<epam>

# Data Quality

**Testing process in DWH/BI development lifecycle**

TRAINING

CENTER

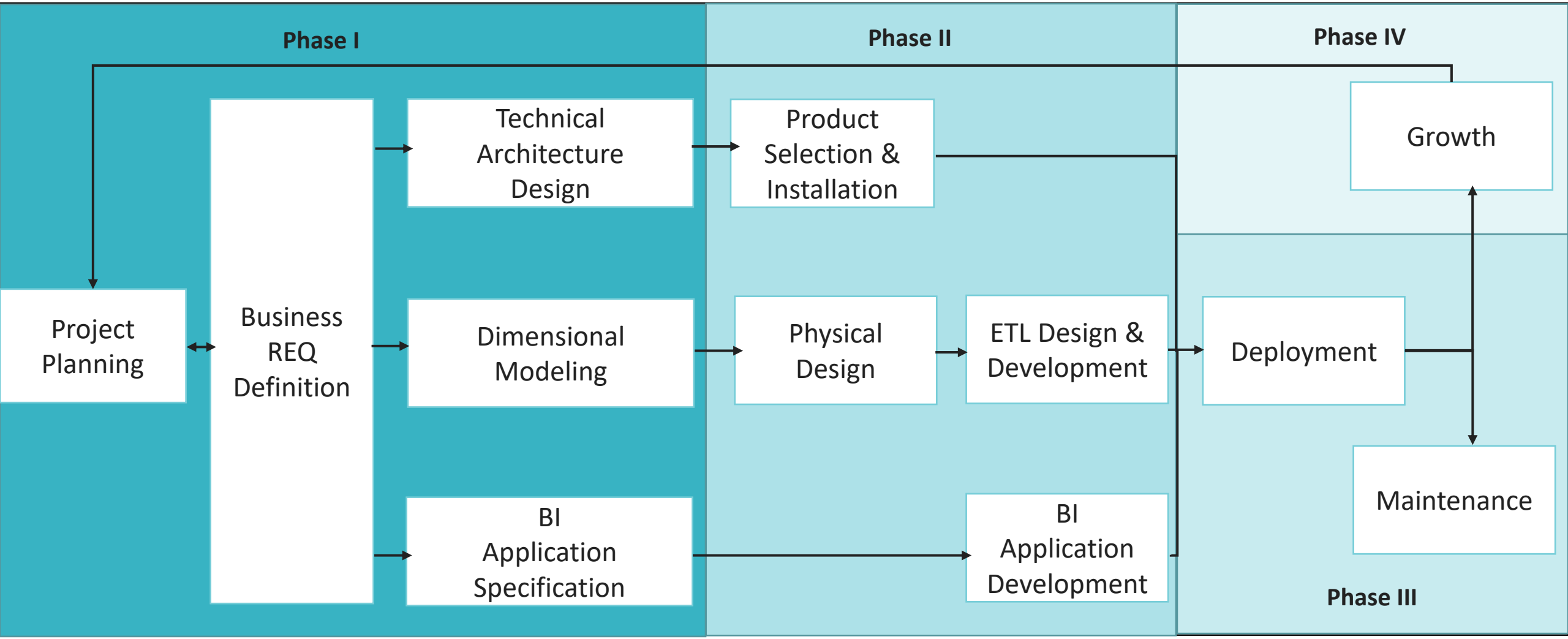
<epam>

# Agenda

- DWH/BI Development Life Cycle
- DWH/BI Testing Process in SDLC

# DWH/BI DEVELOPMENT LIFE CYCLE

# DWH/BI Software Development Lifecycle



# DWH/BI TESTING PROCESS IN SDLC

# DWH/BI Testing Lifecycle

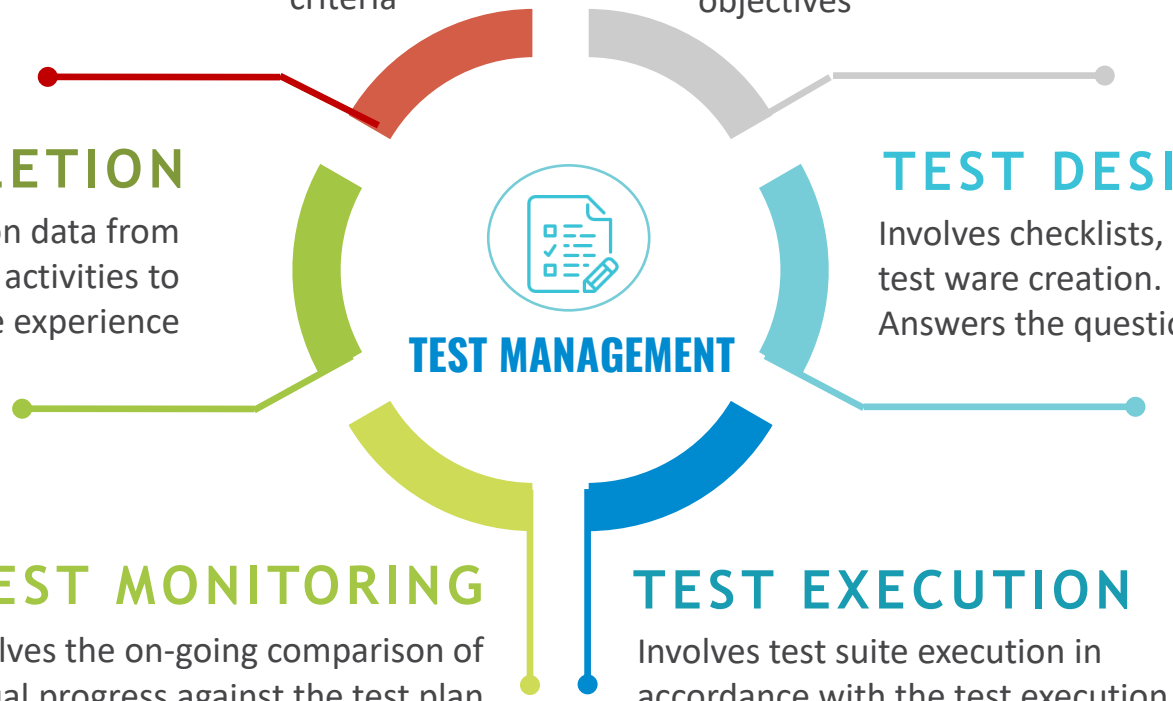
INITIATION

## TEST ANALYSIS

Determines “what to test” in terms of measurable coverage criteria

## TEST PLANNING

Involves activities that define the objectives of testing and the approach for meeting test objectives




## TEST COMPLETION

Involves collection data from completed test activities to consolidate experience

## TEST DESIGN

Involves checklists, test scripts and other test ware creation.
Answers the question “how to test?”

## TEST MONITORING

Involves the on-going comparison of actual progress against the test plan

## TEST EXECUTION

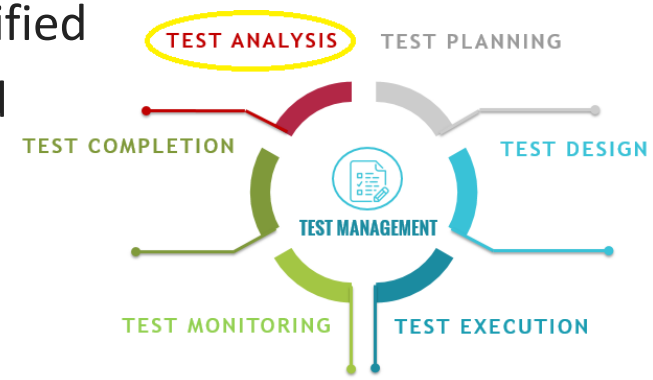
Involves test suite execution in accordance with the test execution schedule

