Contract No:

This document was prepared in conjunction with work accomplished under Contract No. DE-AC09-09SR22505 with the U.S. Department of Energy (DOE) National Nuclear Security Administration (NA).

Disclaimer:

This work was prepared under an agreement with and funded by the U.S. Government. Neither the U. S. Government or its employees, nor any of its contractors, subcontractors or their employees, makes any express or implied:

1) warranty or assumes any legal liability for the accuracy, completeness, or for the use or results of such use of any information, product, or process disclosed; or
2) representation that such use or results of such use would not infringe privately owned rights; or
3) endorsement or recommendation of any specifically identified commercial product, process, or service.

Any views and opinions of authors expressed in this work do not necessarily state or reflect those of the United States Government, or its contractors, or subcontractors.

Minimizing Error Potential Using Computer Applications – 18498

S. David Burke, Susan W. Lawson, and Jason A. Hargrave
Savannah River Remediation, LLC
Savannah River Site
Aiken, SC 29808

ABSTRACT

Many Savannah River Site (SRS) liquid waste facility operations depend upon repetitive engineering calculations and operator actions. Although these calculations and actions are formalized in technical reports and paper procedures, they can still be vulnerable to data entry errors, calculation errors, equipment alignment errors, and occasional changes caused by different people performing these activities. Savannah River Remediation (SRR) is using computer automation to reduce these error precursors by standardizing repetitive calculations and by performing plant alignment checks for repetitive operation processes. Data entry can be automated, calculation methodologies can be standardized, and computers can check to see that predefined plant conditions exist before equipment operation. In addition to improving quality, the time required for these activities to generate technical reports, to perform independent reviews, and to perform equipment alignment checks can be significantly reduced.

In the case of operator-performed calculations, error precursors involve distractions that can occur over a multi-page paper procedure, confusing displays and controls from equipment, mathematical errors, complacency, and unfamiliarity with the procedure among many others. Repetitive engineering calculations suffer from similar error precursors and are often compounded by high workloads, lack of proficiency, lack of standard methodology, mental shortcuts, and biases. Web-based computer applications can eliminate many of these precursors through standardization. Calculation methods are fixed, inputs are fixed, and computer checks can highlight potentially wrong inputs and outputs. Automated calculations are performed almost instantaneously once input is provided. Operators and engineers can spend more time focusing on the results than on the process to achieve the results. Similarly, when operators use repetitive procedures involving equipment manipulation, the same error precursors apply. Automated operation sequences and equipment checks can be programmed into the distributed control system to provide standard operating processes that reduce and eliminate error precursors.

This paper describes how SRR incorporates Human Performance Improvement (HPI) principles when automating calculations and operation processes. Computer applications will be reviewed in detail, highlighting the software development life-cycle that includes requirements gathering, prototyping, design, development, testing, and implementation.

INTRODUCTION

Operations and engineering activities are subject to many Human Performance Improvement (HPI) error precursors. Operators may be distracted during equipment line-ups. They may become complacent when following procedures because of their past successes with a task. Procedures may be complex and confusing. Engineering activities involving technical calculations present the possibilities of knowledge-based errors such as biases, limited capacities, lack of knowledge, lack of competence, and lack of resources [1]. This paper describes how Savannah River Remediation (SRR) uses computer automation to eliminate many of these types of error precursors and to mitigate error risks.

Ideally, this paper will spark readers' imaginations by highlighting how HPI tools have been used in some SRR applications. The subsequent challenge is for readers to learn from these concepts and use their imagination about how similar designs might be used in their own software applications. There is no "one-size-fits-all" for how HPI tools can be designed. The reader is more likely to develop variations of the themes presented.

DISCUSSION

Table I is an abbreviated table of HPI tools found in the Department of Energy HPI Handbook, Vol. 2. [1] The table depicts the applicability for a given tool for operations activities that usually involve equipment manipulations and for engineering activities that usually involve paperwork such as calculations and designs. The computer automation column highlights the HPI tools that have been developed in the SRR distributed control system (DCS) and in software applications for repetitive operations activities and for repetitive engineering calculations.

