Using Six Sigma in a Computer System Validation Project in a GMP environment

Tania Ramos
Master in Manufacturing Competitiveness
José Morales Morales, Ph.D
Industrial Engineering Department
Polytechnic University of Puerto Rico

Abstract *— Computer Systems Validation is a process used to test, validate, and formally document that a regulated computer-based system does exactly what it is designed to do in a consistent and accurate manner that is secure, reliable, and traceable. This this is done under a very regulated environment. Biopharma industries use computerized systems like DeltaV™, an automation system that simplifies operational complexity. The DeltaV™ is an easy-to-use system that simplifies operational complexity and lowers project risk. Six Sigma is a business management strategy used by different industries to improve the quality of products or services produced by the business through the removal of defects and errors. The Six-Sigma methodology is to improve a process with a positive implication on quality of product or, such as this case, a service in the Information Systems Department, to reduce Computer System Validation delays in the IS Department to zero days in a GMP environment.*

Key terms *— computer system validation, defects, GMP environment, Six Sigma*

PROJECT DESCRIPTION

The Federal Drug Administration has a General Principles of Software Validation Guidance, which specifies that, based on the intended use and the safety risk associated with the software to be developed, the software developer should determine the specific approach, the combination of techniques to be used, and the level of effort to be applied [1].

All changes related to the system need to be validated withing the Information System/ Automation department, because this system (DeltaV™) is classified as a GXP system in Biopharma. Changes in the code will be validated in an off-line testing environment, and once the validation is approved, the Automation Engineers will download the code to the production environment. The validation process is part of the Software Development Life Cycle (SDLC). The SDLC involves several major steps that can happen in sequence. Planning, Analysis and Requirement, Design, Development, Testing, Integration, and Deployment are the Stages of the SDLC. Testing stage will involve the process to check whether the system meets the specifications and whether it fulfills its intended purpose. Validation testing can include functional testing, system testing, integration, and user acceptance testing. This is a process to decide whether the product meets the specific requirements.

A typical SDLC consist of the following:

- **Planning and requirement analysis:** Determining user requirements, Functional requirements. Requirement analysis is the most important and fundamental stage in SDLC. It is performed by an Automation Engineer with a Manufacturing Engineer's input. This information is then used to plan the basic project approach. Planning also involves identifying risks associated with the project.
- **Design:** Based on the user requirements, one design or more is proposed in the Design Specification document. This document is reviewed by all stakeholders, Automation, Manufacturing, Engineering, and Validation. The best design is selected.
- **Implementation:** The programming code is generated in this stage based on the Design Specification document. Automation Engineers must follow coding guidelines. In this stage,

internal testing should be implemented while coding.

- **Testing:** In this stage, Validation should occur. Testing activities should involve all possible scenarios based on user requirements and design. In this case, we will call them Specifications (SPECs). Validation is completed when testing reaches the required quality standards. Testing activities should be conducted in an off-line environment.
- **Deployment:** Once the code is tested and all validation activities are completed, it is time to formally release (download) in the production environment.
- **Maintenance:** Code is in production environment. Delays may occur during this phase. For instance, delays may occur due to insufficient information on the process, document approval delays, lack of resources engaged, priorities between departments, or system failure. Discrepancies—when differences between specifications and code (in this case) are found— may occur, which shows that something is wrong and must be explained, justified, and corrected. When a discrepancy occurs, an investigation should be started to identify the root cause analysis, corrective action, and, finally, deviation closure. The investigations and assessment are the responsibility of Automation Engineers and Validation Engineers. Deviations cause delays in the validation process. The Automation Department has due dates and sometimes production windows to implement the change.

In order to reduce Computer System Validation delays, all deviation created in a period will be investigated to analyze root causes to determine and implement a solution and to improve download to production within the Automation Department to production.

PROJECT OBJECTIVES

By conducting this research, the expected outcome is to reduce Computer System Validation delays in the Information System/Automation department to zero days by Q3 2022.

PROJECT CONTRIBUTION

This project will use Six-Sigma concepts to analyze and implement a solution to reduce Computer System Validation process delays in the Information System/Automation Department. This research will contribute to determine the root cause within deviation of delays and improve the download to production time.

