INTEGRATED TEST APPROACH

Facilities Development and Operations Contract (FDOC), Lockheed Martin

555 Gemini, Houston, TX. 77058

Will.l9.Cotton@lmco.com

John.Liechty@lmco.com

ABSTRACT

This paper describes a testing methodology undertaken on the Facilities Development and Operations Contract (FDOC) by Lockheed Martin. The methodology was defined with the intent of reducing project schedule time to enable NASA's Johnson Space Center (JSC) to be able to deliver the Mission Control Center (MCC) 21 project as quickly as possible. 21 represents the 21st century where NASA JSC is updating its control center with new technology and operational concepts in order to support NASA customers wanting to use control center assets to support space vehicle operations.

In collaboration with the NASA customer, a new test concept was conceived early during MCC21 project planning with the goal of reducing project delivery time. One enabler that could help reduce delivery time was testing. Within the project, testing was performed by two entities, software development responsible for subsystem testing and system test responsible for system integration testing. The MCC21 project took a deliberate review of testing to determine how it could be performed differently to realize an overall reduction in test time to support the goal of a more rapid project delivery.

1. INTRODUCTION

FDOC provides both development and sustaining activities for JSC MCC ground assets required to support International Space Stations (ISS) operations and support plan-train-fly user simulations and testing. System requirements for new software and hardware capabilities have typically been implemented via the waterfall approach. Attempting to reduce delivery cycle time was a focus for the MCC21 project. By reducing test cycle time, it was believed the NASA customer could deliver the system to the users for operations faster than it could using the serial testing approach that had been the norm for many previous development and sustaining activities.

Not having an existing or past parallel test approach model to follow, FDOC devised a test approach that took a more integrated approach between subsystem development and system level testing activities. Thus, the Integrated Test Approach (ITA) was conceived. The ITA has evolved from its original concept as testing progressed through months of informal and formal testing.

Figure 1 presents the typical waterfall approach for development activities.

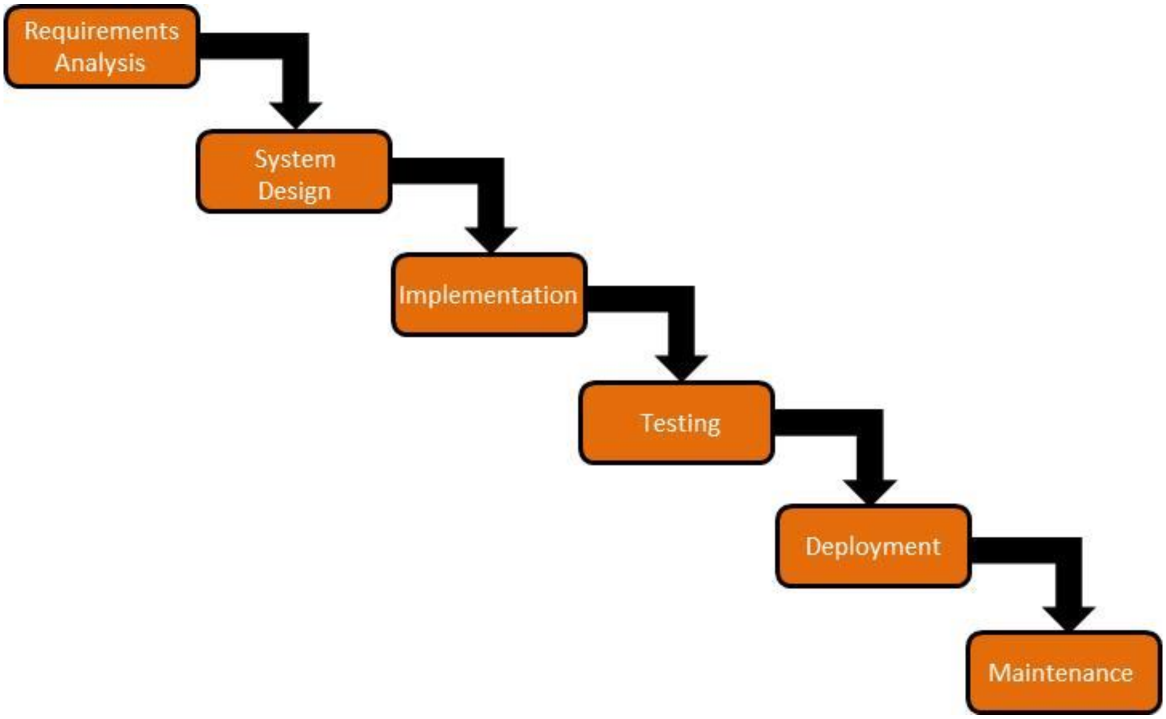

Figure 1 – Waterfall Approach

2. INTEGRATED TEST APPROACH

In the past, using the waterfall approach for development, serial subsystem and system level integrated testing was performed. This approach was used on previous contracts and was accepted as the business model. A downside of this testing approach is its serial nature that results in a longer test schedule compared to testing that might be performed in parallel.