# Activities

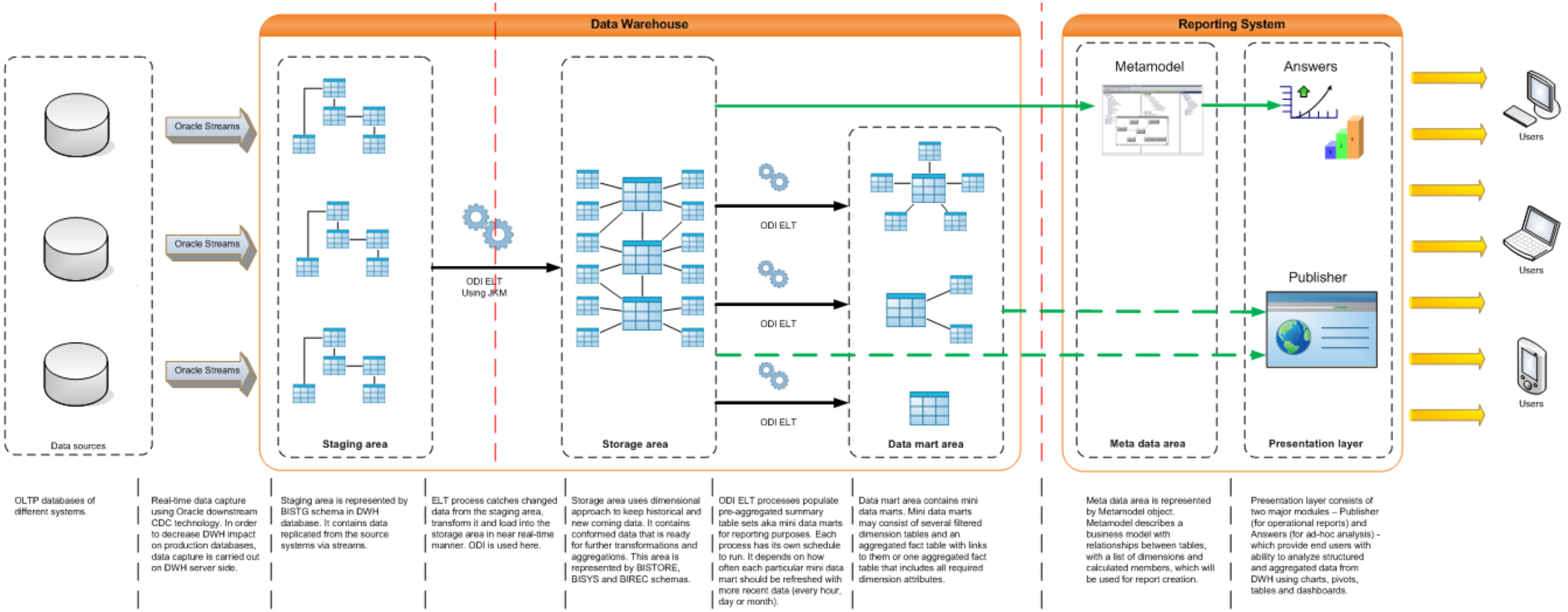
- Software requirements specification, architecture documents analysis
- Evaluating initial documents by characteristics of quality requirement statements
- Identify features to be tested
- Defining and prioritizing test conditions for each feature

