



VISTA Solutions: Configuration Management | Project Management | Code Reusability | Quality Assurance | GOOP Tool

VISTA Services: GOOP Training | Process Improvement | Custom Training & Consulting



Are YOU Maximizing
the productivity of YOUR team?

Achieving CMMI Levels 2 and 3 with LabVIEW[®]

by Wyatt Meek

VISTA Product Manager

VI Engineering, Inc.



Foreword	3
General Problem Areas.....	3
LabVIEW Root Issues	4
Solution	5
Introduction.....	6
What is CMMI?	6
Benefit or Burden	7
Achieving Level 2 per CMMI SE/SW - Staged.....	8
Generic Goals	8
Requirements Management	9
Project Planning	10
Project Monitoring and Control.....	10
Supplier Agreement Management	11
Measurement and Analysis	12
Process and Product Quality Assurance	12
Configuration Management.....	13
Level 2 Summary.....	14
Planning for Level 3.....	15
Technical Solution	15
Verification.....	16
Validation	16
Risk Management.....	16
Summary	17



Foreword

While many authors have written about the benefits and details of CMMI, few have focused on the impact CMMI has on the test & measurement (T&M) community. To understand this impact, it is important to first understand the differences between the T&M community and other development groups.

Like many development groups, the T&M community is predominantly focused on the design and build of systems. At the heart of all of these systems is software, but within the software there is one primary difference between the T&M and the other development groups. Software written by most development groups is in any of the common text based programming languages such as C, C++, VB, Java, PHP, and others. In the test engineering community, however, LabVIEW is one of the dominant programming languages. As National Instruments, the creators of LabVIEW, states, "LabVIEW is the graphical development environment for creating flexible and scalable test, measurement, and control applications rapidly and at minimal cost. With LabVIEW, engineers and scientists interface with real-world signals, analyze data for meaningful information, and share results and applications. Regardless of experience, LabVIEW makes development fast and easy for all users." While this statement is true¹, a number of issues can arise when developing large applications. Unlike their text programming counterparts, many LabVIEW developers are engineers and scientists that lack intensive software engineering training or experience. Therefore, it is common for a new LabVIEW developer to begin developing small scale systems without any formal training or software process in place. Over time, these developers begin working together to develop larger systems. It is at this point many problems begin to surface through over budget projects, missed deadlines, and quality concerns.

General Problem Areas

While there are many issues that may lead to over budget projects, missed deadlines and lower than expected quality, we will highlight five general problem areas. The first problem area is poor project management. Signs of poor project management are difficulty quantifying development progress, inconsistent estimates and inconsistent results between projects or people. When projects surpass deadlines, project management is usually the first place managers investigate; however it is often not the lone culprit.

The second problem area is poor configuration management. Configuration management is defined as the control of configuration items (VIs, configuration files, test plans, requirements, process guidelines, etc.) from design through retirement. Signs of poor configuration management are previously fixed bugs returning, overwriting software changes, inability to know which version belongs in a baseline, inconsistencies between two "identical" stands, and the inability to duplicate or modify software years in the future.

Requirements management is similar to configuration management and is such a large area that it is often viewed separately. Common problems with requirements management are poorly defined requirements, lack of change notification, lack of bidirectional traceability² and inability to accurately assess the impact of changes.

¹[http://zone.ni.com/devzone%5Cconceptd.nsf/webmain/31D31213959B38BA86256BC2006653EA/\\$File/WP2260.pdf](http://zone.ni.com/devzone%5Cconceptd.nsf/webmain/31D31213959B38BA86256BC2006653EA/$File/WP2260.pdf)

² Per "Capability Maturity Model Integration (CMMI), Version 1.1 (CMMI-SE/SW, V1.1)," traceability is defined as the evidence of an association between a requirement and its source requirement, its implementation, and its verification.



Reuse is often thought of as a nice benefit in a project and not a problem area. By having a solid reuse strategy in place organizations not only reduce development time (less development and testing), but also increase quality by using modules that have already been proven in many applications. With this in mind, it is important to be proactive about reuse in order to maximize the benefits. Signs of a poor reuse strategy are developers reinventing the wheel, reuse within islands, and difficulty reusing code due to lack of a common style, documentation, structure and the ability to find reusable code.

The last problem area is quality assurance. Within quality assurance we include the verification and validation of software and systems. Common signs of a poor QA plan are the inability to differentiate between verification and validation, lack of traceability, lack of testing, and more bugs being found during the implementation phase than in the testing phase. Overall one of the greatest problems that lead to poor quality assurance is the lack of, or lack of adherence to, a development process

LabVIEW Root Issues

It is important to note that the general problem areas listed are not specific to LabVIEW; they are just more predominant in LabVIEW, due to the lack of formal software development processes and practices throughout the user community. There are, however, a number of root issues specific to LabVIEW that make implementing processes and practices challenging.