TABLE I: HPI Tool Correlations

HPI Tools	Operations Applicability	Engineering Applicability	Computer Automation
Task Preview	●		●
Job-Site Review	●		
Questioning Attitude	●	●	●
Stop When Unsure	●	●	●
Self-checking	●	●	●
Procedure Use &Adherence	●	●	●
Validate Assumptions		●	
Signature		●	●
Three-way Communications	●		
Phonetic Alphabet	●		
Place-keeping	●	●	●
Pre-job Briefing	●	●	
Peer-checking	●		●
Concurrent Verification	●		
Independent Verification	●		●
Peer Review		●	●
Flagging	●		●
Turnover	●	●	●
Post-job Review	●	●	●
Project Planning	●	●	
Problem Solving (PACTS)		●	
Decision Making		●	●
Project Review Meeting		●	
Vendor Oversight	●	●	

Although HPI principles are traditionally taught with a focus on some type of human implementation, in many cases, an automated tool may be a more reliable option. Computer logic is unbiased. Computers are programmed for explicit actions. They can be programmed to allow users to proceed to the next step only after having satisfied, or attested, that prescribed HPI tools have been performed. Defined processes in a computer application are consistent. Any deviations from the defined norm are quickly identified by flags that identify when data is out of range, or by a failure of the program to execute. Using automation can force more deliberate decision-making when abnormal or unexpected conditions arise and can benefit the user by focusing attention where it is needed instead of on routine, redundant tasks.

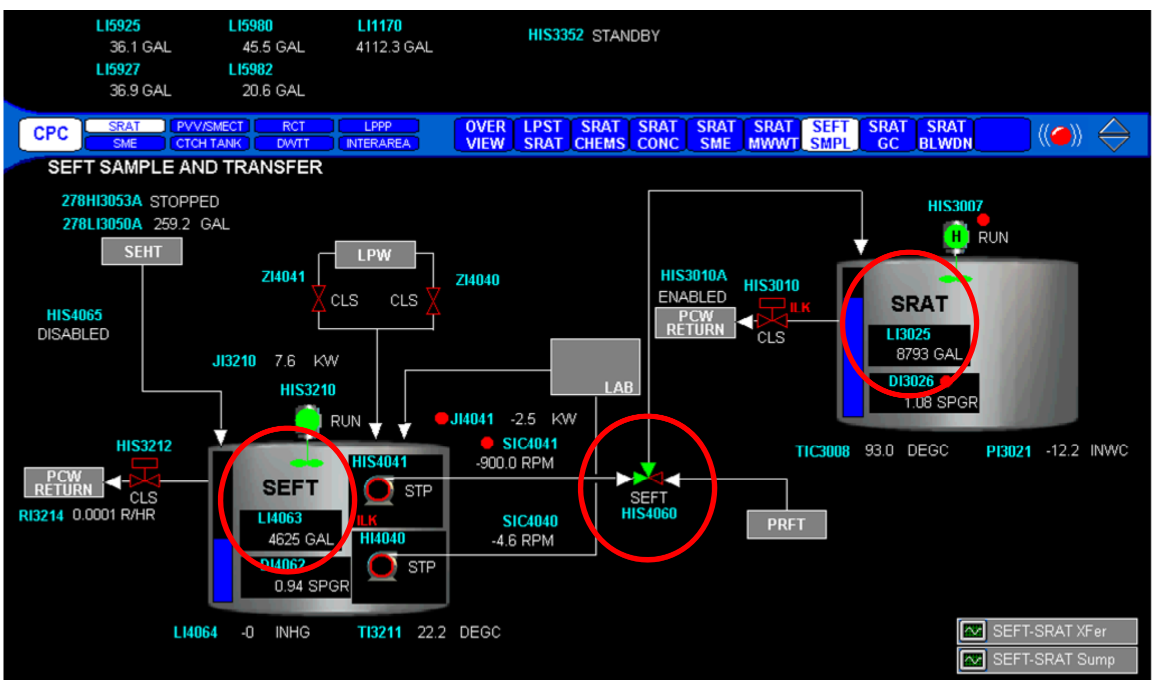
The challenge for a software developer is to develop software that includes the HPI tools listed above. Excluding business software, SRR software falls into two main categories: operations software related to the facility distributed control systems (DCS) and engineering calculation software. The following sections describe how SRR has integrated HPI tools with these software types.