# Artifacts

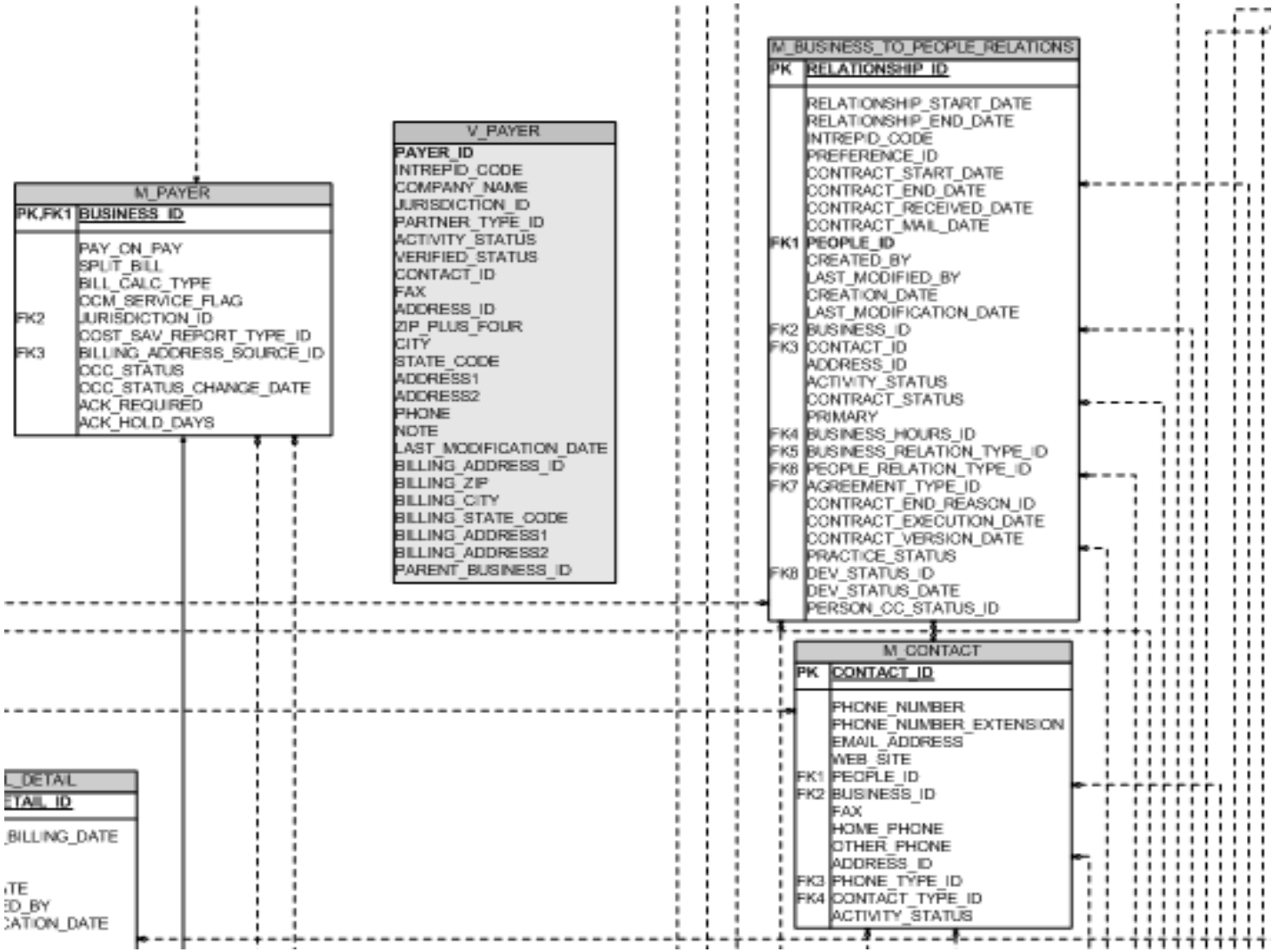
- Documents finalized:
  - Software Requirement Specification (SRS)
  - Source-to-target elements mapping
  - Naming conventions
  - Dataflow diagram
  - High level design
- Defects in documentation identified and fixed
- Outstanding questions clarified
- Acceptance criteria defined



# High Level Design Example



# Source Data Model Example



# Source-to-Target Data Mapping Example

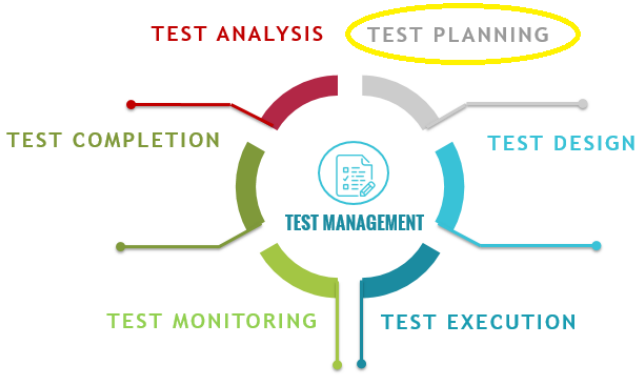
| | G | H | I | J | K | L | M | N | O | P | Q | R | S |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 3NF | | | | | | | | Source Data | | | | |
| 2 | Scheme | Table Name | Column Name | Data Type | Default Value, NULL/not NULL | PK, FK | Expression | Comments | Scheme | Table Name | Column Name | Data Type | Expression |
| 3 | BL_3NF | CE_CUSTOMERS | CUSTOMER_ID | NUMBER(6,0) | not NULL | PK | is generated by sequence | | | | | | |
| 4 | BL_3NF | CE_CUSTOMERS | CUSTOMER_NAME | VARCHAR2(20 BYTE) | not NULL | | | | SA_SRC | EXT_SALES | CUSTOMER_NAME | VARCHAR2(30 BYTE) | INITCAP(CUSTOMER_NAME) |
| 5 | BL_3NF | CE_CUSTOMERS | CUSTOMER_TYPE_ID | NUMBER(1,0) | not NULL | FK (references BL_3NF.CE_CUSTOMER | is generated by sequence | | | | | | |
| 6 | BL_3NF | CE_CUSTOMER_TYPES | CUSTOMER_TYPE | VARCHAR2(10 BYTE) | not NULL | | | | SA_SRC | EXT_SALES | CUSTOMER_TYPE | VARCHAR2(10 BYTE) | ROUND(CUSTOMER_TYPE, 2) |
| 7 | BL_3NF | CE_CUSTOMERS | GENDER_ID | NUMBER(1,0) | not NULL | FK (references BL_3NF.CE_GENDER (G | is generated by sequence | | | | | | |
| 8 | BL_3NF | CE_GENDER | GENDER | VARCHAR2(10 BYTE) | not NULL | | | | SA_SRC | EXT_SALES | GENDER | VARCHAR2(10 BYTE) | |
| 9 | BL_3NF | CE_CUSTOMERS | DISCOUNT_PERC | NUMBER(2,0) | not NULL | | | | SA_SRC | EXT_SALES | DISCOUNT_PERC | VARCHAR2(2 BYTE) | TO_NUMBER(DISCOUNT_PERC, 2) |
| 10 | BL_3NF | CE_CUSTOMERS | CUSTOMER_SRCID | VARCHAR2(10 BYTE) | not NULL | | | isn't transferred to the next layer | SA_SRC | EXT_SALES | CUSTOMER_SRCID | VARCHAR2(5 BYTE) | |
| 11 | | | | | | | | | | | | | |