The first issue is the lack of repeatable metrics to gauge software development, a primary component in project management. An example of a basic metric used by text programmers is SLOC (Source Lines of Code). SLOC is a measure of the number of physical lines in a program and is recommended by the Software Engineering Institute (SEI) at Carnegie Mellon University (CMU/SEI-92-TR-019). Since LabVIEW is a graphical programming language, there are no lines of code to count. Instead LabVIEW developers require a different metric. Two available concepts are GOBS™ (Graphical Objects) and nodes. These metrics are based on sheer size of the application and are similar in function to SLOC. Both may be used to help quantify development progress and create estimates, although using nodes requires greater style consistency between developers. More advanced metrics, such as cyclomatic complexity³, are used to test the complexity of an application and may be used in validation to show path coverage and assist with risk assessment. Until the release of the VISTA® Complexity and Path Managers, calculating cyclomatic complexity for LabVIEW applications is impossible.

The second root issue is associated with the LabVIEW file handling. LabVIEW may relink source code when a high level VI (VI stands for Virtual Instrument and is similar to a function in a text based program) is opened, occasionally causing VIs with the same names to be incorporated incorrectly into a project. This is especially apparent when a sub-VI with a particular name is present in memory and then a parent-VI is opened that links to a VI with the same name but in a different location on the file system (and potentially different behavior). Modifying and saving the parent VI will cause it to link to the incorrect sub-VI from then on, and the hierarchy will no longer be under configuration management. That failure makes traceability and verification impossible.

The third root issue is the lack of a formal design methodology among many LabVIEW groups. As LabVIEW is a powerful language that is easy to learn and use and geared towards test and measurement applications, many LabVIEW developers are engineers or technicians that lack a formal software development background. As a result many do not follow a formal development style or leverage design patterns. This often leads to problems with reuse and delays when they begin tackling much larger and more complex applications.

³ Per http://www.sei.cmu.edu/str/descriptions/cyclomatic_body.html, cyclomatic complexity measures the number of linearly-independent paths through a program module



One of the underlying issues that spans every problem area is the lack of a software development process. When working on small applications that do not require multiple resources, it is often easy to handle the design, development and release. For example a developer may maintain requirements in his head and build the system directly on the production stand. As the developer modifies requirements, he will then make the appropriate changes to the stand. However, problems often begin occurring due to the lack of a formal development process as applications grow in physical size, complexity, and with the addition of more developers.

Solution

The solution to addressing these problems with LabVIEW is through VISTA. VISTA is comprised of Tools (VISTA and 3rd Party), Training Services and Consulting Services designed to maximize ROI (Return on Investment) through process improvement.



Introduction

With the PC revolution came a flurry of standards and processes to address software and system development. Nearly every standards body, industry and government organization developed their own guidelines to tackle specific to general issues; many of which had significant overlap⁴. Soon this became a burden to organizations due to the costs associated with training, certification, and assessments. Further complications existed due to the differences in terminology between guidelines. To solve this problem the Software Engineering Institute (SEI) at Carnegie Mellon University, with the help of the DoD (Department of Defense) and industry leaders, created the Capability Maturity Model Integration (CMMI) initiative. While the goals and concepts behind CMMI are independent of programming languages, the implementation and tools used for process adherence will need to vary between LabVIEW and text based languages in order to account for differences in the code created (please refer to the [Foreword](#) to learn more about these differences). This paper will highlight these differences and provide tips and techniques for meeting the CMMI requirements with LabVIEW.

What is CMMI?

The purpose of CMMI is to provide guidance for improving an organization's processes and its ability to manage the development, acquisition, and maintenance of products or services. CMMI places proven approaches into a structure that helps companies appraise its organizational maturity or process area capability, establish priorities for improvement, and implement these improvements. There are multiple CMMI models available. Consequently, you need to be prepared to decide which CMMI model best fits your organization's process improvement needs. The place to start is to select which disciplines you want to include in your process improvement program and select a model representation. The two representations are *staged* and *continuous*.⁵

Since LabVIEW as a language is focused on the Test & Measurement community, the most applicable disciplines are in the CMMI-SE/SW model which stands for Systems Engineering and Software Engineering. Choosing a representation for the CMMI-SE/SW model largely depends on the organization's requirements. The *continuous* representation enables organizations to select which process areas are best suited for continuous improvement and pursue them in any order. The *staged* representation is designed to provide maximum benefit through an implementation sequence of stages or Maturity Levels (ML). It also requires an organization to satisfy all process areas in a given maturity level before advancing to the next level. It is expected that all DoD divisions and contractors select the staged representation due to DoD requirements. Both representations are comprised of the same process areas.

