



## **IV&V Services for ERP Implementations in Higher Education**

Independent Verification and Validation Services (IV&V) is a risk mitigation service to ensure clients receive the deliverables negotiated for. Although used in government and in the private sector, IV&V is not as common in high profile ERP implementations in Higher Education.

The purpose of this paper is twofold: 1) to provide a definition and background of IV&V services, and 2) to illustrate how these services can be utilized by institutions preparing to engage in an ERP contract for implementation services.

### **Definition and Background:**

The systems engineering process called Independent Verification and Validation (IV & V) has been part of the software and systems development arena since the 1970s and 80s, assisting development organizations in assuring software products' "correctness and quality".[2] IEEE established a standard regarding the components of an IV& V plan in 1986 [3].

Definitions of key terms via IEEE [6]:

- IV & V - "performed by an organization that is technically, managerially, and financially independent of the development organization"
- Verification and Validation – "process of determining whether the requirements of the system or component are complete and correct, the products of each development phase fulfill the requirements or conditions imposed by the previous phase, and the final system

or component complies with specified requirements"

- Verification - "process of evaluating a system or component to determine whether the products of a given development phase satisfy the conditions imposed at the start of the phase, providing formal proof of program correctness."
- Validation – "process of evaluating the system or component during or at the end of the development process to determine whether it satisfies specified requirements."



“Verification” ensures that we are building the product correctly while “Validation” examines whether we are building the right product – that the software products produced throughout the life cycles and the overall application/system meet the customers’ goals, objectives, and requirements. A design requirement states that the report in question will have four columns. Verification examines that the report will have four columns (at the end of the design and coding stages). Validation will check that the report is consistent with the overall goals and objectives – to obtain financial information for enrolled students by semester.

IV & V services are provided by an independent organization – often times by an outside vendor or a separate agency within the main organization. Note that the independence is three prong (technically, managerially, and financially) – hopefully to prevent any compromising in the IV & V provider’s ability to perform the services. Much as our fore-fathers realized the inherent benefits of established “checks and balances” in our system of government, so too have the leaders of highly dynamic and complex software and systems projects –recognizing the burden and complexity of managing the requirements, design and coding stages of a project. The assurance of quality and correctness needs to come elsewhere. (*As a tangential topic - it would be interesting to explore how IV & V contributed to the “silo” era.*)

Historically, IV & V services relate to mission-critical, highly complex and “expensive” projects such as military and defense projects, NASA, FAA projects (see IV & V Facility: <http://www.ivv.nasa.gov>). However, as software projects are becoming larger and more complex across all industries, IV & V providers are increasingly able to assist with these projects as well (see SAIC’s project work related to US Postal Service and State of Florida’s health provider network: <http://www.saic.com/software/independent-verification-and-validation.html>).

## Core Activities:

IV & V constitutes a full life cycle approach. The IV & V activities parallel the project’s life cycle and should verify and validate the project at appropriate milestone events.

The core activities are phase related and described by Walters [1]. The activities are:

- **Management Phase** – consists of the planning activities associated with the IV & V project – including how these activities will be reported and controlled (monitored).
- **Requirements Analysis** – ensures that the requirements are correct and of quality and developed in accordance to a standard process and procedure(s), as well as against a set of criteria. It consists of sub-phases: documentation analysis; interface requirements analysis; requirements traceability analysis. It also ensures that requirements are testable.
- **Design Analysis** – verifies that design documents and code can satisfy and trace back to requirements. The sub-phases are: design analysis planning; design product analysis; interface design analysis; code analysis; design traceability analysis.
- **Independent Testing and Analysis** – consists of the test planning, test execution, and test reporting activities. It also guides the IV & V team to meet with the project team to examine the overall test approach so that issues like test redundancy are addressed and executed effectively.

Walters [1] describes that there are inputs and outputs for each phase and that metrics are captured and reported – as part of the management activity.

How does one verify and validate within each phase of the project? One way is the use of

*inspections*, reviews, walk throughs, and audits. These are opportunities to examine work products such as requirement documents, design documents, and coding with Subject Matter Experts to ensure that the requirement threads are not lost and to evaluate the overall quality of the work products. These activities should include sets of criteria and other standards against which the work products are examined.