LITERATURE REVIEW

Computer system validation was first developed in the 1960s, originated by the military, initially called independent verification and validation. Later on, this process was implemented in a large scale by SAIC for the Safeguard Anti-Ballistic Missile System in 1971. Today, software verification and validation are known as the Institute for Electrical and Electronics Engineers (IEEE) standard P1012 [2]. “This verification and validation standard is a process standard that addresses all system, software, and hardware life cycle processes including the Agreement, Organizational Project-Enabling, Project, Technical, Software Implementation, Software Support, and Software Reuse process groups. This standard is compatible with all life cycle models (e.g., system, software, and hardware); however, not all life cycle models use all of the processes listed in this standard. Validation processes determine whether the development products of a given activity conform to the requirements of that activity and whether the product satisfies its intended use and user needs. This determination may include the analysis, evaluation, review, inspection, assessment, and testing of products and processes” [2].

Computer System Validation has become indispensable in daily life and the importance of the validation has also increased throughout the years. Today’s proper testing is more than storing a safe password; it is, for instance, underlying in ensuring

airplanes, trains, and automobiles safely get where they need to be. “In a scientific setting where we use laboratory informatics software, CSV ensures that a computerized system fits the intended use and functions as designed. And any time that system is upgraded or customized, we validate the change to ensure that the new functionality has not impacted any existing functionality.” CSV has become important due to some fatal mistakes. The most notorious occurred in the mid-1980s: the Thereac-25 incident that involved a malfunction in a million-dollar radiation therapy machine built to give radiation treatment to cancer patients. The high-energy radiation machine was controlled by computer from a separate room to protect the operator from receiving radiation. In 1986, a patient went to a clinic for their usual treatment; the computer gave several error messages that the technician tried to respond to. Because the commands were changed in such a short period of time, the computer did not respond properly. The metal plate moved away showing the technician that it was in low energy electron beam mode. But the beam that actually came from the machine was a blast of 25 000 rads with 25 million electron volts, the maximum setting, which is more than 125 times the regular dose. The patient’s health quickly worsened, and he died four months later from complications of major radiation burns. After that, at least another five similar incidents occurred in those years. Some of the possible causes for failure of the Therac-25 were failure to properly assess the older software when using it for new machinery; error and warning messages that were not well-designed, not fixed or even understandable; frequently recurring problems; proper hardware to catch safety glitches should have been installed; the manufacturer would not believe the machine could fail; and lack of communication and organization among hospitals, the government, and the manufacturer. In 1987, the Health Protection Branch of the Canadian government, along with the United States Food and Drug Administration (FDA), announced that the Therac-25 was dangerous to use and its used was to be discontinued until permanent changes could be made. In many senses, a computer had killed several people. It is hard to believe that simple human error could lead to unnecessary loss of life. Therac-25 was the first to rely on software for its safety protocols.

METHODOLOGY

Six Sigma is a business management strategy used by many different industries in an effort to improve the quality of products or services produced by the business through the removal of defects and errors. The objective of Six Sigma is to improve a process with a positive implication on quality of product or, as in this case, a service in the Information System Department (IS). This philosophy includes planning and organizational learning to achieve a successful method application. Six Sigma techniques and tools are a common approach to continuous improvement in a business sector and often include a framework tool called DMAIC, which outlines a method for identifying and challenging sources of poor quality and inefficient processes, searching for opportunities for improvement.

DMAIC (figure 1) stands for *define*, *measure*, *analyze*, *improve*, and *control*, and each of these words represents consequent stages within the Six Sigma implementation roadmap [3]. In the Define phase, the goal is to understand the project and its purpose and scope, map the current process, detail customer expectations, and estimate timelines. This phase takes approximately 1 to 2 weeks, based on the project inputs. The Measure phase will establish baseline performance of the process, develop a data collection plan and collect data, validate the measurement system, and determine the process capability. This phase takes approximately 2 to 3 weeks. In the Analyze phase, data will be measured to identify the possible cause of the problem and the actual root cause by using brainstorming, the 5 Whys, and other tools. The Analyze phase takes approximately 1 to 2 weeks to allow the review process map to improve the efficiency listing the

probable root causes and identify important factors and inputs that can impact the output. During the Improve phase, feasible solutions for the found root cause will be established, the best solution will be selected, the solution will be tested, and the effectiveness of the solution will be assessed to ensure measurable improvements in the process. This phase will take approximately 1 to 2 weeks, based on the solution implementation plan. The last phase in this methodology is Control, where the solution will be validated and then a control plan will be implemented to ensure the new process is strictly implemented and the project can be formally closed.