HPI and Facility DCS Operations

DCS software benefiting most from HPI tools is associated with equipment manipulations within a facility. Valve alignments and pump operations may be controlled by operators in a central control room via a DCS computer screen. Most equipment commands are governed by use-every-time procedures that move material from point A to point B, that start a tank agitator, start a fan, or start some other process. The specific computer commands are innocuous if performed correctly.

Consider the following typical liquid transfer sequence described below. Although the steps are overly simplified for illustration, operators follow explicit procedure steps for each action leading up to and including the following activities:

1. Establish initial valve line-up for the transfer between two vessels.
2. Record initial tank levels.
3. Open pump isolation outlet valve.
4. Start pump.
5. Pump to a prescribed level in the receiving tank.
6. Stop pump.
7. Close pump isolation outlet valve.
8. Secure operations.



DCS Interface for Strip Effluent Feed Tank (SEFT) to Sludge Receipt and Adjustment Tank (SRAT) Transfer

Figure 1: Example Transfer Path

The primary steps above that manipulate equipment are carried out by a control room operator using a DCS panel like shown in Figure 1. An operator would need to be cognizant of the SEFT level (LI4063), the SEFT three-way valve (HIS4060) position, and the SRAT level (LI3025) among other indicators during the transfer. Directions to take readings of the tank levels, to ensure valve positions, and to start the pump are documented in procedures. Specific steps that have been determined high risks in a procedure may include an HPI control like ensuring that valve HIS4060 is aligned correctly for the transfer path shown above. Other HPI tools like those identified in Table I, such as the Task Review, Pre-job brief, Procedure Use and Adherence, Three-way Communications, and the Phonetic Alphabet, are also used during the evolution. Sometimes checklists are included in a procedure to ensure that each HPI element is conducted. The Process Controls Engineering (PCE) group evaluated DCS command sequences for high risk liquid transfers for opportunities to provide more secure operations by integrating HPI tools with the DCS logic. Although the DCS can be programmed to perform an entire transfer sequence, facility management desired that operators retain some manual control over a transfer sequence. A fully automated sequence might encourage operator complacency, causing them to miss monitoring critical facility operating parameters. The organization evaluated candidates for HPI tools guided by the overarching principle that retains operator responsibility. The best automation candidates were repetitive sequences where the DCS can guard against errors that result from operator distractions and complacency.

Table II describes the correlations between error precursors and the mitigating HPI tools identified in Table 1 that could be automated using the DCS in a transfer sequence like that depicted in Figure 1.

TABLE II: Integrating HPI With DCS Automation

Error Precursor	HPI Tool	DCS Automation
Lack of procedure familiarity	Task Preview Pre-job Brief Procedure Use and Adherence	An automated sequence can prompt the operator to perform a pre-job brief, and a specific sequence prompts the use of a specific transfer procedure.
Violated or improper procedure steps	Self-checking Peer-checking Concurrent verification Independent verification	Software interlocks can be included to require independent verification of specific steps before an equipment change of state is allowed.
Failure to record data	Post-job Review	Automatic logs can be generated to ensure that critical data is recorded.

Options for computer aids must be balanced with the potential for operator complacency induced by automation, a recognized error precursor. The strategy above preempts complacency by using interlocks, involving different people in the same task, using the computer to record information to combat the potential for to be data lost because of boredom with repetitive sequences, and minimizing the complexity of operations. [2]

The following example demonstrates how HPI tools were recently integrated with the DCS for all radioactive material transfers in the SRR Defense Waste Processing Facility. Prior to implementation, a manual supervisor permissive procedure had been developed to implement deliberate operations for high risk transfers. The manual steps and their corresponding DCS replacements are described below in Table III.