Another important activity embedded within IV & V is *traceability* – the listing of requirements, design, coding and testing elements within a grid, matrix or database and then linking these together. A traceability tool requires that the actual work products be documented confirming that business, system, and software requirements documents exist. Then, there is a labeling process of the work products to establish unique identifiers. This labeling process requires much evaluation, analysis, and grouping all determined by how many distinctive requirements are embedded within this design paragraph. Then, the elements are linked together, indicating how each satisfies its predecessor(s). The link could look like the following:

Bus Req -> System Req -> Software Req ->  
Design feature -> Coding Structure ->  
Testing

*Testing* is another critical activity, using the traceability matrix to ensure that all requirements and other elements are satisfied. Developers should test the “code” within the application to ensure that all modules are working properly, including the linkage among them. The IV & V testers should be independent from the developers’ testing. They can follow up by testing the application/system within a production-like environment, recording defects as needed and exercising various test types (i.e., stress, performance, regression, functional, etc).

Finally, a basic component of a mature organization is to *document the planning activities* and have documented procedures

and processes guides. IV & V advocates for this mature practice as these guides indicate that the organization has established best and repeatable processes. Repeatable and best practices evolve by learning from past mistakes and correcting them, assessing the results of a process that has been “tested” via multiple implementations, and documenting and sharing the results of a “true” subject matter expert. Walters described this practice within the Requirements, Design, and Testing phases [1]. In addition, he proposes a phase for Process Improvement – to capture lessons learned as well as means to evaluate new “tools, ideas, and concepts” into the IV & V process [1].

### Comparison with other Models/Disciplines:

IV & V as a systems engineering process does not exist in a vacuum. Its influences are seen in other process engineering models and disciplines, especially the Project Management Institute’s (PMI’s) Project Quality Management process, the Software and System Capability Maturity Models (SW-CMM & SE-CMM/EIS/IS 731), and the Capability Maturity Model-Integrated (CMMI).

First, by evaluating the three components (Project Quality Planning, Project Quality Assurance, Project Quality Control) of the Project Quality Management process (from PMBOK) [8], there are strong alignments to IV & V. PQM attempts to ensure that project performance (process) is executed effectively and efficiently as while also ensuring that the deliverables, both goods and services, are of stated quality.

Project Quality Planning phase identifies the inputs such as standards, regulatory issues, process and procedure documents and other sets of criteria that will serve as guideposts against which the project and products will be evaluated. Planning will review the project’s vulnerable areas and risks, and will then schedule the appropriate quality events

(reviews, inspections, and walk throughs). It will also determine the metrics to capture, and finally, plan for lessons learned and how the project will incorporate previous lessons learned as part of the overall process improvement cycle. These planning activities are seen in the IV & V model – the use of documented standards to evaluate the work products, the planning for different types of quality events (reviews, inspections) and the strong reliance on metrics. Project Quality Assurance and Project Quality Control are then the actual execution of the quality activities to monitor project performance as well as the correctness and quality of the products.

IV & V is also present in the CMMI as there are Process Areas (PAs) for each – one for Verification and one for Validation. These were derived from its predecessor EIS/IS 731 – Focus Areas (FAs) for Verify System and Validate System. In addition, when comparing the CMMI against the SW-CMM, Keefer and Lubecka [7] indicate that the CMM Key Process Areas (KPAs) of Peer Review, Software Product Engineering (which includes testing), and Software Quality Assurance (SQA) have medium to strong correspondence to the CMMI Verification and Validation PAs.

SQA (level 2 KPA) also is similar to the PMI's Project Quality Management, focusing on the planning and execution of activities that promote quality related to project performance and to the development of the deliverables.



From an historical perspective, IV & V most likely influenced both the Capability Maturity Models and PMI, an acknowledgement within the systems engineering and software development industries that IV & V represents a quality standard.

### IV&V for ERP Implementation Projects:

The planning and execution of IV & V activities should be tailored to the needs of the project. Therefore, the ERP implementation life cycle that is typical within the Higher Education environment may not receive the full suite of IV & V services nor require the same level of rigor.

An ERP implementation project does contain a Requirements phase (the discovery event hopefully identifies institutional goals and objectives as well as basic data collection, processing, reporting and communication needs of the individual offices – *some of the information may have been captured as part of the ERP Readiness event*). And while an **implementation project would typically not have design and coding phases** as would a software development project, the data migration/conversion phase does require design and mapping. There is also the issue of addressing the reporting needs of the clients, which may be part of the overall requirements phase, or addressed separately. Finally, there is a testing phase within the implementation project – as each new module is “going live”.

The **requirements**, including high-level goals and objectives, should be documented. They should be reviewed and evaluated for quality, as well as receive signoffs from appropriate stakeholders. The ERP project could benefit from labeling these requirements and capturing them within a traceability tool. This would allow for at least a link to a solution(s) within the product – *Requirement BR\_1 is being satisfied by data entry screen “xyz” in the Student*

*Registration Module.* The testing strategy, including conditions and cases, can also be documented within the traceability tool – again showing explicitly how testing will satisfy the requirements.

