

White Paper



System Separation Approach

A Methodology for Planning Systems

Swanand Sahasrabuddhe

Abstract

This paper discusses various aspects of system separation, as applied to Planning Systems. Planning systems are part of satellite systems used in any enterprise and hence the approach for separating or cloning a planning system is derived from the approach used for primary transaction system. This paper discusses, in detail, one of the more commonly used approach viz. concurrent system separation, its impact on various facets of planning systems, synchronization issues, critical success factors for execution of system separation initiative and key differences in system separation for transaction systems and planning systems.

During the economic boom periods, it is often observed that mergers and acquisitions are used as primary means of achieving higher than industry growth by many companies. This was particularly observed world over during the 1980s and 1990s period. However, slowly, another approach started getting more acceptance as a means to create shareholder value and it was that of divestiture. Divestiture or de-merger, once seen as an indication of the failure of management to run conglomerates, is now viewed entirely differently. In fact, it is now seen as the first choice of approach, instead of the final resort to create shareholder wealth. This has resulted in higher divestiture activity world over, and has created billions of dollars of wealth for the shareholders. Divestitures can be executed in multiple ways, but the most commonly used are spin-offs and direct sell. These divestitures impact multiple facets of the business, including human resources, common business infrastructure, financial reporting and most importantly, the IT systems.

Spin offs or direct sell type of divestitures require IT systems to be separated and handed over to the new company. System separation is a complex initiative due to the reasons such as:

- System separation is usually initiated when a formal announcement of divestiture is not made. Hence, it becomes critical to keep this information confidential.
- It requires coordination between various groups, supporting hundreds of systems.
- Time lines to complete the separation are often very aggressive and driven by strategic decisions.
- System separation projects are not common and hence, there is a lack of standard processes or approaches on such initiatives.

This paper tries to put together an approach for separating planning systems pre-divestiture.

Separation Approach

The IT system separation approach should be divided into phases for better management. The entire exercise is divided into two distinct phase, viz. Pre-Divestiture and Post-Divestiture. The Pre-Divestiture phase is more critical, and is further divided into sub phases, viz. Analysis, Preparation, Execution, and Stabilization. While the Post-Divestiture phase constitutes activities related to Continued Support and often system merger with new company's systems or retirement activities. The discussion, in this paper, will be focused on Pre-Divestiture activities.

Differences between Approach for Transaction and Planning System Separation

There are inherent differences in the approach and tasks for separating a transaction system and a planning system. This is due to the difference in the nature and purpose of the applications. These differences are:

- The approach for planning system has to be derived from the approach for core transaction system. This is due to the fact that transaction system forms the backbone of any organizational system landscape, while planning systems are part of its satellite systems.
- The transaction system can be separated in three ways, viz. Logical Separation, System Copy and System build up. A planning system can follow two approaches, viz. Concurrent separation or non-concurrent separation within each of these three approaches.
- Consistency of transaction data in core systems becomes very important while it is refreshed in planning systems from transaction systems. Hence, in the former, the focus is on maintaining consistency of transaction data.
- Consistency of master data and configurations are important in both system separations, but less critical for planning systems compared to transaction systems. This is because transaction systems are the systems of record for all master data.

Analysis and Strategies for System Separation

Planning systems are the secondary systems used in any enterprise, and hence the approach for separating or cloning a planning system is derived from the approach used for primary transaction system. The core transaction system can be separated in three ways.

1. Logical Separation

This strategy would involve the following steps:

- Prepare new organization structure for divested business (Company codes, Sales Org, Purchase Org, Plants, etc.).
- Identify divested business related master data & extend to new organization structure.
- Separate divested business financially by performing financial postings between the two companies.
- Transfer transactional data - like open sales order, purchase orders, etc. - to new organization units.

