Case Study: QA Strategy Consulting and Testing Services for a Global CRM Development Program

1. BACKGROUND

The client, the world's leading market intelligence and technology provider to the financial industry, was implementing a global unified platform for the customer administration process. The platform was based on Siebel CRM and a set of integrated external applications. The key customers of the platform included Sales, Customer Order Management and Customer Support business streams.

Creating a new platform that would meet the expectations of multiple groups of business users in a large global enterprise is a uniquely difficult task (according to Oracle's own data, this system is one of the most complex Siebel customizations in the world). Soon after the program was launched, many delivery-related challenges inevitably emerged, driven by the following underlying causes:

- ▶ Multiple teams within the delivery organization (solution delivery teams for Siebel, solution delivery teams for Legacy systems, business analysis team, business users community, etc.) worked on the respective parts of the program in silos, with insufficient cross-team communication and coordination

- ▶ Most of the architecture, requirements definition and development (customization) effort was being delivered by external vendors

- ▶ Projects were schedule and budget driven rather than quality driven (in most cases, quality was an afterthought)

- ▶ Most of the business analysts involved on projects were in fact external consultants only engaged in the short term, and the involvement of actual business users was very limited

- ▶ The organization lacked staff with cross-business stream knowledge, which meant that the business could not test the system end-to-end
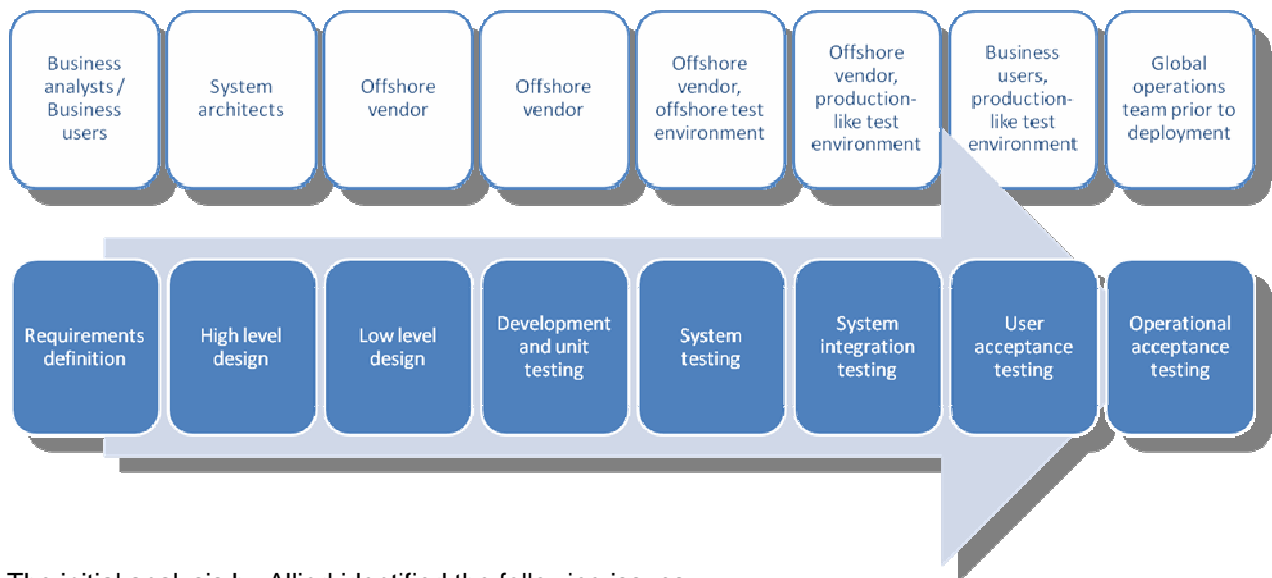
2. CHALLENGE

Allied was asked to come on board soon after the launch of the program to help analyze and improve the quality management process. The two main issues, as identified by the program management were:

▶ Insufficient quality by the time the product gets out of development

▶ Projects could not be delivered without significantly increasing business representatives' involvement in testing (UAT), which was not feasible under the circumstances

Allied specialists started by establishing communications with representatives of various business units and by studying the business processes involved in the customer administration process. In parallel, they performed a comprehensive assessment of the existing QA processes on various stages of the SDLC and analyzed the existing testing tools, procedures, documents and artifacts.