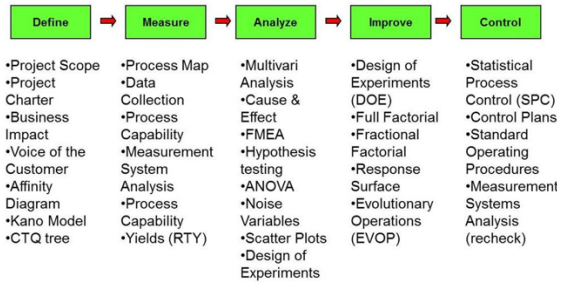

Figure 1: DMAIC

RESULTS AND DISCUSSION

This section provides the results on how the research objectives reduce the Computer System Validation delays in the IS Department to zero days in a GMP environment. The implementation of the Six Sigma strategy using DMAIC helped improve the process. All the DMAIC phases are summarized below.

Define Phase

During this phase, the customer(s) were identified and segmented according to their different needs and requirements.

- Data collected and displayed to better understand the customer(s) critical needs and requirements. Interview the customer to identify their needs and project expectations (Voice of the Customer [VOC]).
- Project charter to identify and understand the problem and its impact on the business.
- Establish the project team to identify and select cross-functional team consisting of Computer System Validation Engineer, QAV, Automation Engineers, Managers, and other team members.

Measure Phase

The Measure phase will establish baseline performance of the process, develop a data collection plan and collect data, validate the measurement system and determine the process capability. A process map (figure 2) outlines the current state of the Computer System Validation Process and help to determine any gaps or issues in the current operation.

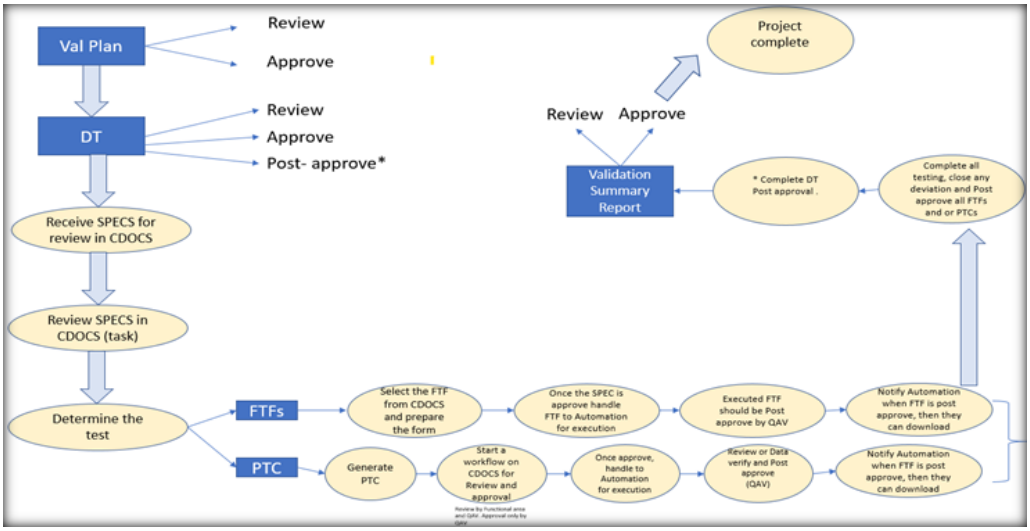

Figure 2: Process map

The process starts with the development of a Validation Plan that needs to be approved and reviewed stages. Development Testing is a testing shell that includes all testing related to the system (graphics, phases, parameters, and control module changes). Changes are documented in SPECs (requirements and designs); after the review and approval, the execution may start. The process is complete when testing and any deviations are post-approved. The final step of this process is complete when the Validation Summary Report—a document that summarizes all validation activities completed—is approved.