This paper will assume the adoption of the staged representation. The staged representation consists of the following five levels.

Level 1 – Initial

The software process is characterized as ad hoc, and occasionally even chaotic. Few processes are defined, and success depends on individual effort and heroics.

Level 2 – Repeatable

⁴ From the "Concept of Operations for the CMMI" at www.sei.cmu.edu/CMMI/background/conops.html

⁵ From the "Getting Started with CMMI Adoption" at <http://www.sei.cmu.edu/CMMI/adoption/CMMI-start.html>

Basic project management processes are established to track cost, schedule, and functionality. The necessary process discipline is in place to repeat earlier successes on projects with similar applications.

Level 3 – Defined

The software process for both management and engineering activities is documented, standardized, and integrated into a standard software process for the organization. All projects use an approved, tailored version of the organization's standard software process for developing and maintaining software.

Level 4 – Managed

Detailed measures of the software process and product quality are collected. Both the software process and products are quantitatively understood and controlled.

Level 5 – Optimizing

Continuous process improvement is enabled by quantitative feedback from the process and from piloting innovative ideas and technologies.⁶

Each level, with the exception of 1, is comprised of multiple process areas. In a staged representation all process areas in a given level must be successfully met in order to achieve that level. Each process area is comprised of specific goals and a single generic goal; all of which must be met in order to successfully meet that process area. The generic goal is the same for all process areas within a given level, whereas specific goals are unique to the associated process area. Specific goals and generic goals are then comprised of specific practices and generic practices respectively; which are considered expected model components. Expected components describe what an organization will typically implement to achieve a required goal. Either the practices as described, or acceptable alternatives to them, are expected to be present in the planned and implemented processes of the organization before goals can be considered satisfied.

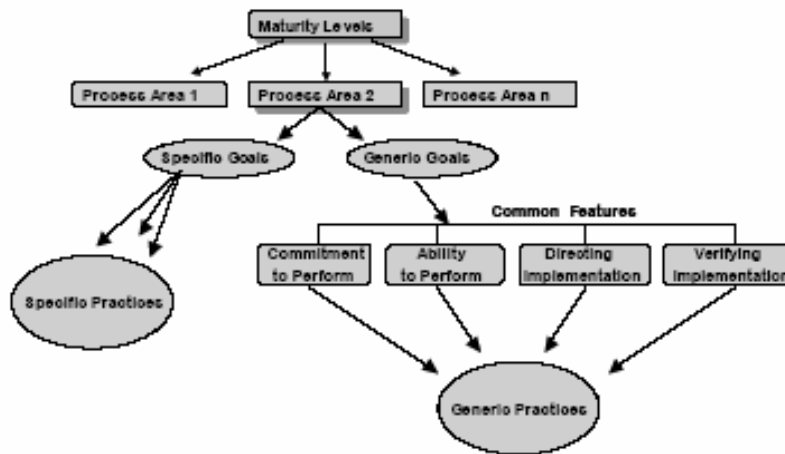


Figure 1: CMMI Model Components

Benefit or Burden

⁶ From the “Capability Maturity Model Integration (CMMI), Version 1.1 (CMMI-SE/SW, V1.1)” at <http://www.sei.cmu.edu/pub/documents/02.reports/pdf/02tr002.pdf>



While CMMI at first glance may appear to be a burden, there are a number of documented cases to prove otherwise. Below are a few of the actual results realized after implementing CMMI.⁷

- Reduced Cost
 - 20% reduction in unit software costs (Lockheed Martin M&DS)
- Decreased Development Time
 - Decreased the average number of days late from approximately 50 to fewer than 10 (General Motors)
- Increased Quality
 - Reduction in defects found from 6.6 per KLOC (1000 Lines of Code) to 2.1 over 5 causal analysis⁸ cycles (Northrop Grumman)
- Improved Customer Satisfaction

While results will vary between organizations and maturity level, nearly all organizations analyzed achieved a positive return.

Achieving Level 2 per CMMI SE/SW - Staged

To achieve CMMI Level 2 an organization must meet the Generic Goals (GG) and Specific Goals (SG) for 7 process areas. Those areas are:

- Requirements Management
- Project Planning
- Project Monitoring & Control
- Supplier Agreement Management
- Measurement & Analysis
- Process & Product Quality Assurance
- Configuration Management

Generic Goals

The Generic Goal for all process areas in level 2 is to Institutionalize a Managed Process, which consists of 10 Generic Practices (GP).