TABLE III: Example of Using DCS Controls to Enhance Deliberate Operations

Manual Sequence	DCS HPI-Enhanced Sequence
Operator requests to start a pump per procedure and initiates the pump command via the DCS console.	The initiating sequence is the same. An operator requests a pump start per procedure and initiates the pump command via the DCS console.
The DCS command is sent to the Control Room Manager for consideration.	The DCS request is sent to the Control Room Manager for consideration.
	HPI Enhancement: Computer checks for correct initial equipment alignment. Operation is denied if initial equipment configuration is incorrect (e.g., initial valve line-up).
The Control Room Manager fills out a paper checklist considering the start request. Specific questions answered included whether a pre-job brief has been conducted, an initial transfer path has been established, and whether the transfer is within the safety basis for facility operation.	The DCS provides an automated checklist for the Control Room Manager to fill out in request consideration.
The paper checklist is taken to the Facility Manager to review and approve the start request and checklist.	The DCS routes the checklist to the Shift Operating Manager* to review and approve the start request and checklist.
If approved, the paper checklist is returned to the Control Room Manager with permission to start the pump.	If approved, the DCS records the approval and gives the operator permission to start the pump.
	HPI Enhancement: Pump operation is denied if the time between the operator's initial request to start pump and the Shift Operating Manager's final approval exceeds a prescribed time.
The pump is started via the DCS.	The pump is started via the DCS.

*A change unrelated to this example was made regarding the parties responsible for approving the pump start.

The new DCS sequence retained the original controls and added checks and balances. The sequence is faster, more consistent, and provides a more definitive log of the requested operations (e.g., who performed each step, date/time stamps). Similarly, HPI tools can be incorporated in web-based engineering applications.

HPI and Engineering Calculations

Web-based computer applications, especially for repetitive calculations, provide many HPI benefits. The traditional paper-based calculations almost always depend upon a knowledge-based approach. Therefore, paper-based calculations are subject to typical knowledge associated error precursors such as task unfamiliarity, biases, lack of approach consistency, and distractions.

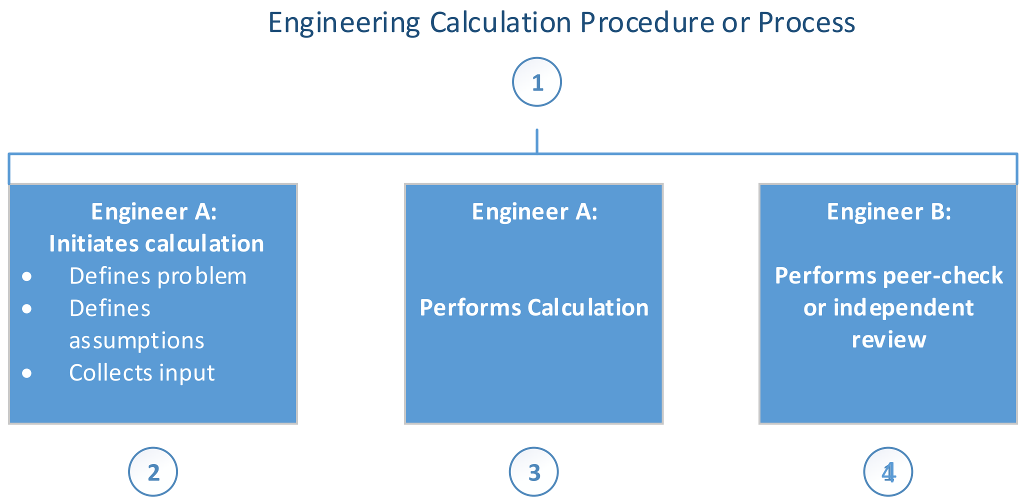


Figure 2: Typical Engineering Calculation Process

Figure 2 depicts a simplified engineering calculation process. Computer automation can replace the traditional paper-based and can include HPI tools with benefits described below:

An overall process, or procedure (Figure 2, Point 1), may have many possible calculation methods, even if it is repetitive. Different engineers may have different technical capabilities, different problem familiarity, and different calculation familiarity. Each of these differences present a range of error precursors. Computer automation can mitigate these issues by establishing a standardized approach. The advantages include documentation, knowledge transfer, consistent analytical methodology, and reduced dependency upon any one engineer performing the calculation.

Figure 2, Point 2, describes the initial activities involved in the engineering process specific to a generic calculation. A problem must be defined and framed. Input assumptions must be established. Other input may be required. Remaining error precursors may include different problem definitions, different input assumptions, and different variable inputs. A custom calculation can be designed to standardize the approach and solution. Basic input assumptions can be agreed upon, hard-coded in the software, and placed under configuration management. The application can also provide for peer-checking and independent review of variable input.

Figure 2, Point 3, represents the core calculation methodology. With a manual approach, different engineers may solve problems different ways. Although engineers may use the same core equations, the sequence of calculations, handling of significant digits, and conclusion format may differ from person to person. A computer application standardizes this methodology.