Analyze Phase

This phase consists of several steps that also require several tools. Value Stream Mapping (VSM) (figure 3) is a technique for visualizing the flow. It may use the data to map the process and pinpoint the locations where waste is occurring. It will allow the team to make better judgments about whether or not to add or remove a process step. Also, it creates visualization of delays in the CSV process and encourages continuous process improvement.

Figure 3 shows the flow between the different areas from Automation, Validation, and Computer System Validation Quality. It also includes the decisions contained in the process, considering the delay problems that the process team members will be analyzing in the process flow focused on deviations areas, which are the areas that may require re-execution, a factor that mostly affects downloads delays.

A Cause-and-Effect Matrix (table 1) was used as tool to understand quantitatively the relationship between the impact of inputs (process = Xs) on the outputs (findings needing to be resolved = Ys) of the CSV process.

Table 1: Cause-and-Effect Matrix

0 No relationship between the input and the outputs
1 Slight relationship between the input and the output
3 Average relationship between the input and the output
9 Direct relationship between the input and the output

	Voice of Customer (Rating of imp to customer)				
Output (y)	Download on time	No deviation	No re execution	Total Value	Total Value %
Priority (Output rating)	9	3	3		
Input (x)					
Validation Plan approval	1	0	0	9	2%
SPECs review and approval	9	0	0	81	21%
Testing assessment	1	0	0	9	2%
Execution	9	9	9	135	35%
Testing approvals	2	9	9	72	18%
Deviations assessment	3	9	3	63	16%
Validation Summary report	1	3	1	21	5%
				390	

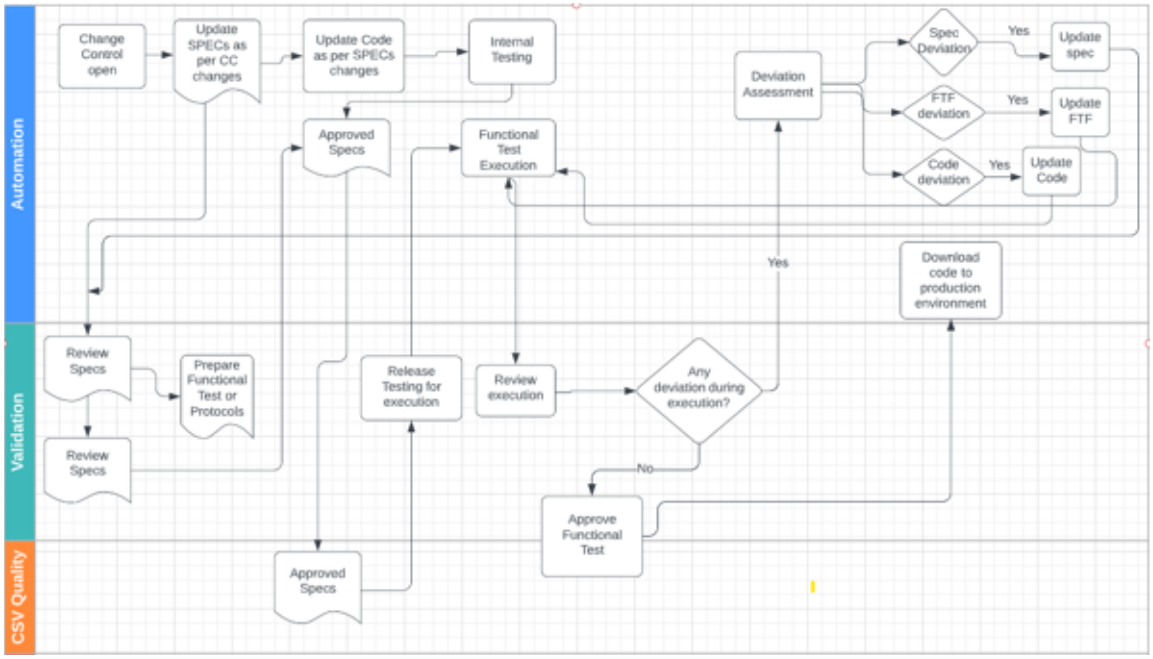

Figure 3: Value steam map

The pareto chart (figure 5) was another tool used to identify areas to focus on first in the process improvement based on the data in the cause-and-effect matrix obtained in the Measure phase. Pareto charts show the ordered frequency counts of values for the different levels of a categorical or nominal variable.