## Activities

- Define a testing scope, objectives, risks of testing
- Estimate testing efforts
- Define the overall approach of testing, testing strategy
- Plan test data, test environment
- Plan automation scope/CI
- Create, confirm, and publish test plan

## Artifacts

- Test Plan
- Test Strategy
- Data Strategy
- Automation plan



# Main Tasks to be Planned

## SCOPE OF WORK

WHAT **SHOULD BE TESTED** FOR EACH TEST LEVEL (UNIT, SYSTEM, INTEGRATION, ETC.),
**TEST TYPE** (FUNCTIONAL/NON-FUNCTIONAL)
WHAT IS **ENTRANCE** CRITERIA
WHAT IS **ACCEPTANCE** CRITERIA
PLAN **AUTOMATION SCOPE/CI**

## TEST DATA

WHAT **TEST DATA** WILL BE USED
HOW TO **BASELINE** TEST DATA
HOW TO **MANAGE** TEST DATA

## TEST ENVIRONMENT

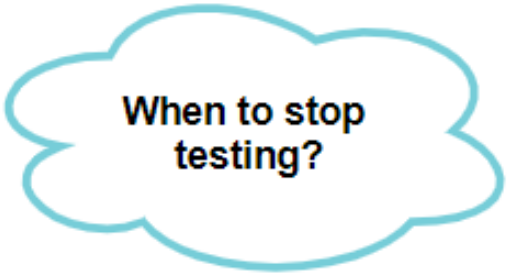
WHAT **TEST ENVIRONMENT** IS REQUIRED
WHAT **SOFTWARE** IS NECESSARY
WHEN **ACCESS** CAN BE OBTAINED

## QA ARTEFACTS

HOW **TEST CASES/CHECKLISTS** WILL BE PREPARED
WHAT **TEST SCRIPTS** ARE REQUIRED
**WHERE** QA ARTEFACTS WILL BE STORED
HOW **DEFECTS/TEST RESULTS** WILL BE MANAGED
WHAT SHOULD BE **DELIVERED**