Figure 2, Point 4, represents peer-checks and independent reviews. Some independent reviews have an engineer unrelated to the task independently solve the problem using a different approach to validate the integrity of the specific calculation under review. As with the challenges described above for the subject calculation, the independent review process and procedure have some of the same challenges. An application, after having been tested and verified to provide accurate output over a defined range of input, can be trusted to provide accurate results without further output review. There are, however, conditions when an independent review or approval of the output may be desired. These reviews and approvals can be designed into the application to replace signature blocks on paper calculations. Finally, conclusions may be formatted differently by different engineers at Point 4. An application provides a standard reporting format highlighting the conclusions thus reducing the likelihood of misinterpretation.

An application mitigates risks by standardizing the method for repetitive calculations. The integrity of the calculation is no longer dependent upon anyone's memory or them having the capability to perform a specific calculation. The standard methodology is fixed and has been vetted by subject matter experts. Biases that may have factored into an individual calculation are precluded with standardization. The software life-cycle documentation transfers knowledge of specific calculation methodologies. Some error-likely conditions, however, may remain after encoding standard methods, and software designs should include HPI tools to preclude these errors. Evaluating a work process and flagging steps that have error precursors can highlight the potential uses for HPI tools.

A specific example of how this approach is being applied at SRR is the replacement of a manual waste qualification report with an automated report. Figures 3 and 4 are simplistic before-and-after depictions of a salt batch qualification process. They show how the principles described above were used to "error proof" the qualification process.

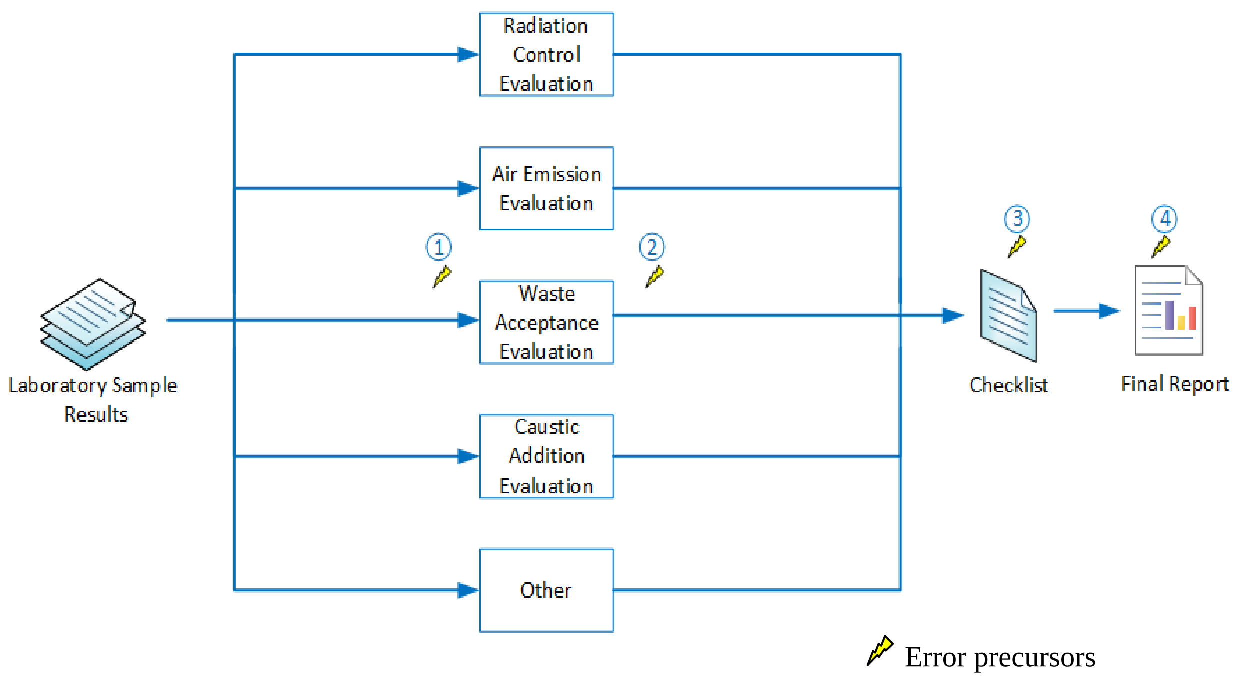


Figure 3: Salt Batch Qualification Manual Process

Figure 3 depicts the manual process for developing a final qualification report. When a laboratory sample report becomes available, several evaluations including a radiation control evaluation, an air emission evaluation, a caustic addition evaluation, and a waste characterization evaluation are initiated in parallel. Each evaluation undergoes development and approval as was described in Figure 2. Errors may occur due to erroneous input and erroneous calculations. Independent reviews are used in the development of each report to preclude potential errors.