The execution cause covers 35% of the total correlation score (table 2). This is the most influential factor contributing to the delay in code downloading.

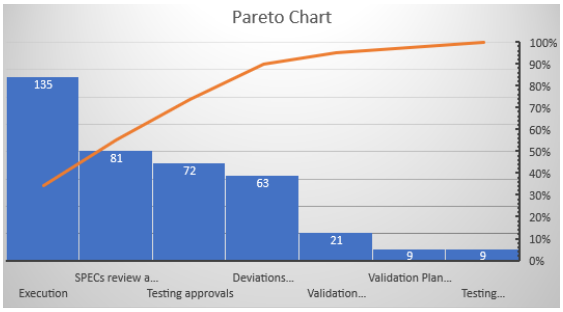

Figure 5: Pareto chart

Table 2: Pareto chart analysis

Cause	Correlation Score	Total Value
Validation Plan approval	9	2.0%
Testing assessment	9	2.0%
Validation Summary report	21	5.0%
Deviations assessment	63	16.0%
Testing approvals	72	18.0%
SPECs review and approval	81	21.0%
Execution	135	35.0%

With this information and analysis, the team members identified potential root causes of the delays in the code download (table 3).

Table 3: Potential root causes

Root Cause Analysis		
Area	**Effect**	**Root Cause**
Delays in Specs review and approval	Testing delay	Different priorities from approvers Communication Schedule not updated
Execution	Testing delay impact the code download window	Re testing as per specs fails Re testing as per code fails Re testing as per execution errors

The first area is the Delays in Specs review and approval; this cause had 21% of the correlation score causing a delay in testing, since the testing cannot start until the SPECs are approved. Possible root causes for that effect are that approvers may have other priorities, the communication was not effective, and sometimes schedules are not updated as required.

Execution has 35% of the total correlation score causing several delays effects, like delays in code download. A possible root cause for this effect is that, during the execution, some testing fails for reasons such as fails in the SPECs, fails in the code, or fails per executions error. These types of failures impact the execution, causing a re-testing and consuming time from the schedule.

Process data collection (table 4) collected all the data related to execution for two different Change Controls that involved changes in the Delta V Control System. Table 4 shows the different phases of the process, highlining the ones that delayed the expected results. For the First Change Control TR-529784, 60% of the executions have deviations and 67% need to be re-executed. Those results cause the effect on delays during the execution. The second Change control TR-536224 shows that 57% of the executions have deviations and 25% need to be re-executed; those results cause the effect on delays during the execution.

A variance test was used to determine the variability of the two different groups: Pre-Implementation and Post-Implementation (figure 6). The variance test will allow to compare groups variances, because the variance is a measure of the spread (variability) within a dataset. Figure 6 shows that the Standard deviation for Pre-Implementation is 0.996 and for Post Implementation, 0.0.389. The test shows that the p-value is 0.161 > 0.05. A p-value greater than 0.05 means that deviation from the null hypothesis is not statistically significant, and the null hypothesis is not rejected.

Table 4: Process data collection

Change Control	Description	SPECS affected	SPECS review (days)		Specs approval (days)		Execution								
							Functional Test Forms	Execution Status	Deviations	Deviation type	Re-execution	Days in Execution		Execution Total days	
			Expected	Actual	Expected	Actual						Expected	Actual	Expected	Actual
TR-529784	Remove Resin counter from purification Delta V graphic	SPEC-433727	3	6	2	1	FTF-CV-01	Completed	0	N/A	No	3	2	3	5
							FTF-CV-02	Completed	0	N/A	No	3	4		
							FTF-CV-03	Completed	1	Code	1	3	5		
							FTF-CV-04	Completed	0	N/A	No	3	4		
							FTF-CV-05	Completed	2	SPEC Code	1	3	5		
							Total 5		Total 3 60%		Total 2 67%				

