1 day	24-Sep-2007	24-Sep-2007
Lvl 1: Design Doc Complete Hardware/Software System Specifications	1 day	25-Sep-2007	25-Sep-2007
Lvl 1: Design Doc Complete Software Management Plan	1 day	26-Sep-2007	26-Sep-2007
Lvl 1: Design Review Approval	2 days	27-Sep-2007	28-Sep-2007
Lvl 1: Design Review	2 days	1-Oct-2007	2-Oct-2007
Lvl 1: Release eDR	3 days	3-Oct-2007	5-Oct-2007
Lvl 1: Complete Check List	2 days	8-Oct-2007	9-Oct-2007
Lvl 1: Request Level 1 from the Control Component	1 day	10-Oct-2007	10-Oct-2007
Lvl 2 Phase 1: Complete Software Requirements Description	25 days	11-Oct-2007	14-Nov-2007
Lvl 2 Phase 1: Complete Software Test Plan	10 days	15-Nov-2007	28-Nov-2007
Lvl 2: Design Review Approval	2 days	29-Nov-2007	30-Nov-2007
Lvl 2 Phase 1: Design Review	2 days	3-Dec-2007	4-Dec-2007
Lvl 2 Phase 1: Develop Application	30 days	5-Dec-2007	15-Jan-2008
Lvl 2 Phase 1: Static Code Check	2 days	16-Jan-2008	17-Jan-2008
Lvl 2 Phase 1: Standards Check List	2 days	18-Jan-2008	21-Jan-2008
Lvl 2 Phase 1: Load to Program Library	2 days	22-Jan-2008	23-Jan-2008
Lvl 2 Phase 2: Complete Software Test Report	3 days	24-Jan-2008	28-Jan-2008
Lvl 2 Phase 2: Complete User Manual	5 days	29-Jan-2008	4-Feb-2008
Lvl 2 Phase 2: Design Review	4 days	5-Feb-2008	8-Feb-2008
Lvl 2 Phase 2: Release Level 2 eDR	3 days	11-Feb-2008	13-Feb-2008
Lvl 2 Phase 2: File Test Suite	4 days	14-Feb-2008	19-Feb-2008
Lvl 2: Complete Check List	3 days	20-Feb-2008	22-Feb-2008
Lvl 2: Request Level 2	3 days	25-Feb-2008	27-Feb-2008

Appendix C

Legacy application understanding and documentation

(This appendix has been removed due to potentially confidential / proprietary content.)

Appendix D

Software Requirements Document (SRD) Examples

(This appendix has been removed due to potentially confidential / proprietary content.)

Appendix E

System Design Examples

(This appendix has been removed due to potentially confidential / proprietary content.)

Appendix F

Prototype Development Examples

(This appendix has been removed due to potentially confidential / proprietary content.)

Appendix G

Software Test Plan Examples

(This appendix has been removed due to potentially confidential / proprietary content.)

Appendix H

Letter of Gratification

From: Vreeland, David G (GE Infra, Energy)

Sent: Thursday, March 06, 2008 3:28 PM

To: Bowman, Scott (GE Infra, Energy); Head, Jerald G (GE Infra, Energy)

Cc: Kurul, Necdet (GE Infra, Energy, US); Vedovi, Juswald (GE Infra, Energy); Shelton, Steven B. (GE Infra, Energy); Crider, Leah (GE Infra, Energy); Huff, Greg (GE Infra, Energy)

Subject: ANALYZE Level 2 Design Review
Scott and Jerry,

Just wanted to let you know how impressed I was with the design team for the ANALYZE code (Lead by Greg with Juswald, Steve, Necdet and Leah). They have taken a complex non-level 2 code that has caused us numerous headaches over the past +5 years and produces something really remarkable. It not only corrects known errors but is generic its application to handle many unforeseen situations.

They have removed what has been a persistent thorn in the Stability group.

As the process lead, I am very grateful.

David

Appendix I

Test Calculation Results Example

(This appendix has been removed due to potentially confidential / proprietary content.)