indium
Make Technology Work

Building A Migration Testing Strategy For Early Defect Detection

White Paper

Introduction

Organizations always explore many ways to control the strength of their existing system while attempting to improve their accessibility or while considering an application re-design. More often the result will be moving from a traditional system to a client/server or web based architecture. The major reason for the conversion is to make them available more broadly – this often includes the end users outside of the organization and the consumer market. In many cases, these users expect a single sign-on access, with some privacy as to the identity or characteristics of the user. At this juncture the migration testing takes up its role.

Generally, software migration requires enormous effort. The same is with software migration testing; the effort is enormous and distinct. The migration testing does not verify the function of the application system, but rather, the process that places that application system into a production status. The process is attempting to validate the following:

- Proper programs are placed into the production status.
- Needed data is properly prepared and available.
- Operating and user instructions are prepared and used.

An effective migration test cannot be performed until the results expected from the software migration have been identified. The results should be predetermined and then tests can be performed to validate the expected results. Most defects occur in the early stages of a project but most defects are also found in its later stage, this costs 10 to 100 times as much to fix a defect in the later phases of a project. Finding defects in the early stage of software development helps organizations significant savings to their bottom line performance and also reduces the impact on their brand. So there is a need to get hold of the defects before they gain momentum.

Migration Testing - The 3 W's

What?
- Mechanism to leverage the strength of the existing system while attempting to improve the accessibility or while considering an application's re-design

Why?
- Ensure Compatibility
- Ensure existing Functionality
- High possibility of large number of defects

When?
- » Traditional Process – Post-Migration Testing
- » New Process – Pre-Migration Testing

Types Of Migration

- Database Migration
- Application Migration
- OS Migration
- Server Migration
- Hybrid Migration

Database Migration

- Understanding Data
- Resource Scheduling
- Scoping the Requirements Accurately and On Time
- Data Quality Framework
- Data Validation vs. Data Testing Strategy

Example – Migrating from an older version of database to a higher version (Oracle 9i to 10G)

Testing Challenges
- Data in the source databases may get updated during the test
- Table level and field level mappings are frequently changed

Application Migration

- Application Consolidation
- Application Development
- Existing-Application Enhancement

Example – Migrating a system from ASP to ASP .Net technology.

Testing Challenges
- More time is spent in the requirements phase to identify the scope of the migrated application
- Collecting information about business flows/processes in absence of proper or complete documentation of legacy application

Application Migration

In information technology, migration is the process of moving from the use of one operating environment to another operating environment i.e... in most cases, is thought to be a better one. Migration can be small-scale, such as migrating a single system, or large-scale, involving many systems, new applications, or a redesigned network.

Example – Migrating to Linux platform from windows platform.

Testing Challenges
- Understanding user applications and settings
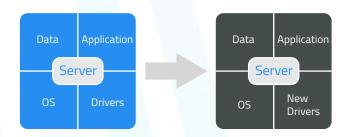- The process flow in the new OS may vary from the existing OS

Server Migration

The process of moving data from one storage pool to the next storage pool defined in the hierarchy

Example – Migrating from Windows Server to Mainframe Server.

Testing Challenges
- Understanding the migration rules
- Frequent data model changes

Server Migration

Traditional Migration Testing

Traditionally, migrations have been tested as a form of post-migration testing. Where in this strategy makes testing to start relatively late in the overall process, becomes labor intensive and causes many data-level errors. These limitations come into play particularly in highly regulated companies where the required margins of error are not feasible.

Post-Migration Testing
Post-Migration testing is performed once the migration is completed and the new environment is readily available for carrying out the testing process. Post-migration is typically performed in a test environment and includes:

Post-Migration Process

- Testing the throughput of the migration process (number of records per unit time)
- Comparing the Migrated Records to Records Generated by the Destination System.
- Summary Verification.
- Comparing Migrated Records with Source.
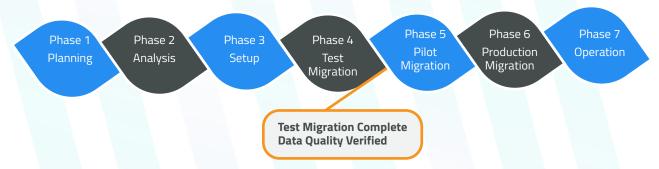- Migrated content has special considerations.

Additionally, an end-to-end testing can be carried out to ensure that maximum defects are addressed.

In the above approach, the sum of errors identified are very significant and with a much difficult effort at the later stage. So this necessitates looking forward to a new strategy.