2. System Copy

This strategy would involve the following steps:

- Prepare a new set of hardware and system landscape (Development + Quality + Stress + Production Environments)
- Restore Production backup on the new system landscape
- Financial separation of the businesses
- Archive the parent company master and transactional data in divested company system
- Archive divested company master and transactional data in parent company system

3. Building Divested Business Systems on New Hardware

This strategy would involve the following steps:

- Prepare a new set of hardware and system landscape (Development + Quality + Stress + Production Environments)
- Build System configuration on new system landscape
- Determine the divested business relevant master and transaction data for extraction from parent company production system
- Conversion of data to the new system landscape
- Data cleanup for divested business data in production system of parent company

There are many advantages in choosing the option of 'Logical System Separation' - such as less complexity, shorter lead times, less down times, etc. - but in cases where divestiture is being done through direct sell or spin off, this approach is not suitable. As in such cases, physical separation is required of the two systems so that the newly cloned system can be handed over to the new company formed. Hence, Physical Separation though Copy or Build-up is a preferred choice for most core transaction system separations.

The Planning Systems can be separated in two ways:

1. Concurrent Separation

The Planning systems that have an interface with the Transaction System are separated at the same time when Transaction System instance gets separated.

2. Non-Concurrent Separation

- a. Before Core Transaction System separation
- b. After Core Transaction System separation

The Planning systems having interface with Transaction System are separated either before or after the separation of Transaction System instance.

The selection of an appropriate strategy for planning systems requires detailed analysis and evaluation. The non-concurrent separation approach has advantages - such as less cutover time, due to staged separation and lesser contingency planning. However, it has the following disadvantages

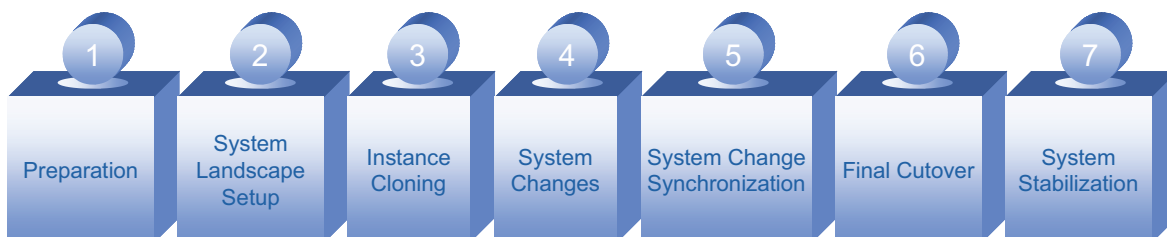
- a. It is not a feasible approach in case of Logical Separation or System build-up approach as the planning systems and transaction system should have same organization structures, locations, legal entities etc.
- b. In case of the System copy approach, it results into interface establishment between the new transaction system and the old planning system (or vice versa) for short durations which cause complexities and inconsistencies in the data.
- c. This also results in higher testing efforts

On the other hand, the concurrent separation has the following advantages:

- a. Concurrent separation is only possible in case of Logical Separation or System build-up approach, as the planning systems and transaction system should have the same organization structures, locations, legal entities, etc.
- b. In case of the System copy approach, there is no need to establish an interface between the new transaction system and the old planning system (or vice versa) for short durations.
- c. Lesser testing efforts

Thus, in whatever way the system separation for core transaction system is done, it is always advisable to use concurrent system separation for planning systems.

The following sections will discuss in details the concurrent system separation approach. Once the approach for core and satellite systems - like planning systems - is decided, the actual execution of the separation is carried of in seven stages:



1. Preparation

Preparation for an IT system separation project involves establishing common project planning processes, defining clear objectives, developing project plans, defining the scope and requirements of the project, and resourcing and staffing the project. Due to confidential and critical nature of the project, apart from normal project preparation activities, attention must also be given to the following things.

Approach selection should be done only after proper consideration of the pros and cons of each approach. Use most skillful and experienced resources to staff this project from the organization. This is important as there are so many intricacies and nuances of the IT systems that are known or that can be handled by experienced and skilled resources better and faster than inexperienced resources. Also, as far as possible, this type of work should not be out-sourced to new third party vendors.