Engineering decided to automate the waste characterization evaluation (Figure 4) because it is the most complex and time-consuming evaluation in the process. Over 600 different calculations are performed in this evaluation. The Electronic Waste Acceptance Characterization (eWAC) application was proposed to perform the calculations and to include HPI "error proofing" features.

Application error proofing starts in the requirements phase of the software life-cycle. The development team met with end-users to identify the process workflow and the steps with high potential for errors during the calculations. At each step where the team identified an error precursor, the team identified HPI tools that would mitigate the error risk. The requirements and design documents for the application included this mitigation functionality as standard features in the application.

The application designers then created a prototype to show users how the application would look and how the process would flow before programming began. The design documents included any calculations to be completed by the computer. These calculations were formally tested and reviewed using validated inputs and outputs before formal document approvals. This disciplined process standardized the calculation process and eliminated risk from computational errors. Independent reviews and approvals were inserted at critical points to ensure data and report integrity. The application ensures only qualified subject matter experts are involved with the development of a final report by restricting users and providing specific access rights for defined process roles.

Configuration management and change control are two more benefits of using an application. Any revision to the application requires that the software documents and code to be revised and that all changes be properly tested to protect application integrity. The configuration management process ensures that all end-users will be using the same fully tested version of software.

Table IV describes how the error-likely conditions identified in the manual process depicted in Figure 3 were addressed with HPI tools designed into the application as depicted in Figure 4.