A New Migration Testing Strategy

Pre-Migration Testing
The concept of pre-migration testing is not often covered during migration planning. The professionals involved in migration planning are not much aware of comprehensive pre-migration testing and the value it can add to a migration and particularly those migrations that are considered complex. Pre-migration testing takes place prior to the actual migration of any data, including test migrations.

| Phase 1 Planning | Phase 2 Analysis | Phase 3 Setup | Phase 4 Test Migration | Phase 5 Pilot Migration | Phase 6 Production Migration | Phase 7 Operation |

**Test Migration Complete
Data Quality Verified**
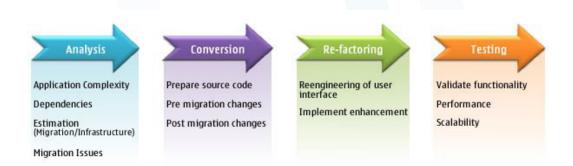
Pre-Migration Process

Pre-migration testing will assist with

- Defect Detection in Early Phase - Major Defects will be identified in the early phase of migration process. This at times will result in identifying incorrect requirements resulting in reduction of huge number of defects.
- Risk Reduction - We will be able to complete the migration ahead of the schedule. Defects identified/fixed in the earlier stages will result in high quality. Since all pre-measures have been taken care before the production migration, the actual down time is reduced. So risk with respect to Schedule, Quality and Downtime is reduced in Pre-Migration testing.
- Cost-Effectiveness - Since defects are identified in the initial stages, the cost required to fix those are less when compared to the defects identified in the later stages. Overall cost of the project can be reduced in the Pre-Migration testing.
- Less challenges during production migration.

Dawn of the New Migration Testing Strategy

The invested effort of implementing and testing migrated software is extremely high, mostly because the basic requirement is that the new system will provide the correct outputs comparing to the old one. This requirement is reasonable, but very durable to achieve, because usually there is no similarity between the new and the legacy systems. Moreover, even if the business flows are not changed, then achieving the same results is difficult because other components are different, such as infrastructure, operative environment, parameters, interfaces, etc.

The method of testing migrated software which described below need to be fitted according to the special conditions that exist in each project

The testing group will be divided to three active teams.
 Acceptance Test (AT) Team – The famous testing team which approves the quality of changes or/and regression tests into real live environment.

- Operational Team – This team includes all the people that will use the system on a daily basis in real live.
- Business Component Team – This team includes the business people from the legacy system.

The first two teams (AT and Operational) should work in parallel, but separately. Both will use the same repository of test cases, but with different indication of test team execution, meaning some of the test cases will be executed by both teams and others will run only by one of them. The execution will perform on different environments. AT will continue to work on their regular test environments, while Operational will execute the test cases using the real live configured machines.

The Business Component team will spend equal time with each one of the other teams.

Acceptance Test and Business Component

- AT team is responsible for designing, running and analysing all the business test case's scenarios. Though it looks like a regular task for AT's testers, it may require demanding effort. This will be the first time AT's testers are being exposed to the new system (migrated), their business background and knowledge of the new system is zero or close to zero. Business Component team will guide and direct the AT team and more important, it will verify and approve that the system results are accurate.
- In order to achieve the main system test role, it is mandatory to have many professional testers as possible. Professional tester must be well aware of the system and should also be talented, assertive, accurate and fluent.
- In order to approve the system quality, AT and Business teams will go over each result and approve the correctness of: rates, wording, error handling, values, buttons, menus, files etc. Operational and Business Component
- The purpose of Operational testing team is totally different from AT. As mentioned, the tests will be done on the intended real live machine which is configured with all the relevant parameters. The purpose is to approve connectivity, operability and infrastructure ability of the entire system.
- Since Business Component team is familiar with the general system behavior, they will work with the Operational team to approve elements like system performance, backups, recovery, daemons, down times etc. Stabilizing the entire system running is a time consuming issue and requires several testing intervals.

Different Phases of Migration Testing

Phase 1: Pre-Migration Planning
- Assess the feasibility of your migration with a pre-migration impact assessment
- Identify the key project resources and assign them
- Have a configuration management policy and software in place

Phase 2: Project Initiation
- Create a stakeholder communication plan and stakeholder register
- Create a high-level first-cut project plan
- Define the hardware and software requirements for the later phases

Phase 3: Landscape Analysis
- Create a comprehensive data dictionary
- Create a high-level source to target mapping specification
- Share the risk management process with the team and have them update the risk register
- Refine your project estimates

Phase 4: Solution Design
- Create an edge design specification
- Define your production hardware requirements
- Agree on the service level agreements for the migration

Phase 5: Build & Test
- Create a migration retreat policy
- Obtain sign-off for anticipated data quality levels in the target
- Create a gap-analysis process for measuring actual vs. current progress

Phase 6: Execute & Validate
- Keep an accurate log of SLA progress
- Independently validate the migration

Phase 7: Decommission & Monitor
- Complete your system retirement validation
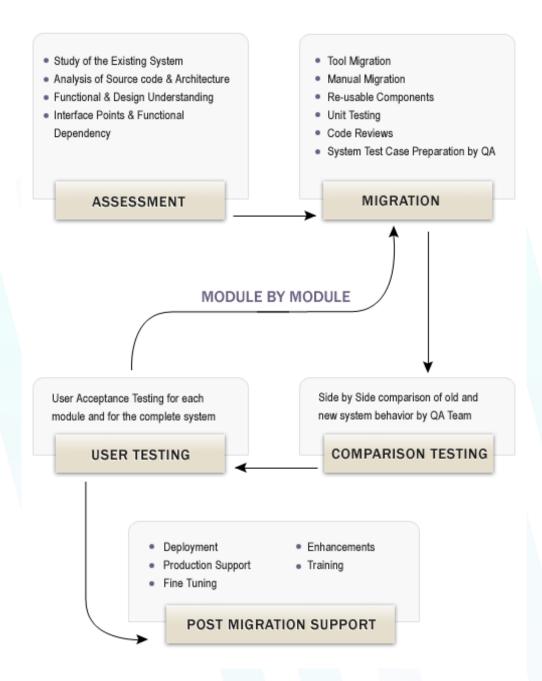- Hand over possession of the data quality monitoring environment