When Allied first joined the program, each project was delivered through the following stages:

Business analysts / Business users	System architects	Offshore vendor	Offshore vendor	Offshore vendor, offshore test environment	Offshore vendor, production-like test environment	Business users, production-like test environment	Global operations team prior to deployment
Requirements definition	High level design	Low level design	Development and unit testing	System testing	System integration testing	User acceptance testing	Operational acceptance testing

The initial analysis by Allied identified the following issues:

▶ Testing scope for the ST and SIT phases was is not properly documented

▶ Outcomes of the ST and SIT phases (test results, logged and fixed defects, etc.) were not visible to the project teams

▶ Test teams did not participate in the review of requirements and high level design documentation

▶ The ST, SIT and UAT phases used different defect tracking methods, and none of them utilized a defect tracking tool (as a result, there was no unified defects database)

▶ Many necessary tests types were not executed prior to UAT due to lack of business knowledge on the part of offshore vendors

▶ No full regression testing scope was defined

▶ Each business stream used its own approach to UAT, and in many cases the processes were not documented or formalized

▶ Lack of business process end-to-end testing prior and during UAT

▶ Lack of test planning: QA and test plan documents existed, but in most cases were irrelevant to the testing executed in reality

3. QA STRATEGY

Allied Testing was asked to develop the end-to-end QA strategy for the program, and to help introduce a unified QA process. The overarching challenge was to design an approach that would be suitable to all the teams involved, including business user teams.

The QA strategy was developed in several iterations. Each of iteration involved interviewing key members of the program and a review with stakeholders; in total, over 100 people were consulted. The strategy was developed by onsite Allied consultants embedded into the program team and supported by a small offshore team involved in data processing. In addition, Allied specialists were also consulting on active projects.

The **key principles** of the QA strategy accepted by the client included the following objectives and activities:

1. Introduce a unified QA management system for test scripts development and storage, tests execution and defect management.

2. Create a dedicated testing team that will build up and accumulate the business processes knowledge. Involve this team in testing prior to the start of UAT, and hand over most of the existing UAT scope to that team

3. Introduce a unified defect management process

4. Introduce unified standards for test scripts development

5. Develop unified regression test library and establish library maintenance procedures

6. Implement transparent QA reporting and third-party monitoring for all test phases

7. Formalize and improve UAT process

8. Improve sign-off processes for all testing phases (to ensure that quality is sufficient by the end of each phase)

4. IMPLEMENTATION

After submitting the QA strategy to the client, Allied Testing took responsibility for implementing it, working with 3rd party vendors and in-house development and QA groups.

4.1. UNIFIED QA MANAGEMENT SYSTEM

The Allied team built the unified test management system on the basis of Mercury (HP) Test Director. Allied performed the initial system customization to fit the program's needs, and formed a dedicated Test Director (TD) support team, whose responsibilities included:

▶ Providing training on TD usage for the program team (especially relevant for business representatives without IT background)

▶ Performing ongoing system customization (to address project-specific issues)

▶ Maintaining test scripts and consistency of the defects databases

▶ Routine technical support (backup, upgrades, user administration, etc)

4.2. DEDICATED OFFSHORE TEAM WITH BUSINESS PROCESS KNOWLEDGE

Allied formed a dedicated offshore test team to support the client's QA activities. Understanding that the effectiveness of that team will be in direct proportion to the level of knowledge of the client's business processes that resides with the offshore staff, Allied established communications with all business streams involved in the program and initiated a comprehensive business knowledge transfer and training effort. Allied also established the business process specialist role in the offshore team, and those specialists acted as local subject matter experts, or "SMEs by proxy".

The creation of dedicated, cross-trained offshore team also mitigated the problem of lack of the end-to-end business process testing. Creating and executing a test set when something is done by one business stream and the results are processed by another requires excellent synchronization of testing work performed across all different business groups. That can be quite challenging, especially in test execution, since those groups are distributed all over the world, and in many cases dedicate little time or attach little priority to testing, compared to their day-to-day business responsibilities. With Allied's dedicated team, the test scripts often include steps related to multiple business streams in any sequence, as they all can be executed by a single tester. Allied's team is collocated, which means that the testing staff can easily share business knowledge they receive from SMEs with the rest of the team. Thus, the Allied team is capable of executing end-to-end tests without the need to sync its efforts with other groups.

4.3. INTRODUCE UNIFIED DEFECT MANAGEMENT PROCESS

Allied introduced a formal defect lifecycle and developed a set of rules for defect descriptions. Each defect logged by business was reviewed by Allied (most of them within 8 hours, critical – within 1 hour) to ensure the development team has enough information to fix it. When the description was insufficient, Allied would get in touch with a business representative and add necessary updates. Allied would also the defects to make sure they are only closed by the defect originator (to ensure that proper retesting was done). All defects pending retest from business were first retested by Allied to prevent the business from wasting time testing fixes that have not been done properly. Allied's experts reviewed the defects database on the ongoing basis to ensure that defect management rules were followed.

4.4. INTRODUCE UNIFIED TEST SCRIPTS DEVELOPMENT STANDARDS

Apart from making sure that all existing test scripts were stored in the central test management system (TD), Allied also developed and documented standards for test script development.