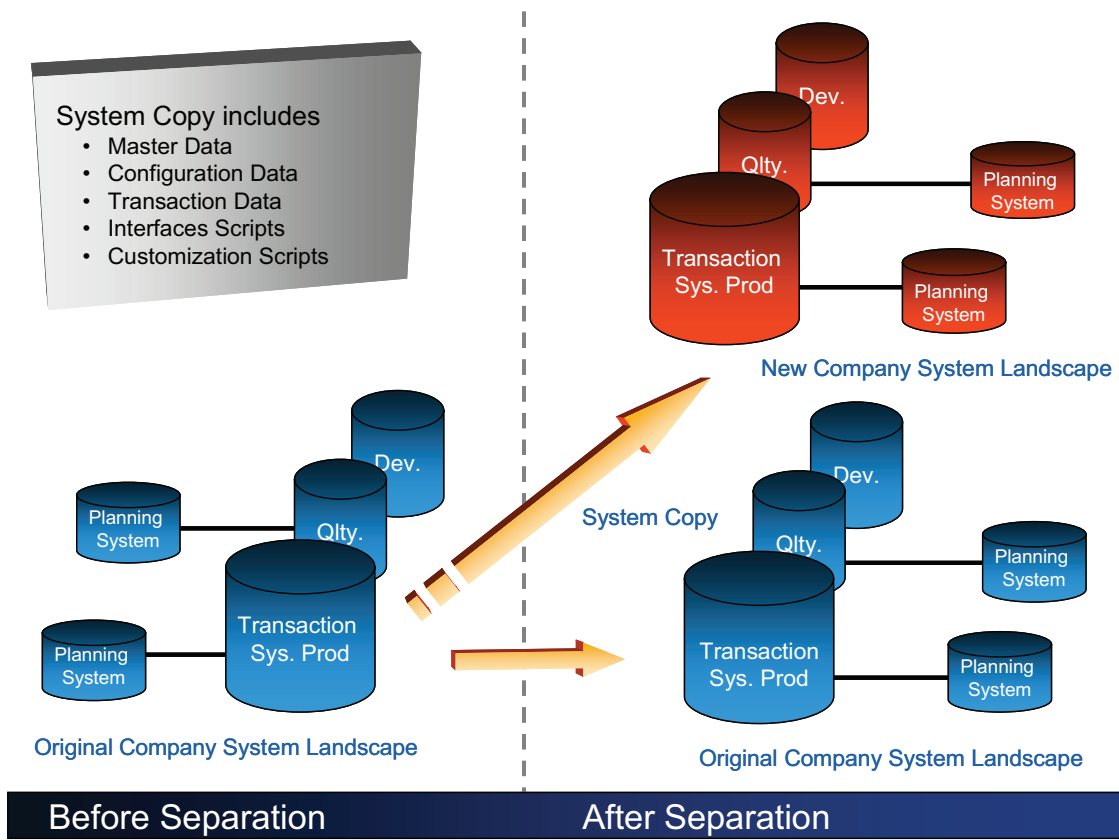
2. System Landscape Setup

Once the basic project processes have been set up, the next step is to set up a new system landscape. This involves purchasing similar hardware and software licenses, commissioning the hardware, installing the appropriate software, establishing network connections, etc. This activity should be done in parallel for core transaction and planning systems.

Usually, original landscape includes multiple environments - like development, quality, production, and stress for core transaction system, as well as for most of the satellite systems. Planning System, which is part of satellite systems, will also have environments like development, quality, production, and stress. Quality and production environments will be integrated with the transaction system's quality and production environment through various interfaces. Hence, setting up of new system landscape will include setup of all environments for planning system, along with core transaction system.

3. Instance cloning

Once the system landscape is set up for the new company, the next task is to clone the entire set of instances. This includes copy of four basic entities



1. **Database Instance:** This includes master data, transaction data and configuration data from respective parent system instance.
2. **Interface Scripts:** Planning systems get the transaction and master data from core transaction system and various other legacy systems through a set of interfaces. These interface scripts or middleware need to be cloned too. Usually, when the interfacing is done through flat file integration, the scripts are copied over to the new system and changes to the system identifiers are made in the scripts.
3. **Customizations:** Planning systems invariably have many customizations to address to peculiar requirements of the user companies. These should be copied over to the new system to maintain the functionalities in the new system.
4. **Job Scheduler:** Various actions - such as the running of interface jobs, processes, etc. - for a planning system are usually done using a Job Scheduler. Such job scheduler instance also needs to be set up and job definitions need to be copied over to new environment.