GG 2 Institutionalize a Managed Process

- GP 2.1 Establish an Organizational Policy
- GP 2.2 Plan the Process
- GP 2.3 Provide Resources
- GP 2.4 Assign Responsibility
- GP 2.5 Train People
- GP 2.6 Manage Configurations
- GP 2.7 Identify and Involve Relevant Stakeholders
- GP 2.8 Monitor and Control the Process
- GP 2.9 Objectively Evaluate Adherence
- GP 2.10 Review Status with Higher Level Management

⁷ <http://www.sei.cmu.edu/pub/documents/03.reports/pdf/03sr009-revised.pdf>

⁸ Per "Capability Maturity Model Integration (CMMI), Version 1.1 (CMMI-SE/SW, V1.1)," casual analysis is defined as the analysis of defects to determine their cause



For some organizations, meeting this generic goal simply requires mapping their current processes to the generic practices. As noted earlier, many Test & Measurement organizations fail to have detailed processes and thus require more assistance.

For those organizations the VISTA consulting services are available to assist in a three phased approach. The first phase (Analysis) consists of a process assessment to understand the organizational goals, current development processes, development and process tools and the guidelines which may be mapped to CMMI with minimal alteration. The result of this phase is a structured improvement plan to achieve maturity level 2 and the creation of a vision team. The second phase (Development) consists of mapping available guidelines to the generic and specific practices within each process area. Where there aren't any guidelines in place, the VISTA consulting services will provide documentation and tailor it to meet both CMMI and customer requirements. During this phase the development lifecycle (Modified Waterfall, Staged, etc) is defined to match organizational goals and project development. In this phase, tools are also selected as needed to assist with GP 2.6, 2.8 and 2.9. The final phase (Implementation) consists of implementation of the process, documentation, guidelines and tools. This phase includes training for the organization on the process, process related tools, and development guidelines. It is typical at the conclusion of this phase to provide an analysis of what is needed to achieve the next set of goals.

Requirements Management

Requirements Management (RM) is the ability to manage requirements and the change impact throughout the development lifecycle. This is extremely important in a lifecycle process based on iterative development since changes are ongoing. There is a single goal within the RM process area; that goal is to manage requirements. That goal is further defined as "Requirements are managed and inconsistencies with project plans and work products are identified." To assist in implementing the SG 1, CMMI lists five specific practices.

SG 1 Manage Requirements

- SP 1.1 Obtain an Understanding of Requirements
- SP 1.2 Obtain Commitment to Requirements
- SP 1.3 Manage Requirements Changes
- SP 1.4 Maintain Bidirectional Traceability of Requirements
- SP 1.5 Identify Inconsistencies between Project Work and Requirements

SP 1.1 and 1.2 require a defined process to capture requirements and ensure their feasibility. Once the requirements are accepted, they then need to be managed for changes. SP 1.3, 1.4 and 1.5 address this process. Once a requirement is changed there must be a way to understand the impact that change will have on other requirements and work products. This link is often referred to as traceability. With bidirectional traceability it is then easy to identify discrepancies between work products and the requirements. It is possible to meet SP 1.3, 1.4 and 1.5 through intensive manual tracking and involvement. Unfortunately, this approach is prone to human error and can be costly for a large group. There are a variety of COTS packages available to assist in meeting these three areas with prices ranging from a few hundred dollars to many thousands per user. High end users may be interested in tools such as IBM Requisite Pro[®] or Telelogic Doors[®]. These tools are further enhanced with integration to the LabVIEW development environment through VISTA to ensure bidirectional traceability between requirements, test plans, and VIs. For those users that aren't interested in the high end tools, an easy solution utilizes a Source Code Control (SCC) provider, such as Microsoft Visual SourceSafe[®] or IBM ClearCase[®], with the VISTA Traceability Tracker. Both solutions may be tailored to meet CMMI requirements; therefore the VISTA Consulting Services are available to help devise a solution to fit each customer's needs.



Project Planning

The Project Planning (PP) process area consists of three specific goals. These goals are focused on creating reliable estimates, developing a project plan based on those estimates, and gaining commitment to the plan by the stakeholders. Since changes are likely to occur throughout the project, a plan needs to be devised to address changes. Each goal consists of a number of specific practices designed to assist with implementation.

SG 1 Establish Estimates

- SP 1.1 Estimate the Scope of the Project
- SP 1.2 Establish Estimates of Work Product and Task Attributes
- SP 1.3 Define Project Life Cycle
- SP 1.4 Determine Estimates of Effort and Cost

SG 2 Develop a Project Plan

- SP 2.1 Establish the Budget and Schedule
- SP 2.2 Identify Project Risks
- SP 2.3 Plan for Data Management
- SP 2.4 Plan for Project Resources
- SP 2.5 Plan for Needed Knowledge and Skills
- SP 2.6 Plan Stakeholder Involvement
- SP 2.7 Establish the Project Plan

SG 3 Obtain Commitment to the Plan

- SP 3.1 Review Plans that Affect the Project
- SP 3.2 Reconcile Work and Resource Levels
- SP 3.3 Obtain Plan Commitment

There are a variety of methods and tools available to meet SG 1 for software projects, such as COCOMO, Delphi, SEER, and others. Each method requires a variety of inputs to assess risk, size and other factors. Many of these inputs are dependent on historical data from similar projects. Without reliable historical data the chances of creating an accurate estimate diminish quickly. Therefore, it is important to have access to past projects metrics. For LabVIEW, this is easily accomplished through the VISTA Metrics Tracker which keeps track of nodes, GOBS and man time spent by project, user and lifecycle phase.

