Solutions  
  
  
 The Number one solution that we should focus on is about dividing task, into different stages like how we solve most of the problems we encounter.   
  
 First, migration process should be categorize by the type of data from the most important and least important data. The reason for this is that the important data are the data that the system or organization required to still continue the process without interruption, when these important data such as financial records, key customer information, or inventory data was migrated first, the organization now had enough information to continue the management. Also, when we prioritize important data first, unexpected technical errors from the old system wouldn’t be a problem since important data is already transferred.  
  
 Visualizing possible obstacles and errors before Data Migration will help us plan and prepare for the worst technical scenario, such as data loss or data corruption, security vulnerabilities, user adoption and even actual system downtime. After, analyzing and visualizing possible errors, applying an actual back up for the data will be crucial in case this errors occur during data migration.  
  
 Testing Data Migration with the decoy data. Random data from the old system will act as a test data for data migration, attempting a data migration with different size of data, from small amount of decoy data to large amount to see if how the migration and the new system will react to the migration.  
   
 Comparing database columns from old system to a new one is crucial before data migration. It is because new system might use different fields from the old one. For example in address, old system handle address such as City – Town – Street and place it under once field or columns, while on the other system they use separate fields for each part of the address, one field for City, one for Town and one for the Street. Using this kind of method will help us to also plan and understand how to avoid future obstacles for data migration.   
  
 Most important of all, the existence of Middleware is necessary for data exchange because different system use different database structure and also use different format, the indifference can cause data confusion. Creating a Middleware that translate different data from different software is an efficient approach to data migration.  
  
  
 Lastly, We can assign a team or a person to manually and visually compare the data transfer from old system to a new system, checking each formats, fields and comparing old data to a new data to check from errors, data lost and bugs.   
  
 Tracking a progress from previous migration from analyzing the transfer of important data before less important data to creating a Middleware. Taking notes about the process of executing these steps and also storing it for future analysis is crucial for a company’s success on data migration.