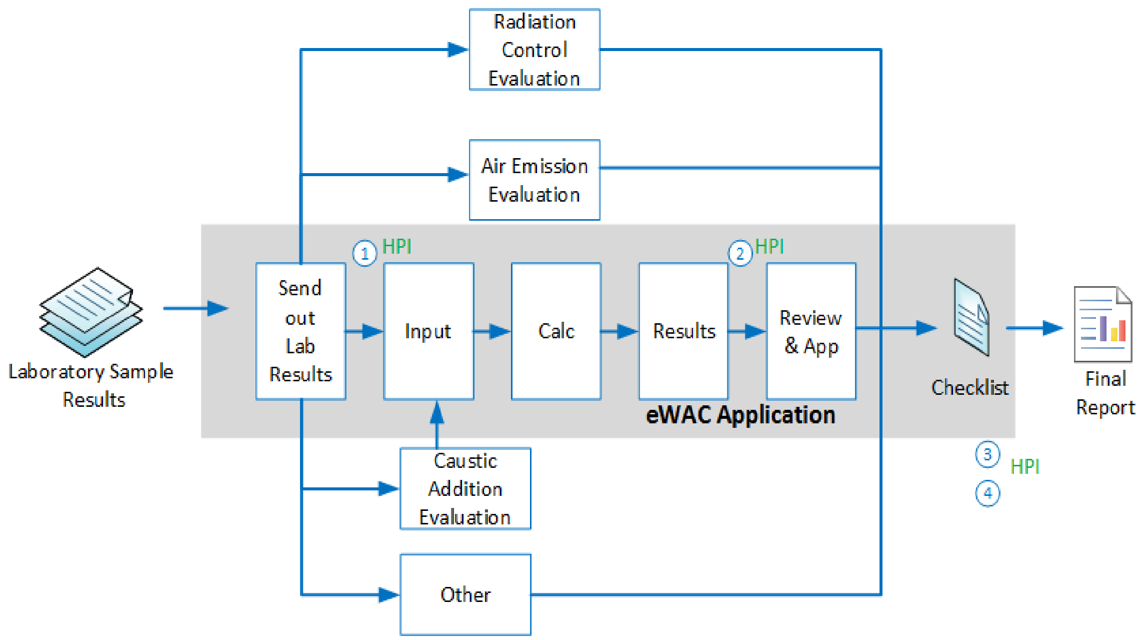


Figure 4: Salt Batch Qualification Automated Process

TABLE IV: eWAC Application Feature to Prevent Errors

Step	Action	Error Precursor	HPI Tool	Application Features
1	Engineer enters sample values from lab memo	Data entry error	Self-checking Independent Review	Validate data is within specified range to catch obvious data errors. Do not allow application to move forward to calculations until a second independent person has verified data with the lab memo.
2	Perform calculations Will be performed by the application in the future	Calculation Errors Different methodologies by people Interpretation requirements	Independent Review	Calculations are performed by the application that were coded in compliance with engineering calculation documents. Documents go through an approval process as technical basis for application. Application tested against known inputs and outputs for accuracy. Calculation results tested during acceptance test and on startup and do not require independent validation every time. Application = standard calculations.
3	Review report	Assumptions not clear Wrong conclusions reached Differences in supporting evidence between calculation results and report developed by engineer	Independent Review Approval process	Application includes links to all supporting evidence for reviewers. Application routes report and calculated values to independent reviewer. Application routes report to other approvers.
4	Issue report	Report not aligned with checklist to qualify batch Distractions	Standard workflow	Application notifies people that report is approved and checklist is approved. Organizes flow of information from single source.

CONCLUSION

Conceptually, integrating HPI tools with software is easy. The challenge is for process owners to think about what processes they might automate and then determine where they might insert conditional "gates" in their process to ensure that error precursors have been considered and mitigated. Then, the software code can be designed to include the conditional logic that will preclude errors. Table V summarizes how the different software tools described in Table I have been implemented via automation at SRR.

TABLE V: HPI Tool Implementation

HPI Tools	Automated Implementation
Task Preview Procedure Use & Adherence	These tools are integral to the process defined in the software requirements document. The use of a computer application standardizes an approach to a process. The requirements phase forces users to examine their process integrity and to define the limits of application use. The tasks for the application have been previewed, and use of the application ensures adherence to the expected process methodology.
Questioning Attitude Stop When Unsure Procedure Use & Adherence Validate Assumptions Peer-checking Independent Review	Conditional checks and permissions can be included in the software design. These statements might require a user to supply information before proceeding to the next step, causes them to stop if unsure, and ensures that valid input is being used. Conditional permissions can also be designed to require additional users to approve a step before the primary user is allowed to advance to the next step, thereby incorporating the concepts of peer-checking and independent validation. Finally, automatic data pulls from another computer application can help avoid the error precursors associated with typographical errors on input for which these HPI tools would catch.
Signature	Approvals are common in software work flow processes. These approvals can be augmented with messages, warnings, and guidance as appropriate.
Post-job Review	Post-job reviews can be improved by the use of computer logs designed to capture information of particular interest during application execution.
Decision-making	User interfaces can be designed to facilitate better clarity for decisions and to help guide a user when results may be found unacceptable or questionable.

Adherence to the principles of the software life-cycle can help ensure that HPI tools are well-understood and designed accordingly. Application development policies can communicate the expectations to include HPI tools in applications. The requirements phase can clearly define at what part of the process the tools should be included and what types of tools should be used. The software design phase can implement the required tools, and the test phase can ensure that the tools work as expected.

Computer applications and their development are excellent ways of capturing business knowledge. Application requirements and design documents are records of a precise methodology for performing a specific work scope. Development of these documents requires a subject matter expert to record what needs to be performed and to record assumptions and bases. To ensure knowledge transfer as requirements are gathered, the developing organization gets input from subject matter experts and develops a prototype for end users to view before any coding begins. This critical step closes the communication gap between users and developers. With clear documentation, a new process owner should be able to understand the technical baseline and approved methodology. In addition, the integrity of the process baseline can be maintained indefinitely using the software life-cycle.

REFERENCES

1. *DOE Standard: Human Performance Improvement Handbook, Volume 2, Human Performance Tools for Individuals, Work Teams, and Management*, DOE-HDBK-1028-2009, U.S. Department of Energy, Washington, D.C., June 2009.

2. *DOE Standard: Human Performance Improvement Handbook, Volume 1, Concepts and Principles*, DOE-HDBK-1028-2009, U.S. Department of Energy, Washington, D.C., June 2009.