Once a reliable estimate is established, a project plan is needed to monitor progress and ensure successful completion. Developing this plan requires a defined process to account for potential issues such as project risks, data management, resources, skills, and stakeholder involvement. COTS packages such as MS Project® and Primavera® are useful in developing the initial plan to a corresponding budget and schedule requirements.

The last goal within the PP process area is Obtain Commitment to the Plan. This includes getting commitment to the initial plan and any and all future modifications. Due to this requirement SG 3 needs to be closely linked to the Requirements Management and Configuration Management process areas to ensure traceability and commitment to the correct plan.

Project Monitoring and Control

After a project plan is established and the project begins, the Project Monitoring and Control (PMC) process area accounts for quantifying progress and providing corrective actions based on actual vs plan deviations. The process area consists of two specific goals and specific practices to assist with implementation of each.

SG 1 Monitor Project Against Plan

- SP 1.1 Monitor Project Planning Parameters
- SP 1.2 Monitor Commitments



- SP 1.3 Monitor Project Risks
- SP 1.4 Monitor Data Management
- SP 1.5 Monitor Stakeholder Involvement
- SP 1.6 Conduct Progress Reviews
- SP 1.7 Conduct Milestone Reviews
- SG 2 Manage Corrective Action to Closure
 - SP 2.1 Analyze Issues
 - SP 2.2 Take Corrective Action
 - SP 2.3 Manage Corrective Action

In order to meet SG 1, an organization must have metrics to monitor the project against. Therefore, this process area is closely linked to the Measurement and Analysis process area which defines the metrics to be used for project plans and project monitoring. Examples of metrics for project monitoring include SLOC, GOBS, nodes, status, hours, issues, and earned value. It is often useful to track multiple metrics to gain more insight into the actual performance. For LabVIEW the best way of tracking actual results is through the VISTA Project Management Tool and Metrics Tracker which record time, GOBS, nodes, status and issues and compares it by user, file and lifecycle phase. These results may then be displayed through built in actual vs planned reports or exported to MS Project for progress and milestone reviews.

In order to meet SG 2, an organization must have a process to address issues and deviations from the plan once they occur. There are a number of COTS based tools to help automate issue tracking from inception to resolution. The benefits of these tools are heavily based on the quantity of issues and the size of the organization. Many smaller organizations are successful by implementing issue tracking spreadsheets controlled with Configuration Management. For LabVIEW developers the VISTA Issue Tracker is integrated within the development environment enabling developers to easily assign and review issues for specific VIs or the project.

Supplier Agreement Management

One area that is often neglected in a process improvement plan is Supplier Agreement Management (SAM). While an organization may be highly efficient in developing and managing their internal tasks, the overall project could be at risk if the suppliers are not able to deliver an acceptable solution. Within the SAM process area are two goals.

- SG 1 Establish Supplier Agreements
 - SP 1.1 Determine Acquisition Type
 - SP 1.2 Select Suppliers
 - SP 1.3 Establish Supplier Agreements
- SG 2 Satisfy Supplier Agreements
 - SP 2.1 Review COTS Products
 - SP 2.2 Execute the Supplier Agreement
 - SP 2.3 Accept the Acquired Product
 - SP 2.4 Transition Products

To meet SG 1, an organization must have a process to determine the source of products (COTS, internal, outsource), then evaluate the available suppliers on predetermined criteria and finally establish a formal agreement to set expectations. With supplier agreements in place, the organization must ensure that the desired supplier's products or services meet all of the requirements of the project. Once this is determined, the organization may proceed in executing the agreement. Prior to final acceptance the organization must once again ensure that the delivered product or service meets the requirements outlined in the agreement. After acceptance, the organization must then integrate the product or service into the project. To assist



in meeting the Supplier Agreement Management process area, the VISTA Consulting Services offer custom plans and templates to meet SG 1 and 2.

Measurement and Analysis

The purpose of the Measurement and Analysis (MA) process area is to create quantifiable results for all aspects of the engineering, management and business areas. These results are the basis for many other process areas including Project Planning, Project Monitoring and Control, and Quantitative Project Management (ML 3). The initial focus of the MA process areas is at the project level; however the results may prove useful in addressing organizational and/or enterprise wide needs.⁹ The MA process area includes two specific goals and many specific practices to guide implementation.