The original testing phase of the waterfall approach was divided into four specific sub-phases: Subsystem testing, system level integration testing, one month duration to generate documentation and conduct a system acceptance review, followed by user validation testing. The total time allocated for subsystem and system level testing and preparation of documentation spanned eight months. Figure 2 depicts test and certification activities used by the waterfall approach.

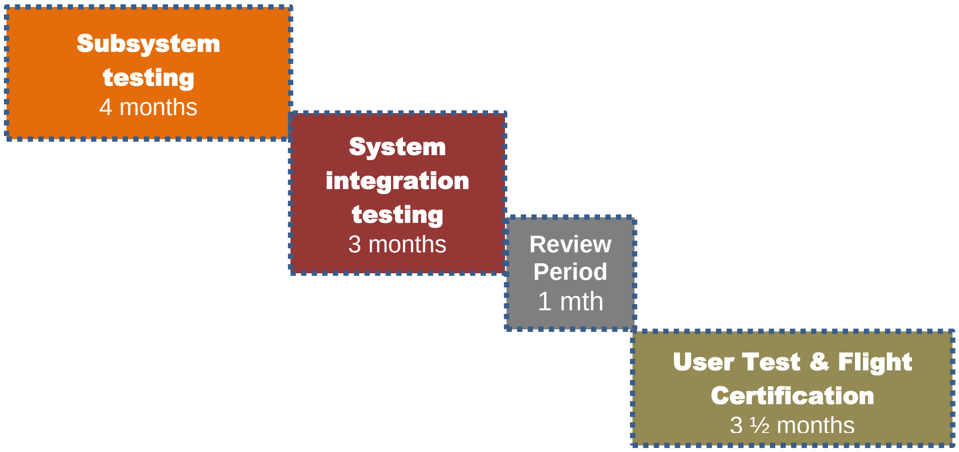

Figure 2 – Original Test Phase Linkage (46 weeks) using Waterfall Approach

The foundation for the ITA consisted of:

- System level testing performed early to flush out software bugs that could impact FDOC's ability to deliver on project requirements
- Implementation of the ITA would abandon sequential testing and attempt to perform subsystem, system, and user testing on software as soon as it was available.
- Conflicts in anomaly resolution fixes were determined by applying priorities set by the project, which were then assessed by an independent and diverse defect working group
- Each Level B requirement was to be tested once and only once if at all possible to minimize redundant testing
- The system test organization was to accept additional requirements and to test them on behalf of the software development organization so that a focus on developing software could be maintained
- System level testing was to be performed simultaneously by system test (requirement verification) and the user community (validation of operational scenario) to minimize schedule

3. ITA EVOLUTION

With the MCC21 project, the belief was that subsystem, system level, and user testing could be performed in parallel to reduce the test cycle and contribute to a reduced delivery schedule. A key element for success was an integrated approach to link subsystem and system level testing together where possible. As noted in the introduction, the ITA was championed to reduce test cycle time and contribute to an overall reduced project delivery cycle.

The initial ITA cycle accelerated system level testing to occur almost in parallel with subsystem system testing. There was a one month delay in starting system testing.

Figure 3 depicts the acceleration of system level testing to start one month after the start of subsystem testing. The one month delay was intended to facilitate the discovery and resolution of any anomalies affecting the platform and its associated services. This would also allow software development to complete some of its subsystem testing so that system integration testing would begin on a stable platform with essential services providing basic capabilities.

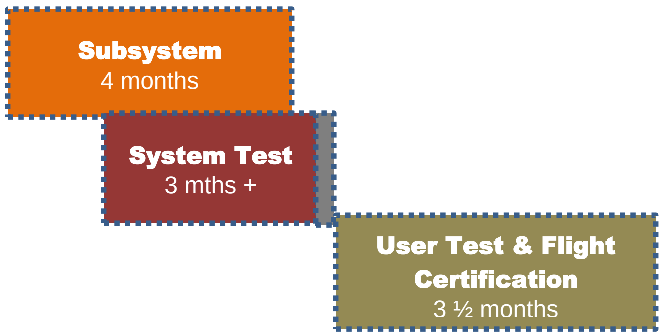

Figure 3 – Initial Integrated Test Approach (13 week reduction)

The newly developed ITA test phase reduction was seen as a significant schedule improvement that provided acceptable risk to the project. Integrated testing would allow subsystem testing to start as originally planned, followed in three weeks with system testing. After subsystem testing completed, system test would have two weeks for an overall regression test. Lastly, the previous one month gap for documentation and acceptance review was reduced to one week. Overall, this resulted in a 13 week reduction for the entire test phase.