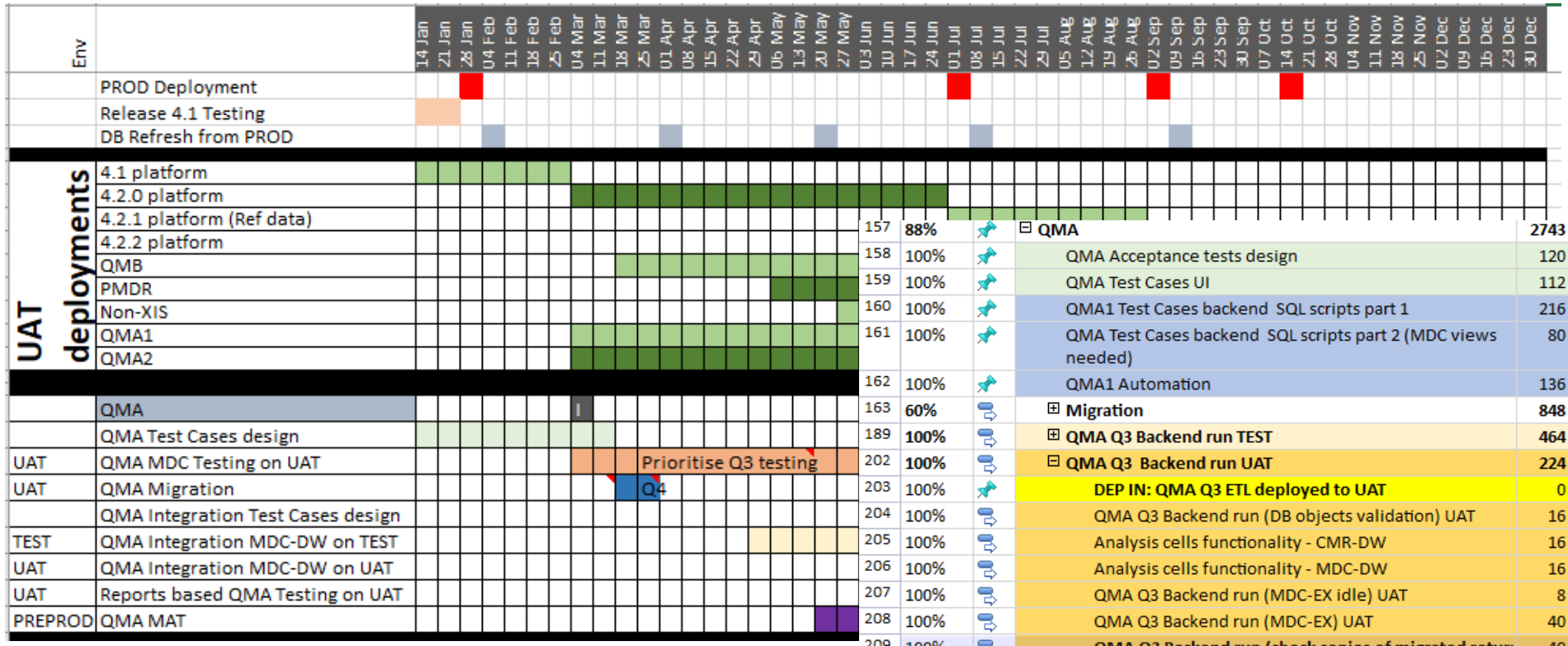
# Entry & Exit Criteria


**Entry criteria** - conditions, which should exist or be met in order to start a process




**Exit criteria** - conditions, which imparts the completion of an activity or meeting of the targets and goals

# Test Schedule - Example



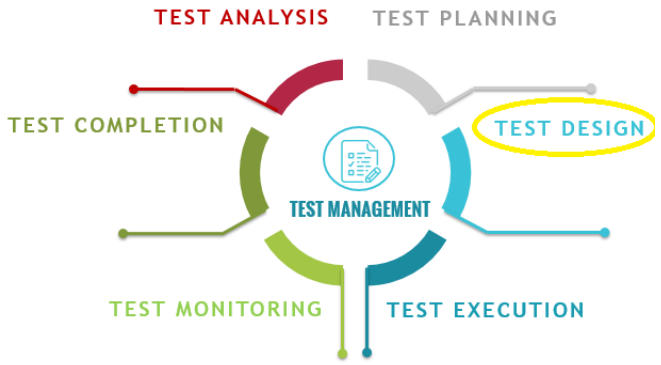

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 157 | 88% | | QMA | 2743 hrs | 150 days | 31/12/18 08:00 | 07/08/19 17:00 | |
| 158 | 100% | | QMA Acceptance tests design | 120 hrs | 19 days | 08/01/19 08:00 | 22/02/19 17:00 | MDC UI |
| 159 | 100% | | QMA Test Cases UI | 112 hrs | 14 days | 25/02/19 08:00 | 15/03/19 17:00 | MDC UI |
| 160 | 100% | | QMA1 Test Cases backend SQL scripts part 1 | 216 hrs | 27 days | 06/02/19 08:00 | 15/03/19 17:00 | Backend |
| 161 | 100% | | QMA Test Cases backend SQL scripts part 2 (MDC views needed) | 80 hrs | 10 days | 19/03/19 08:00 | 08/04/19 17:00 | Backend |
| 162 | 100% | | QMA1 Automation | 136 hrs | 17 days | 05/02/19 08:00 | 27/02/19 17:00 | Backend |
| 163 | 60% | | **Migration** | **848 hrs** | **202.6 days** | **31/12/18 08:00** | **21/10/19 13:48** | |
| 189 | 100% | | **QMA Q3 Backend run TEST** | **464 hrs** | **184.25 days** | **19/03/19 08:00** | **10/12/19 10:00** | **Backend** |
| 202 | 100% | | **QMA Q3 Backend run UAT** | **224 hrs** | **161.25 days** | **15/04/19 08:00** | **04/12/19 10:00** | **Backend** |
| 203 | 100% | | **DEP IN: QMA Q3 ETL deployed to UAT** | 0 hrs | 0 days | 04/06/19 08:00 | 04/06/19 08:00 | |
| 204 | 100% | | QMA Q3 Backend run (DB objects validation) UAT | 16 hrs | 2 days | 15/04/19 08:00 | 16/04/19 17:00 | Backend |
| 205 | 100% | | Analysis cells functionality - CMR-DW | 16 hrs | 2 days | 05/09/19 08:00 | 06/09/19 17:00 | Backend |
| 206 | 100% | | Analysis cells functionality - MDC-DW | 16 hrs | 2 days | 09/09/19 08:00 | 10/09/19 17:00 | Backend |
| 207 | 100% | | QMA Q3 Backend run (MDC-EX idle) UAT | 8 hrs | 1 day | 03/10/19 08:00 | 03/10/19 17:00 | Backend |