- SG 1 Align Measurement and Analysis Activities
 - SP 1.1 Establish Measurement Objectives
 - SP 1.2 Specify Measures
 - SP 1.3 Specify Data Collection and Storage Procedures
 - SP 1.4 Specify Analysis Procedures
- SG 2 Provide Measurement Results
 - SP 2.1 Collect Measurement Data
 - SP 2.2 Analyze Measurement Data
 - SP 2.3 Store Data and Results
 - SP 2.4 Communicate Results

To successfully implement SG 1, an organization needs to understand improvement objectives and devise a method of measuring those objectives. For LabVIEW, there are a variety of measures available to meet specific improvement objectives. The following table outlines a sample list of project objectives and corresponding measures.

Table 1: Measurement Objectives and Measures

Objective	Measure
Monitor Development Progress	Man Time Spent GOBS Nodes
Monitor Budget	Cost Earned Value
Monitor Quality	Defects Peer Review Coverage Path Coverage
Monitor Complexity	Cyclomatic Complexity

Once the measures are defined, SG 2 requires a process to collect, analyze and report results. For LabVIEW projects this is easily accomplished through integration of the VISTA Metrics Tracker with the VISTA Project Management Tool and MS Project.

Process and Product Quality Assurance

The purpose of Process and Product Quality Assurance is to provide staff and management with objective insight into processes and associated work products. The practices in the PPQA process area ensure that planned processes are implemented, while the practices in the

⁹ From the "Capability Maturity Model Integration (CMMI), Version 1.1 (CMMI-SE/SW, V1.1)" at <http://www.sei.cmu.edu/pub/documents/02.reports/pdf/02tr002.pdf>



Verification (ML 3) process area ensure that the specified requirements are satisfied. These two process areas may on occasion address the same work product but from different perspectives. Projects should take care to minimize duplication of effort.¹⁰

SG 1 Objectively Evaluate Processes and Work Products

SP 1.1 Objectively Evaluate Processes

SP 1.2 Objectively Evaluate Work Products and Services

SG 2 Provide Objective Insight

SP 2.1 Communicate and Ensure Resolution of Noncompliance Issues

SP 2.2 Establish Records

To successfully meet SG 1, an organization needs to develop a plan to objectively evaluate the processes and products. An organization may evaluate the processes and products by using predefined criteria to measure adherence to process descriptions, standards and procedures. For products, it is important to complete the evaluations at incremental milestones and prior to customer delivery. After completing the evaluation an organization must communicate the results to relevant stakeholders, address all noncompliance issues, and establish records to meet SG 2.

Configuration Management

The purpose of the Configuration Management (CM) process area is to ensure the integrity of processes and products throughout the project lifecycle. This is accomplished through the implementation of three specific goals and many specific practices.

SG 1 Establish Baselines

SP 1.1 Identify Configuration Items

SP 1.2 Establish a Configuration Management System

SP 1.3 Create or Release Baselines

SG 2 Track and Control Changes

SP 2.1 Track Change Requests

SP 2.2 Control Configuration Items

SG 3 Establish Integrity

SP 3.1 Establish Configuration Management Records

SP 3.2 Perform Configuration Audits

The first step in meeting the CM process area is the establishment of baselines¹¹. The CM process must first outline which items are required parts of a baseline. Typical configuration items may consist of code, support files (.mnu, .ini, .dll), documentation, requirements, and process guidelines. Once the configuration items are identified, organizations need to store the configuration items in a configuration management system. The configuration management system is responsible for tracking and controlling changes to configuration items in a central repository. There are a number of COTS CM repositories such as Microsoft Visual SourceSafe and Rational ClearCase. When combined with the COTS CM repositories the VISTA Configuration Management and Project Management Tools offer the easiest solution to create, restore and deploy baselines for LabVIEW applications.

¹⁰ From the "Capability Maturity Model Integration (CMMI), Version 1.1 (CMMI-SE/SW, V1.1)" at <http://www.sei.cmu.edu/pub/documents/02.reports/pdf/02tr002.pdf>

¹¹ Per "Capability Maturity Model Integration (CMMI), Version 1.1 (CMMI-SE/SW, V1.1)," a baseline is defined as the configuration information formally designated at a specific time during a product's or product component's life.



To meet SG 2, an organization requires a defined process to track and control changes. Once a change request occurs, the impact of the change should be assessed in order to make corrective action to the project plan. Once a change is approved and made, the baseline must be updated.