Since there was a need for software development to maximize software generation, while minimizing testing support, the ITA team began looking at the Requirements Traceability / Verification Matrix (RTVM) that each organization maintained. As part of its process, the system test organization verified level A requirements by successfully executing the derived level B requirements associated with it. The ITA noted that subsystem teams were also verifying some of the same level B requirements as elements of their unit and regression testing. The only difference in verification for many of the same level B requirements was whether it was an application/unit or system level emphasized approach.

The ITA team recommended to the Project that verification of level B requirements should only be tested by one test team, not both teams. Thus, the ITA team consolidated the software development and system test matrices into one master ITA matrix. The intent was to minimize redundant testing so each item was verified thoroughly, but only once. This meant that system test had to add additional steps to make its testing more subsystem focused while still verifying the system at large. Alternately, software development had to look at operational system-wide use in order to enhance its unit level focused verification. Thus, a comprehensive RTVM was created and maintained by the system test organization with ITA team oversight. The mantra of "*Test it once, but test it right*" was emphasized throughout the requirements reallocation sub-phase.

This approach, with system test responsible for the RTVM and its updates, allowed the software development team to concentrate on developing software by limiting its required testing. Software test results were then provided to system test via an Excel spreadsheet which could be read into the DOORS, the contract's requirements management tool, via a set of ITA developed scripts.

In order to ensure all requirements were being verified, an initial RTVM was generated and the missing requirements were identified and allocated. FDOC then presented the concept of minimized redundant testing to the customer along with the preliminary RTVM. Initially concerned about risks in utilizing this approach, one of the customer managers spent extra time investigating the concept and the mechanics behind the consolidated RTVM. After this in-depth review of the matrix and the enhanced schedule maximizing ITA methodology, the customer accepted the new approach.

In a follow-up project schedule review, it was determined that user acceptance testing (UAT) could be combined with the already consolidated requirements verification testing. UAT is the end-user activity that validates that the software performs to the current flight operational scenario. The goal was not only to reduce schedule, but also allow user input and participation in early platform and software configurations. Certification of software to support flight operations would officially commence once UAT had been completed for specific software items. This would allow some applications to complete the UAT and certification process ahead of others. The shelf-life requirement for software varies, but all flight-critical software requires the longest period. Thus, it was envisioned that the certification time would extend past the ITA, but total test phase time would still be less than originally planned by approximately 6 weeks. This final configuration started with subsystem testing followed within 2 weeks by system test and user testing for an estimated total of 7 months (28 weeks).

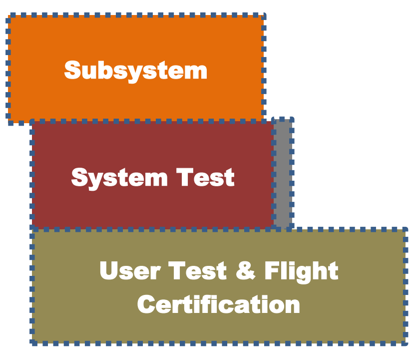

Figure 4 – Final Integrated Test Approach (28 weeks)

The ITA team realized that the risk of consolidating subsystem and system testing could be internally monitored by FDOC so that mitigation steps could be taken when necessary. However, with the introduction of user testing, additional risk was being incorporated that would make it more difficult for FDOC to control. As mitigation, monthly assessments of test plan progress were held with the NASA customer by FDOC with user test representatives. The project also reviewed relevant test information in a critical path project meeting with the customer each week. This allowed test progress and issues to be reviewed and evaluated for risk impacts as they developed.

Project priority of anomaly resolutions became an increasingly important element in minimizing schedule impacts. Initially, with the two test groups documenting platform (OS), hardware, and software anomalies, it was obvious that conflicts existed between developers and system test problem resolution goals. Then, with the addition of simultaneous user testing, many different viewpoints collided regarding discrepancy resolution urgency and importance. The project defined anomaly resolution priority based on project milestone achievements. The actual order was ascertained by the project and presented to the customer for concurrence. The concept of resolution priority was a strong point for the ITA and helped to keep the development team from collapsing under the weight of competing test team demands (developer, system, and user) as well as customer and FDOC management concerns about specific issues. Once an anomaly was prioritized, it would be placed into a group of similar discrepancies which could then be scheduled as a potential baseline of anomaly fixes.