| 208 | 100% | | QMA Q3 Backend run (MDC-EX) UAT | 40 hrs | 5 days | 11/09/19 08:00 | 17/09/19 17:00 | Backend |
| 209 | 100% | | **QMA Q3 Backend run (check copies of migrated returı** | 40 hrs | 5 days | 04/06/19 08:00 | 10/06/19 17:00 | Backend |
| 210 | 100% | | QMA Q3 Backend run (Return Header validation) UAT | 16 hrs | 2 days | 15/10/19 13:00 | 17/10/19 12:00 | Backend |
| 211 | 100% | | QMA Q3 Backend - LOAD new MDC data UAT | 0 hrs | 0 days | 04/06/19 08:00 | 04/06/19 08:00 | Backend |
| 212 | 100% | | QMA Q3 Backend run (EX-DWH) UAT | 8 hrs | 1 day | 21/10/19 13:00 | 22/10/19 12:00 | Backend |
| 213 | 100% | | QMA Q3 Backend run (DWH-DM) UAT | 40 hrs | 5 days | 01/10/19 13:00 | 08/10/19 12:00 | Backend |
| 214 | 100% | | QMA Q3 Backend run (Status workflow) UAT | 16 hrs | 2 days | 17/10/19 13:00 | 21/10/19 12:00 | Backend |
| 215 | 100% | | QMA Q3 Backend run (end-to-end) UAT | 8 hrs | 1 day | 03/12/19 10:00 | 04/12/19 10:00 | Backend |
| 216 | 100% | | **QMA Q3 UI** | **463 hrs** | **39 days** | **19/03/19 08:00** | **16/05/19 17:00** | |
| 219 | 100% | | QMA Q3 MAT coordination | 80 hrs | 20 days | 20/05/19 08:00 | 14/06/19 17:00 | MAT |

## Activities

- Select test methods in order to cover risks
- Create, confirm and publish checklists, test scenarios, test cases
- Define automation testing framework architecture
- Detailed requirements analysis, peer reviews
- Establish test environment
- Prepare test data

## Artifacts

- Checklists, test scenarios, test cases, test data
- Traceability matrix
- Automated test framework
- Automated test scripts



# Making the test environment decision

A testing environment is a setup of software and hardware for the testing teams to execute test cases

- Do you need a separate QA env ?
- How many environments do you really need ?
- What is specific of these environments ?
- Is it possible to satisfy your request ?
- Working closely with DevOps team