The final step in implementing the CM process area is establishing integrity of the baseline. The first part to establish integrity is to maintain records about the configuration items. One way to accomplish this is to assign a status to each configuration item and version. The second part to establish integrity is through configuration audits. The role of the audits is to ensure the integrity of the items in the configuration management system and the baselines deployed. When deploying LabVIEW source code instead of executables, the best way to meet SG 3 is with the VISTA Configuration Management Tools and Integrity Tracker, which include an integrity check algorithm to ensure files match the correct baseline.

Level 2 Summary

In order to meet Maturity Level 2, an organization needs to have well defined processes for each process area, tools to meet or automate processes, and training. For organizations that find implementing all of the areas at once difficult, choosing a few focus areas and the continuous representation may be a better choice. With over 100 man years of LabVIEW experience, process improvement tools, and advanced training options, the VISTA team is available to assist any organization in meeting their CMMI and process improvement goals.

Figure 2: CMMI Implementation with VISTA shows an example of how VISTA can help an already structured development team improve to meet CMMI Level 2 and begin a foundation to meet Level 3.

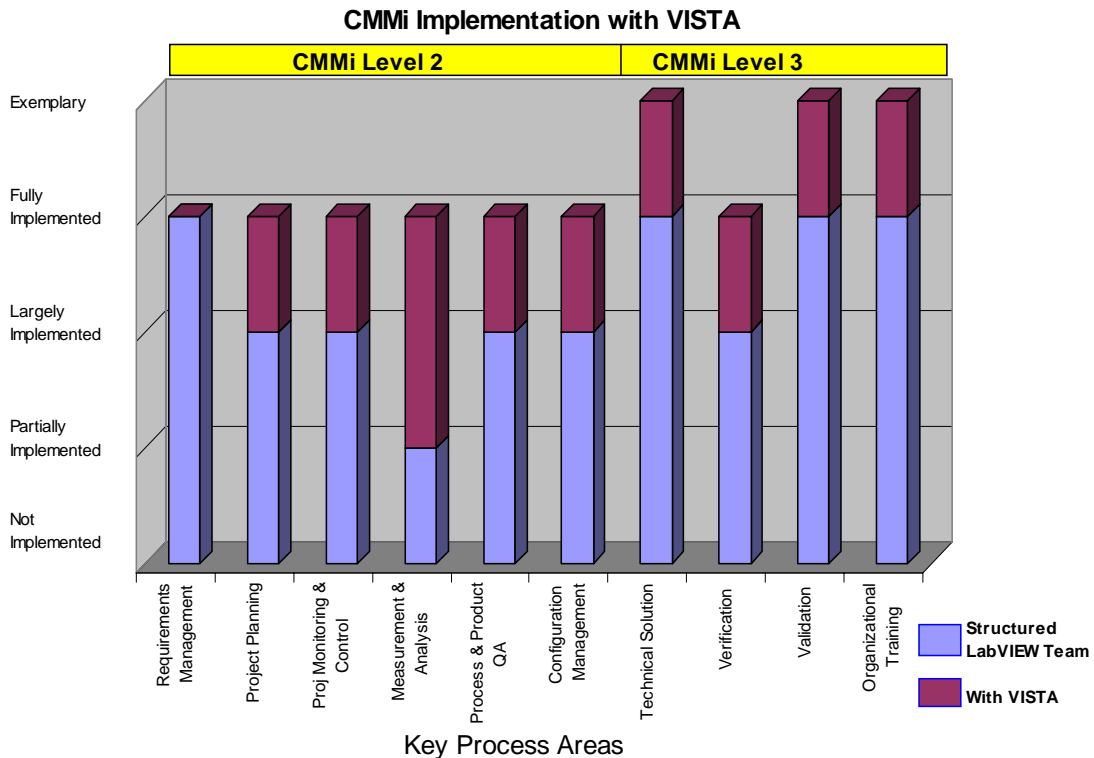


Figure 2: CMMI Implementation with VISTA



Planning for Level 3

Achieving Level 2 is a large accomplishment for most organizations resulting in a vast array of benefits. For other organizations Level 2 is just the beginning for continuous process improvement. While Level 2 is focused on creating a process to manage projects in order to gain repeatable results on similar projects; the goal of Level 3 is to create a process for the entire organization that defines not only how to manage the project but how to develop and test it as well. Since the goal is to have a single process for the entire organization it is important to define tailoring guidelines to accommodate all types of projects.