Change Control	Description	SPECS affected	SPECS review (days)		Specs approval (days)		Execution								
							Functional Test Forms	Execution Status	Deviations	Deviation type	Re-execution	Days in Execution		Execution Total days	
			Expected	Actual	Expected	Actual						Expected	Actual	Expected	Actual
TR-536224	Chrome TOC update		3	6	2	1	FTF-CV-01	Completed	0	N/A	No	3	2	3	5
							FTF-CV-02	Completed	0	N/A	No	3	3		
							FTF-CV-03	Completed	0	N/A	No	3	3		
							FTF-CV-04	Completed	0	N/A	No	3	2		
							FTF-CV-05	Completed	1	Code	1	3	5		
							FTF-CV-06	Completed	3	SPEC	0	3	2		
							FTF-CV-07	Completed	0	N/A	No	3	3		
							Total 7		Total 4 57%		Total 1 25%				

Method

σ_1: standard deviation of Pre Implementation
σ_2: standard deviation of Post Implementation
Ratio: σ_1/σ_2

Descriptive Statistics

Variable	N	StDev	Variance	95% CI for σ
Pre Implementation	12	0.996	0.992	(0.455, 2.606)
Post Implementation	12	0.389	0.152	(0.158, 1.148)

Ratio of Standard Deviations

Estimated Ratio	95% CI for Ratio using Bonett	95% CI for Ratio using Levene
2.55930	(0.619, 13.125)	(*, *)

Test

Null hypothesis H_0: $\sigma_1 / \sigma_2 = 1$
Alternative hypothesis H_1: $\sigma_1 / \sigma_2 \neq 1$
Significance level $\alpha = 0.05$

Method	Test Statistic	DF1	DF2	P-Value
Bonett	1.97	1		0.161
Levene	1.82	1	22	0.191

Figure 6: Variance test analysis

Improve Phase

The Improve phase focuses on developing ideas on how to remove sources of variation in the computer system validation process, specifically the delays in the process. This phase deals with testing and standardizing potential solutions. The idea at this point is to understand what is really occurring in the process and not what is perceived to be the root cause(s) of any variation. Basically, it is used to recognize findings addressed by this project as potential modes, determine their effect on the code downloading delays, and establish actions to reduce those delays.

Based on the results from the Analyze phase and the potential root cause for the delays in the process, the team proposed these solutions for those causes.

- **Priorities**
  - Prioritize project between cross-functional teams identifying urgent and business values.
  - Extend meeting invitations to other stakeholders.
  - Include Project Managers in Change Control updated for Delta V system changes.

- **Re-executions (Re-testing)**
  - Re-testing as per Spec failure and Re-testing as per code failure. Revise Automation Standard operating procedures and Guides to improve their internal testing. Automation engineers should

review internal testing with the team leader and make sure to address all code and specs failure before the Validation process starts, so that the Validation can be completed with no deviations, avoiding the delays.

- **Execution errors**
  - Ensure all executers and Automation engineers are trained in Delta V system.
  - Ensure that all SOPs are up to date on their learning training list.
  - Include a witness during executions in which changes include phases, parameters, and graphics updates.
  - Validation Engineers should participate in the executions as verifiers.
  - Implement system audits using existing policies and Work Instructions.
  - Create a Deviation tracking tool in order to track deviations and for easy closure to avoid delays.

Implementing and testing some of the proposed solutions (table 5) addressed the effect of delaying the process and reduce them. Pos-Implementation Process data collection (table 6) reflects that, after the solution implementation, the number of deviations was reduced by 38% from 58% deviations before the implementation to 20% after the implementation. Even though the process had a deviation in the total executions days was on track. Standard deviations from table 4 improved after the implementation and comparing the hypothesis variance the Implemented process was statistically significant.

Table 5: Implemented solutions

Implemented Solutions	
Priorities	PM assigned, timelined monitored
RE-executions	Automation Eng. Include revisions in their internal testing
Executions	Validation eng included during the executions as a verifiers
Deviation Tracker	Validation Eng creates a Smarsheet including all the execution steps, tracking any delays. Smarsheet monitored by the PM

Table 6: Post-Implementation process data collection

Change Control	Description	SPECS affected	SPECS review (days)		Specs approval (days)		Execution								
							Functional Test Forms	Execution Status	Deviations	Deviation type	Re-execution	Days in Execution		Execution Total days	
			Expected	Actual	Expected	Actual						Expected	Actual	Expected	Actual
TR-400287	Remove Batch reports and logviewer applications from DeltaV Graphics	SPEC-410230	3	2	2	1	FTF-CV-01	Completed	0	N/A	No	3	2	3	3
							FTF-CV-02	Completed	0	N/A	No	3	3		
							FTF-CV-03	Completed	1	SPEC	No	3	3		
							FTF-CV-04	Completed	0	N/A	No	3	2		
							FTF-CV-05	Completed	0	N/A	No	3	3		
							Total 5		Total 1 20%		Total 0 0%				