User Acceptance Testing
User acceptance testing provides an opportunity for the user community to interact with legacy data in the destination system prior to production release and most often, this is the first such opportunity for the users. Attention should be given to reporting, downstream feeds, and other system processes that depend on migrated data.

Production Migration
Though all of the testing is completed prior to the production migration, it does not guarantee that the production transition process will be smooth. Challenges seen at this point include procedural errors, and at times, production system configuration errors. If an automated testing tool has been used for post-migration testing of data and content, executing another testing run is straightforward and recommended. If an automated approach had not been used, some level of sampling or summary verification is still recommended.
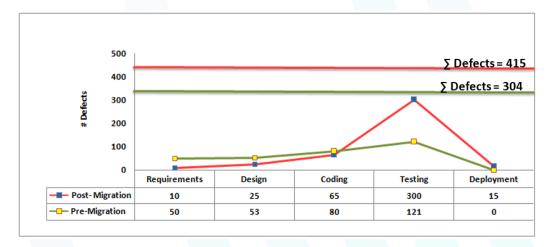
Migration Success Roadmap

ASSESSMENT
- Study of the Existing System
- Analysis of Source code & Architecture
- Functional & Design Understanding
- Interface Points & Functional Dependency

MIGRATION
- Tool Migration
- Manual Migration
- Re-usable Components
- Unit Testing
- Code Reviews
- System Test Case Preparation by QA

MODULE BY MODULE

USER TESTING
User Acceptance Testing for each module and for the complete system

COMPARISON TESTING
Side by Side comparison of old and new system behavior by QA Team

POST MIGRATION SUPPORT
- Deployment
- Production Support
- Fine Tuning
- Enhancements
- Training

Migration Success Roadmap

Analysis on ∑ Defects, Risk & Cost

1. Defect Detection in Early Phase

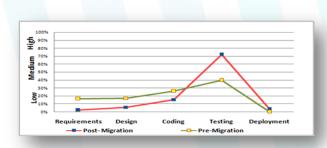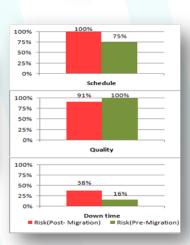|	Requirements	Design	Coding	Testing	Deployment
Post-Migration	10	25	65	300	15
Pre-Migration	50	53	80	121	0

∑ Defects = 415
∑ Defects = 304

Post/Pre-Migration Testing – Defect Comparison

- Reduction of defects using Pre-Migration method
- Early defect identification results in transparency of requirements
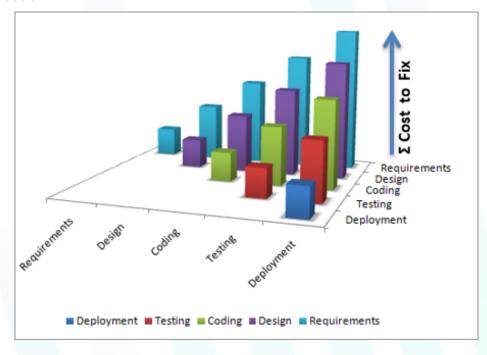- Ease in rework

2. Risk Reduction

Post/Pre-Migration Testing – Risk Level Comparison

- Early Completion of Development/Testing
- High in Quality
- Reduction of Downtime

3. Cost Reduction

Post/Pre-Migration Testing – Cost Level Comparison

- Early Defect leads to low cost.
- Earlier a defect is found/fixed, the cost is less.
- Overall Cost is reduced in Pre-Migration testing.

Conclusion

This new technique of early defect detection in migration testing fine-tunes the application and reduces the laborious effort at the last moment and it yields a good leverage in business point of view.

References

http://www.datamigrationpro.com/data-migration-articles/2009/11/30/how-to-implement-an-effective-data-migration-testing-strateg.html

http://www.2000technologies.com/files/Migration Project Testing V3 2TC.pdf

http://publib.boulder.ibm.com/infocenter/tpfhelp/current/index.jsp?topic=%2Fcom.ibm.ztpf-ztpfdf.doc_put.cur%2Fgtpm1%2Fm1thpln.html

http://en.wikipedia.org/wiki/Data_migration

INDIA

Chennai | Bengaluru | Mumbai
Toll-free: 1800-123-1191

USA

Cupertino | Princeton
Toll-free: +1 888 207 5969

UK

London

SINGAPORE

+65 9630 7959

Sales Inquiries
sales@indiumsoftware.com

General Inquiries
info@indiumsoftware.com