To achieve Level 3, an organization must meet 11 new process areas. In addition, the organization must also implement a new generic goal for all Level 2 process areas. That goal is to institutionalize a defined process; which means the process used for each process area must be part of the organizational standard process or a tailored version thereof, and the used process must contribute to building the organizational assets. Meeting this generic goal does not require the addition of new tools, but does require the creation of a more tightly coupled process for the entire organization and appropriate tailoring guidelines. The 11 new process areas that comprise Level 3 are:

- Requirements Development
- Technical Solution
- Product Integration
- Verification
- Validation
- Organizational Process Focus
- Organizational Process Definition
- Organizational Training
- Integrated Project Management
- Risk Management
- Decision Analysis and Resolution

Since Level 3 is focused on creating a process for the entire organization, a Level 3 process should not be affected by the differences between developing a LabVIEW program or building an instrument. Instead, these differences need to be addressed through tailoring guidelines specific to each type of project. Instead of going through each process area, the remaining portion of this paper will highlight four specific process areas that may require tailoring for LabVIEW projects or where tools are available to aid in adherence. These process areas are Technical Solution, Verification, Validation, and Risk Management.

Technical Solution

For a LabVIEW project, the Technical Solution (TS) process area includes the design, development, and implementation of any components (models and VIs) to defined requirements. Several should be addresses to move from the ad hoc approach to application design that is typically found in the T&M community to more formal processes. They include application design training via courses such as the LabVIEW Advanced Application Development class from National Instruments and the System Design with GOOP course from V I Engineering and Endevo. Successful designs and architectures should be documented and made available to developers for assessment for their use in new projects and for very specific training purposes in the organization. Reviews of proposed designs with senior resources help maintain high quality expectations and provide training opportunities for more junior developers.



A large part of the TS process area is the decision to make, buy or reuse components. If reuse is defined as a primary objective in the organizational process, then the TS process area also needs to have guidelines on what requirements are needed for a piece of made software to be reused. An example would be the level of documentation required, naming conventions, and style of the block diagram. In order to successfully meet the TS process area, it is important to have a defined style guide for reusable and non reusable code and a reuse library.

Both VISTA and National Instruments offer LabVIEW Style Guides, which provide a good starting point for development. Further, the VISTA Reuse Library consists of basic functions that may be applied to future applications with little effort. Finally, the VISTA Documentation Tracker and Help/Search Generator allow users to quickly document VIs and then search by keyword through the descriptions of thousands of VIs at a time to find possible code for reuse.

Verification

Verification and Validation are often confused with each other due to the similarities. CMMI defines Verification as “you built it right,” whereas Validation is “you built the right thing.” Verification is also closely linked to the Technical Solution process area. After creating a component, in LabVIEW’s case a VI, it must then be verified.

The verification process area consists of three goals. The first is to prepare for verification, which may be accomplished through a verification plan that details the components and procedures. The VISTA Documentation Template Set features sample plans for both verification and validation to assist organizations. The next goal is to perform peer reviews. To increase the effectiveness of peer reviews, it is important to have well documented and easy to follow code. The VI Analyzer Tool from National Instruments also complements peer reviews to look for specific issues. The last goal is to verify selected components, which requires a comparison of the current components to the requirements. The VISTA Configuration Management Tools make linking the CM process area to Verification easy by establishing appropriate labels to versions of each component and establishing new baselines.

Validation

After the individual components are developed and verified the components need to be integrated together into a final product. This process is defined in the Product Integration process area. Following the integration process the resulting product must be validated according to the process defined in the Validation process area. It is important to note that both Verification and Validation may occur at different stages of completion to ensure the components and products “were built correctly” and “are the correct items.”

The validation process area consists of two goals. The first is to prepare for validation, which may be accomplished through a validation plan that details the products, environment, and procedures. The VISTA Documentation Template Set features sample plans for both verification and validation to assist organizations. The second goal is to validate selected products and components, which requires a comparison of the expected performance to actual. The VISTA Configuration Management tools make linking the CM process area to Validation easy by establishing appropriate labels to versions of each component of the product and establishing a new validated baseline.

Risk Management



The Risk Management process area defines the process for identifying, analyzing and addressing risks. Risk management lasts the entire duration of the development lifecycle and may include risks ranging from technical to organizational. An example of a technical risk is the complexity of the code, as more complex code will result in the risk of greater maintenance, verification, and validation time. Therefore the risk management strategy may state that a piece of software must be rewritten if it fails to meet a defined maximum complexity. For LabVIEW applications the VISTA Path Tracker and Metrics Tracker may be used together to calculate the complexity as well as assess the risk associated with modifying code.

Summary

Achieving CMMI Level 2, 3 or higher is an intensive process that requires the dedication of all involved. For LabVIEW groups, it may be difficult to understand the benefits, as most do not currently measure productivity, quality, and delivery. For these groups, the first step is to understand the true cost of development. Only by understanding the cost of development are organizations able to set realistic goals and track improvement.

By understanding process improvement and the LabVIEW community, VISTA is able to provide organizations with the resources needed to improve. Since organizations vary in their needs and maturity, VISTA offers solutions to address each customer's unique situation. The solutions often include a mix of consulting, training, and tools. To learn more about a solution for you, please visit <http://vista.viengineering.com>.