Change Control	Description	SPECS affected	SPECS review (days)		Specs approval (days)		Execution								
							Functional Test Forms	Execution Status	Deviations	Deviation type	Re-execution	Days in Execution		Execution Total days	
			Expected	Actual	Expected	Actual						Expected	Actual	Expected	Actual
TR-532791	Harvest vessel modification	SPEC-490698	3	3	2	2	FTF-CV-01	Completed	0	N/A	No	3	3	3	3
							FTF-CV-02	Completed	1	SPEC	No	3	3		
							FTF-CV-03	Completed	0	N/A	No	3	3		
							FTF-CV-04	Completed	0	N/A	No	3	3		
							FTF-CV-05	Completed	0	N/A	No	3	3		
							Total 5		Total 0 0%		Total 0 0%				

Control Phase

During this phase, the implementation of the actual changes involved putting in place to measure and monitor the new established process. The main activity in this phase is the Improvement Plan (figure 7). At the end of this phase, Automation Engineers are responsible for ensuring the new process.

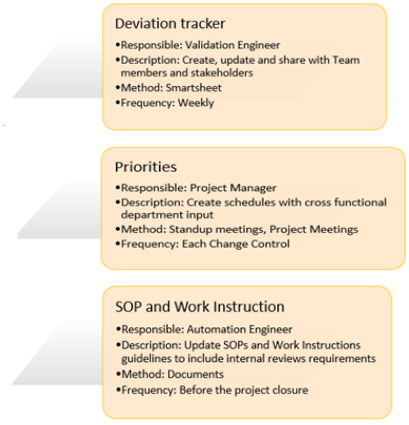

Figure 7: Improvement plan

CONCLUSIONS

The problem stated in this project has been solved. The Six Sigma methodology implemented with the DMAC tools represent consequent stages within Six Sigma implementation roadmap to reduce the delays in coding the download to production in the Information System/Automation Department.

Most Important Findings

Internal testing is key in the process. Taking time for testing before the actual validation made a difference. Adding resources to the process, such as including a verifier, demonstrate that deviations and time reduce significatively.

- **Tracking tools:** Robust tracking tools and their monitoring helped the Automation Department have visibility on the time.
- **Project manager:** Facilitates the project timeline to reduce any delays.

Limitations

Each Change Control is different and may require different validations. Some of them may include several specifications and more or less Functional Test Forms requiring different scheduling.

Summary of Contributions

The implementation of Six Sigma concepts contributes to analyzing and implementing solutions in a Computer System Validation Process within the Information System/Automation Department. This implementation determined that the delays in the specifications and executions were the root cause for the delays in the coding download to production. In addition, the new process, well implemented and maintained, successfully transitioned all responsibilities to the new owner.

Future Research

There are some gaps in our knowledge around Computer System Validation delays following our findings, and we would benefit from further research, including a realistic evaluation to extend and further test the theory we have developed here:

- **In-depth exploration of how Change control can influence the code downloading delays:** Research could explore the types depending on the system change, which may include parameters, graphics, recipes, formulas, or phases, and how these changes may impact coding.
- **Opportunity to explore the implementation of the solutions found in another system (DeltaV), Logmate, RT reports, or any other computerized system:** Research could explore which systems may adapt to implementing Sig Sigma in their process.

REFERENCES

[1] *General Principles of Software Validation*, FDA-1997-D-0029, U.S. Food and Drug Administration, Rockville, MD, 2002 [Online]. Available: https://www.fda.gov/media/73141/download

[2] *Standard for System, Software, and Hardware Verification and* Validation, IEEE P1012, May 14, 2018 [Online]. Available: https://standards.ieee.org/ieee/1012/7324/

[3] J. Dudovskiy, "Six Sigma," Business Research Methodology, March 18, 2014 [Online]. Available: https://research-methodology.net/six-sigma/