Regarding the *data migration/conversion process*, some type of data-mapping grid/tool could be used to control quality. The recommendation is that the mapping artifact is reviewed and approved by all key stakeholders and should be compared against the requirement traceability tool to ensure that the requirements thread is not broken.

If the reports and *reporting needs* were addressed separately from the requirements phase, then it's a good idea to use something similar to the data-mapping grid – mapping client current reports to the new ERP reports – maybe include a percentage of match (this new ERP report matches approx 80% of client's current report "xyz"). Again, this allows for review and agreement by all key stakeholders, it communicates expectations, especially when the ERP reports do not match completely with legacy reports. It also allows for capturing customized reports that may be part of the implementation – or at least allows for the assignment of a resource to it, including the anticipated due date for delivery.

*Testing* is a key activity within the ERP project. It should focus on test planning activities including scheduling, resources, specific test areas, defect tracking, etc.. The planning should also consist of reviewing the traceability tool(s) so actual test conditions and cases can be planned and documented within the tool and be ready for the test execution phase. For example, the new Business Office Form is tested via the following documented test cases: 1) Form is generated on-line (it loads properly); 2) Student information is displayed correctly - based upon the semester parameter; 3) Can select another student's ID and then the form is refreshed correctly; 4) Can print form; 5) Can send form as attachment via email. Documenting the test cases using the

traceability tool indicates that the testing is planned according to the requirements via this explicit linkage.

Finally, if not already in place, the implementation teams could benefit from *documented procedures* for each of these activities. For discovery and requirements gathering, these procedures could address such issues as: how to review and prioritize these requirements; how to elicit the reporting needs; the steps to conduct gap- analysis for "critical" legacy reports and ERP reports, and how to handle ad-hoc reporting needs. Data Migration and Conversion could document issues related to: how to examine the current legacy database; approaches on determining mapping between legacy and new ERP database; will there be a single identification record per individual or multiple ones as needed. Data Migration and Conversion could also document what the key interpretations are among the users within the institution regarding the legacy database (the institutional researcher may have subtle interpretations of the data that is different from the registrar – what are these and does it affect the data conversion process?). Finally, the approach to testing should be documented based upon past knowledge and experiences such as: the end users should run every report after the data migration process; every screen should be loaded; all customizations are tested, printed, and signed off. Having key activities guided by documented procedures allows for continual process improvement. As the process changes based on experiences, insights, environmental changes, and lessons learned, these enhancements should be documented accordingly, allowing the organization to become more mature by capturing and continually assessing repeatable steps and best practices.

While an ERP implementation project may not require the same IV & V rigor that may be present in a highly complex software development project, there still is a need to ensure that the requirements are documented

(including those associated with data migration and reporting), identified, and labeled. These various requirements should be reviewed with subject matter experts against a set of criteria and standards. These also should be captured as part of a

traceability tool so that all of the work products' elements are accounted for, the relationships among elements are intact, and testing can be planned and executed to ensure the quality of the overall software implementation.

## References:

1. D. Walters, *Writing an Effective IV&V Plan*, Crosstalk, November 2000;  
<http://www.stsc.hill.af.mil/crosstalk/2000/11/walters.html>
2. NASA Independent Verification and Validation Facility, *IV & V Overview*,  
<http://www.ivv.nasa.gov/ivvservices/IV&V/Overview.ppt>
3. IEEE Std 1012-1986, *IEEE Standard for Software Verification and Validation Plans*.
4. IEEE Std 1012-1998, *IEEE Standard for Software Verification and Validation*, March 9, 1998.
5. IEEE Std 1059-1993, *IEEE Guide for Software Verification and Validation*, December 2, 1993
6. IEEE Std 610.12-1990, *IEEE Standard Glossary of Software Engineering Terminology*.
7. G. Keefer, H.Lubecka, *CMMI Pathfinding in 45 Minutes*, AVOCA GmbH, 2002
8. PMI, *A Guide to the Project Management Body of Knowledge (PMBOK Guide)*, 2000 Edition;
9. Lewis, Robert O. *Independent Verification and Validation: A Life Cycle Engineering Process for Quality Software*. New York: John Wiley & Sons, Inc., 1992.

**If you found this article helpful and would like to read more, please go to our website at [www.collegiateproject.com](http://www.collegiateproject.com) and browse our ERP Library.**