Once the basic setup is complete, a thorough testing of the environment needs to be carried out. This should include the testing of:

- Access to all system environments
- Access to application
- Interfaces with transaction and other legacy systems
- Core functionalities of the application
- Customizations
- Critical/ Key jobs runs.

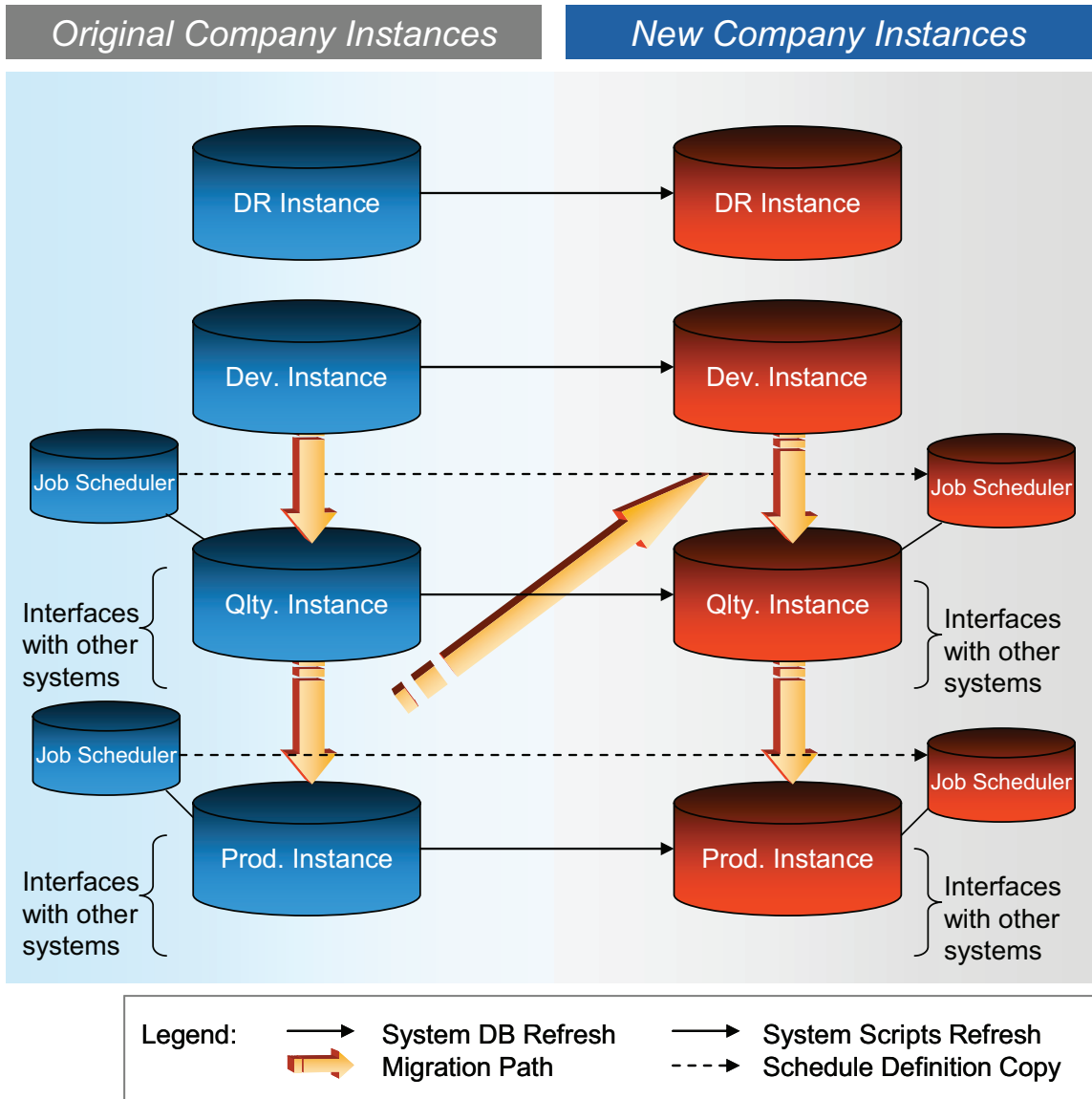
4. System Changes

After the cloned instance has been set up, the next step is to identify the changes required in the new systems as well as original system as a result of de-merger of the businesses. This, at a high level, includes changes related to master data, interfaces, customizations, and configurations. The following steps, of a normal Software Development Life Cycle, can be carried out to execute this.

- **Identification of Changes:** As mentioned earlier, the changes related to master data, configurations, interfaces, job definitions and customizations are identified for both parent and new system. Interface changes, normally, would constitute changes related to new system identifiers, new locations, removal of non-applicable locations, removal of non applicable interfaces, etc. Customization and Configuration changes will be minimal as those will be carried on to the new business systems. If there are any customizations or configurations specific to a business, these will have to be considered for making changes. Master data will have to be identified and mechanism should be developed to quickly identify the master data, mark it and delete it (or preserve it and delete the rest).

This is a very important activity during the cutover. For planning systems, the master data that will undergo changes will be primarily locations, items, product groups, transaction data related to the items, etc. Identification of such changes should involve Design owners of the application, business and project team. Job definitions will have to be changed as per the changes in interfaces and also taking into consideration the changes in server names, instance names, etc.

Environment Build-Up method for Planning Systems of De-merged Company



- Implementation of Changes:** Once the changes are identified and requirements are clear, the project team can work on Technical Design for each of the changes and implement them in quality and then production environments of the new system. This process should again follow the normal software development life cycle. The changes, before moving to production, should be unit tested in development and system tested in quality. At this point in time, no changes should be done in the original system as it is being used as production instance by the parent and the de-merged company. Although new system production system is called production environments, they are still not being used as production instance by the new company.