Priorities included:

- Subsystem test completion
    - Platform (OS)
    - Hardware
    - Software
- System Test
    - Test Readiness Review (start of formal testing)
    - System Test completion
- User Test
    - Operational Test Reviews
    - Simulation (emulation of flight operations) Tests
- System Acceptance Review (contractual acceptance)
- Operational Review and Acceptance

As part of this prioritization effort, the project created, with customer concurrence, a Defect Working Group (DWG) of cognizant lead developers, testers, system engineers, hardware and platform engineers, and user representatives. All critical anomalies were directed to this working group, which was given the independence to assign priority based on the milestone criteria applicable to the issue. While one or two teams might have a vested interest in a particular discrepancy resolution, the team approach reduced this effect such that the DWG was considered fair in its assignments. Occasionally the customer or project would need a specific anomaly worked at a different priority than originally assigned, but this was very infrequent. The users were originally resilient to this concept, but later accepted it when they saw the end result where more of their critical issues were resolved over the lower assessed ones.

4. ITA APPLICABILITY

The MCC21 project's use of the ITA presented an opportunity for FDOC, the NASA customer, and users to collaborate on a testing approach to result in schedule reduction. As a result, the approach evolved from its original intent to a method that has supported the project through months of integrated subsystem, system level, and user testing.

Based on FDOC's experience with the ITA, the ITA supports parallel testing where a project challenge may be schedule constrained. As noted in the following section, benefits and drawbacks, the ITA can be tailored to fit the needs of the project. FDOC's strong customer relationship and insight to user operability enabled the ITA to evolve from its original testing concept to expanding requirements testing and creation of a collaborative working group to prioritize and manage anomalies.

5. ITA BENEFITS AND DRAWBACKS

Implementing the ITA has been an evolutionary process that has provided project benefits as well as drawbacks. Noted below are the more significant benefits that have been realized and drawbacks from implementation of the ITA.

BENEFITS

- If software is in a state that it can be integrated, system level testing can be started to potentially reduce schedule. System level testing requires enough functional components to focus on interaction between the major subsystems. If you have single elements which can't communicate with each other, then system verification is not possible.
- Early platform system level testing provides the opportunity to discover and resolve discrepancies before planned software releases, which decreases downtime for developer verification testing. Several major platform anomalies were resolved very early in testing. This ensured the platform was ready when the User application software was released. Where early platform verification was not available, such issues delayed software installation and testing.
- Early software system level testing can identify software bugs that can be fixed early in the life cycle and well before delivery. Early testing and detection of problems supports anomaly resolution earlier in the process so that blocked testing can then be completed.
- For projects where it is critical to release the software for user testing, the ITA supports incremental software releases to meet user test requirements. Small

user groups for early testing and then larger teams once software releases become significant and repeatable.
- Early user testing, performed with the right mix of Users (see drawbacks section,) provides early discovery of issues as well as detection of missing operationally needed functionality. This allowed the project to prioritize and schedule required fixes to meet test milestones, while deferring User desirable (instead of mission required) issues to the post-project sustaining phase.

DRAWBACKS

- The early introduction of user testing can be detrimental to the overall schedule and development resources. The customer often asked for development support to host special user test events. This happened when inadequate system capabilities existed, but a desire to get users on the system prevailed.
- The introduction of users early in the process led to some experiencing burnout and discouragement about the stability of the early system. Those users who had more testing experience realized that executing procedures on an immature system can lead to unexpected and unrepeatable results. Also, as platform services and system capabilities are upgraded and anomalies fixed, software may work very differently after significant baseline releases. This led to extra explanations and written documentation on what was being released, which would not have been necessary if user testing had been delayed until the platform was more user friendly.
- The focus and inherent testing philosophy for the software development and system test organizations is different. It can be overcome with diligence, but it takes more time than originally envisioned. With the MCC21 project, the customer learned that applications that function very well independently can fail or provide unexpected results when integrated with other system components and subsystems.
- Having the system test organization be responsible for the consolidated RTVM took more time than originally envisioned or later planned. The RTVM updates require diligence and perseverance to get a quality end-product.
- An integrated approach is very risky when the testers cross contractual boundaries and there is no prime to provide direction for conflicts and risk mitigation.

6. CONCLUSION

The ITA concept was implemented with the goal of reducing schedule. Due to project delays brought about by factors not related to testing, the ITA has resulted in more efficient testing where an emphasis was placed on software resources focusing almost exclusively on software development to meet an aggressive project schedule. Collaboration with the customer and users proved to be valuable in accelerating user testing and ultimately supporting a project goal to deliver as quickly as possible.