4.5. DEVELOP UNIFIED TEST LIBRARY AND ESTABLISH LIBRARY MAINTENANCE PROCEDURES

Allied collected all existing test scripts and built a full regression test library. Existing scripts were analyzed, all test cases were normalized to the same format, obsolete test cases were updated or removed, and a large number of new test cases were developed to close gaps in test coverage. The resultant test library included over 9,000 test cases.

After the completion of each project, the library was updated by Allied to add newly delivered functionality to the regression scope.

4.6. IMPLEMENT TRANSPARENT QA REPORTING AND THIRD-PARTY MONITORING FOR ALL TEST PHASES

Using a unified tool for QA management enabled test managers to produce relevant and detailed QA reports at any time. Allied built a specialized reporting tool that could extract custom data from Test Director's database and build detailed reports.

Allied test managers would deliver regular reports to the project team during any project and any test phase, so the risks of not meeting quality requirements by expected date could be identified early.

4.7. FORMALIZE AND IMPROVE UAT PROCESS

It is always challenging to involve business users in the testing process, and it is even more difficult to ensure that the agreed standards are followed. To bring improvement to this area, Allied did the following:

- ▶ All defects logged by the business were reviewed by Allied (see section 3 above). In many cases business representative would simply notify Allied of the defect, and a member of Allied team would log it on a business representative's behalf.

- ▶ Many UAT scripts were either developed on the basis of Allied test scripts, or simply created by the Allied team and reviewed and signed-off by the business afterwards

- ▶ Some business users preferred working with printed test scripts, rather than using Test Director. In those cases, Allied team would load business test results to TD themselves

Allied also introduced an additional test phase before going into UAT that was executed offshore and made a great impact on the projects' quality and schedules.

- ▶ Allied created a comprehensive test library that business representatives could reuse for their own tests. For some business streams, Allied specialists actually created the UAT scripts ourselves. Client's subject matter experts assisted Allied with consultancy and reviews, but the most labor intensive part – script writing – was performed by offshore resources.

- ▶ On most projects, 70% or more of the testing effort fell on regression. Allied took over the regression testing for most of the business streams on the program. For each new release, Allied would do full regression testing, while business representatives would only run some high level regression checks during UAT. The test scope and test results were very thoroughly documented, and the test library was developed by Allied in very close collaboration with the business, which made possible such high level of trust the client put in Allied's specialists.

- ▶ Most of the issues that would be found in the UAT phase under the old process, would now be found by Allied before going into UAT, and therefore would not require a fix during UAT. With between 10 and 20 test>fix iterations performed by Allied prior to UAT for major projects, a significant amount of business users' efforts was thus saved.

4.8. IMPROVE SIGN-OFF PROCESSES FOR ALL TESTING PHASES

Allied and the client agreed on the measurable entry and exit criteria for each phase. The criteria, the measurement approach, and the phase sign-off process were included in each project's test plan, which was created and tracked by Allied's dedicated test manager.

5. LESSONS LEARNED

1. Many quality related issues (defects not found or found late, release delays, etc) can be fixed by introducing formal rules (e.g. unified test library, standardized approach to test execution, defect management, reporting, test planning, etc.)

2. Understanding how the system is used by the business is essential to establishing a successful QA process for complex applications. It is not feasible to test everything in the system with all possible test data. Therefore, business knowledge is required to identify the appropriate areas of the systems, scenarios and data relevant to the end users in order to create a test library that is relevant and not redundant. It is also not always feasible to fix every single defect (especially with internal systems, some defects do not impact business processes and therefore do not impact end users). In our experience, 50% or more of logged valid do not necessarily need to be fixed defects prior to production. Therefore, prioritizing the defects correctly from the start can save significant amounts of development effort and impact project delivery schedule.

3. Properly prioritizing the regression library allows for running limited regression cycles on early stages of the projects when there are many defects in the system, and full regression cycles only when the quality is good enough for that. On the program in question, a full regression run could require over 50 person/weeks for some projects. Allied used several subsets of regression tests:

   a. Smoke test: Small subset of tests executed after each release from the development team. Takes from 30 minutes to 3 hours. Testing scope is aimed at identifying ~80% of possible critical issues

   b. Minimal regression: ~10 person/weeks (can be executed within 3 business days). Testing scope is aimed at identifying 100% of critical issues and ~70% of all issues requiring a fix

   c. Limited regression: ~30 person/weeks. Scope is aimed at identifying ~90% of all issues requiring a fix

   d. A major project may include more than 30 releases from development and more than 10 regression cycles, so a properly prioritized regression testing scope for each run saves a lot of time and effort.

4. Proving to the program management that automating the entire test library does not make sense can be challenging. Automation is only beneficial when applied to the right kinds of tests (the ones that do not change the system significantly and contain a lot of similar repeated actions). Otherwise, the testing team will spend more time and effort automating the tests and maintaining the test library than it actually saves, compared to manual testing. In this program, more than 60% of the library would not benefit from automation.