- Testing:** Comprehensive Testing needs to be carried out after the changes are applied as this whole exercise is very complex, and testing can prevent any defects going to production and reduce the rework later. Testing needs to be carried out in three stages – Unit testing after making the changes in development, System/ Integration testing in Quality, and final regression and User approval testing in Production. Once user acceptance test sign off is obtained, only then can production instance truly be considered for final cutover. In addition to normal testing of SDLC process, another testing is required for cut over simulation. This involves identifying and removing original company specific master data from the new system. Results, in terms of time taken, quality of the output, etc. should be in sync with final cutover activities. As indicated earlier, no changes are applied to the parent company systems at this point. Those are applied only after the final cutover is executed.

5. System Change Synchronization

The time elapsed between the system copy and completion of integration testing can be a few weeks to months. During this period, a number of changes can be applied to the original production system and some may be in a process of development, quality testing or ready for promote to production. These changes will be missed in the new system, if not synchronized with the original system. Hence, the project team needs to get these changes from the production support team of the original system, apply them to new system; unit and system test them and put them in production environment of the new system. Also, it should be made sure that any changes going into original production system after this, are also applied to new systems production through proper migration path. To achieve this, a periodic – not less than once per week – review of changes that the Production Support Team is working on, should be done with project team. Either the same versions of code, being migrated to Production of original system, should be used to apply to Quality environment of the new system for system testing. But, this has certain disadvantages - such as, all changes have to be redone in the new system related to environment paths, hard coding of system specific entities such as system identifiers, locations etc. Hence, another approach can be used which does not require code changes to be done again but requires additional unit testing on new environment. In this approach, current production version of the code in new environment is taken for any code change required according to requirements of original system. Changes are made in the development environment of the new system and a normal migration path is followed on the new environment in sync with the migrations within the old environment. This reduces the complexity of code change, but increases testing effort, which is often favorable.

