### Slow Navigation issues:

### Case Assessment:

1. Identify the operation that is slow
2. Identify the page or tree where navigation is slow and make sure that the issue is reproducible (Identify the components affected)
3. Identify the platform
4. Identify if there are customizations
5. Collect data about the number of entries on a page (or under a node)
6. Get topology info – (load balancers, cluster of application servers, JVM used, proxy etc.)
7. Check if database statistics are run weekly or longer frequency
8. Is it only to specific folders or all folders – if specific, how large is the folder?
9. Has it just recently become slow or has it always been slow?
10. Have there been any upgrades to the system (Documentum or 3rd party products) recently?

### Info needed from Customer:

1. Collect complete System info about the Application server, Content Server and the database
2. Get environment variable settings for the Content server
3. Collect isolated DFC trace with stack depth=0

dfc.tracing.enable=true

dfc.tracing.verbose=true

dfc.tracing.max\_stack\_depth=0

dfc.tracing.include\_rpcs=true

dfc.tracing.mode=compact

dfc.tracing.include\_session\_id=true

1. Collect Charles or Fiddler trace for the operation
2. Get configurations of the affected component
3. Get implementation classes in case of customization
4. Collect network data – ping / trace route information between client and the application server, application server and the content server
5. If the database is Oracle, get a database AWR report covering the slow operations
6. If the database is SQL Server, ask the DBA to monitor database and server performance
7. Get database side logs (such as alert logs in case of Oracle)
8. Get Content server log
9. Get a list of indexes

* For Oracle:
* For SQL Server: 

1. Get load information on the page or the nodes (number of folders, number of documents)

### Analyses checklist (for Performance Engineering):

1. Analyze network info and look for high latency
2. Analyze the Content Server log and look if there are errors.
3. Analyze the database (alert) log for errors
4. Parse DFC trace to get histogram of RPC timings
   * Use strategy appropriate for the RPC (we need a checklist for each RPC)
5. Get costly DQL queries
6. If SQL Server, verify that DM\_LEFT\_OUTER\_JOIN\_FOR\_ACL=T setting is enabled at the Content server (and server restarted)
7. Analyze the queries:
   * Check structure of the query for tuning, if the query is a custom query. Query modifications are possible in case of custom queries
   * Get the SQL query (or queries) for each DQL query, get the SQL trace for each costly query (using SQL\_TRACE option of apply method), get its execution plan (with full statistical info)
   * Analyze the execution plan and suggest DB tuning:
     1. Indexes, optimizer mode, PGA/SGA recommendations, session cached cursor recommendation
   * Check for queries related to vdoc\_count and adoc\_count calculations (they can be eliminated by means of browsertree configurations)
   * Analyze the AWR report
     1. Look for instance efficiency, top events, segment stats, tablespace stats, Oracle inititalization parameters, DB load information and the advisories for buffer pool, shared pool. SGA and PGA
   * If the DFC traces do not show delay, analyze Charles traces
   * Verify component configurations