6. Final Preparation & Cutover

Final cutover activities will have to be carried out on both the systems in sync with cutover activities of core transaction system. The activities for the original system would be:

1. Stopping of interfaces that become redundant or removal of locations from the current interfaces
2. Cleaning up of new company related data
3. Data conversions, if required
4. Master data refresh from core transaction system
5. System access removal for new company users

The activities for the new system would be:

1. Copy of the latest master data and transaction data from original production system. This should be in sync with transaction system copy over
2. Cleaning up of original company related data before next Planning run in the new system
3. Data conversions, if required
4. Batch/ Planning run
5. Master data refresh from core transaction system
6. System access removal for parent company users

Since this cutover is more complex than any other normal project cutover and multiple systems have to be kept in sync during the cutover, it is advisable to avoid any other migrations and changes to any of the systems during this period. A migration blackout can be observed, starting a week or two before the cutover and continuing till a couple of weeks after cutover.

7. Stabilization

During the project stabilization phase, the project team will continue to support the new company systems till the project is declared stabilized. The objective of this stage is to make sure that the new support team gets the new set of applications in fully stable conditions. Some of the important activities that are handled during this phase are:

- Support the new systems and applications to stabilize the operations.
- Documentation of all the changes made to the current design.
- Identifying the new support structure and processes. There is a possibility that if the de-merged company is bought by some other company, then the new parent company's process would apply. Otherwise, the original processes can continue. Also, the new company needs to establish support teams for the new set of applications. Thus, identifying resources and hiring them (or a sourcing partner to support these applications) becomes important.
- Transfer of all relevant system documentation and knowledge base to the new company, taking into consideration the intellectual property rights issues.
- Documenting the 'delta' training.
- Completing the 'delta' training for the new support group.

Critical

Success Factors

Considering the strategic nature of system separation initiatives, it becomes necessary to identify critical success factors early on, and make sure that those are taken care of during the execution of such projects. Some of the critical success factors are:

- Technically skilled and experienced implementation resources.
- Effective Coordination with multiple organizations -such as System Administrators, Database Administrators, Core transaction systemseparation team, other Legacy System, business users, and design owners.
- Availability of all key personnel -from business user community, support groups and external vendor support staff -is critical to the project.
- Performing technical evaluation of separation strategies and assessment of the impact of each strategy on project timeline.
- Information to all the participating stakeholders of their contribution to the project, deliverables and the project timetable.
- Commitment from each group to the attainment of the project objectives and timelines.
- Synchronization of both environments throughout the project timeline.
- Decisions regarding details of divestiture should be made early on and should be provided to the project team. These include decisions regarding segregation of item masters, locations, changes in business processes related to internal material transfers, systems to be cloned, etc. This helps in keeping the ambiguity and complexity low throughout the execution.

A Last World

Planning systems are part of satellite systems that are used in any organization. The approach for separating or cloning a planning system is derived from the approach used for primary transaction system. While the transaction system can follow one of the approach of logical separation, system copy or system build up on new environment planning system can follow concurrent or non-concurrent separation, using any one of the above mentioned approaches. The system separation is carried out by tailoring usual software development life cycle and package implementation life cycle. Special care should be taken to identify the 'To-Be'system landscape very early on in the project. Such strategic initiatives should be staffed with experience resources. Considering the large scale impact of the program, coordination between the various project groups becomes critical to the success of the project.

About the Author

[Swanand Sahasrabudde](#) is a Senior Consultant at SCM practice in Infosys. His diverse supply chain roles include IT Consulting and project management. He has several years experience across SCM functions - such as Production, Planning and Manufacturing Systems in Petrochemicals, Auto components, Hi-Tech Manufacturing, and Retail Industry. He has played leadership roles in multiple implementations, production support, testing projects and consulting engagements in areas of Logistics and SCM, Global outsourcing and IT/ SCM product strategy. Swanand can be contacted at Swanand_S@infosys.com



For more information, contact askus@infosys.com

About Infosys

Many of the world's most successful organizations rely on Infosys to deliver measurable business value. Infosys provides business consulting, technology, engineering and outsourcing services to help clients in over 30 countries build tomorrow's enterprise.

For more information about Infosys (NASDAQ:INFY), visit www.infosys.com.