# Making the test data decision

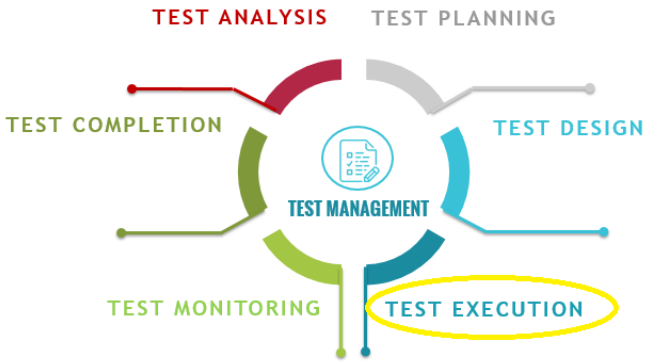
| | What is it? | When we need it? | Advantages | Disadvantages |
|---|---|---|---|---|
| Production data | A subset of production data to represent a portion of the database that is relevant to a test case | Complicated logic and dependencies<br>Historical data required<br>Performance testing | ✓ High quality software in case of complex systems and dependencies<br>✓ Ability to quickly reproduce client's issue | ✓ Security violation: risk of exposing sensitive user data<br>✓ Email addresses, phone numbers, and the like can be accidentally reach users by integration tests<br>✓ Data is changing all the time, so it's more difficult to write stable assertions |
| Production like data | Snapshot of production that has been masked or obfuscated | Only production sensitive data can cover requirements | ✓ Same as production data | ✓ Legal or regulatory requirements mandate anonymizing PII, patient data, financials, and so on, which requires extra effort |
| Synthetic data | Data that is artificially created rather than being generated by actual events. | To protect customer information<br>Required data does not exist<br>Required data has some gaps<br>No access to prod data | ✓ More efficient and cost effective<br>✓ Cover missing cases in real data/ specific cases/ conditions<br>✓ Increased flexibility<br>✓ You are the only owner of your data<br>✓ No secure risks<br>✓ Using less data | ✓ If the system is complex it is a challenge to create high quality synthetic data |

## Activities

- Smoke test execution after deployment
- Thorough testing: test cases and automated scripts running
- Defects reporting, verifying after fix
- Test cases, test scripts, automated scripts updating
- Traceability matrix updating with test results

## Artifacts

- Notification – build verification passed/failed
- Defect reports
- Test artifacts updated

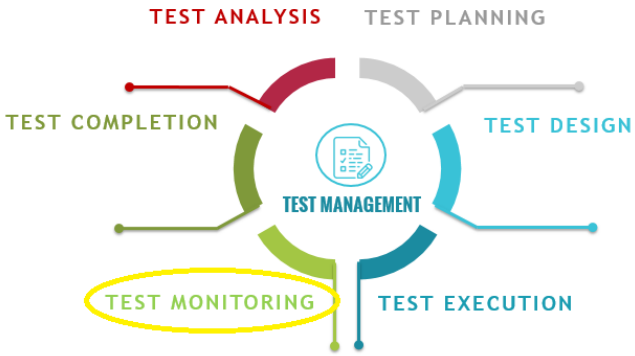


## Activities

- Analyze test results against specified coverage criteria
- Assess the level of component or system quality based on test results
- Determine if more tests are needed
- Create test result report

## Artifacts

- Test result report
- Test cases updated/added
- Test scripts updated



# Test Result Report - Example

## DEV\dev_varyvodA@Virtual_Machine_Number 2020-04-09 02:28:04

**Results Summary**

Pie View

Passed: 142 (100%)

Failed: 0 (0%)

Inconclusive: 0 (0%)



| Tests Statuses | |
|---|---|
| Total | 142 |
| Passed | 142 |
| Failed | 0 |
| Inconclusive | 0 |
| Skipped | 0 |
| Warnings | 0 |

| Run Time Summary | |
|---|---|
| Start Time | 2020-04-09 02:28:04 |
| End Time | 2020-04-09 03:07:31 |
| Duration | 39 min 26 sec |

| Tests Details | |
|---|---|
| User | DEV\dev_varyvodA |
| Machine | GBVM022178 |
| Folder | D:\RunTestsOnTEST |

| Runtime filter list | |
|---|---|
| Category | Compare1to1R1 |
| Category | CompareTransformedDataR1 |
| Category | CompareActualDataR1 |
| Category | ValidationTestsR1 |

**All Tests Group By Classes**

| Time | Status chart | Classes 4 | | Result | Duration | More |
|---|---|---|---|---|---|---|
| 2020-04-09 02:28:05 | | Lloyds.MDC.AutomationTests.Tests.IntegrationTests1_SOURCE_EX<br>The tests compare the data between MDC and EX | Show log | All 39 tests passed | 38 min 31 sec | Show Tests |
| 2020-04-09 03:06:37 | | Lloyds.MDC.AutomationTests.Tests.IntegrationTests2_EX_DWH<br>The tests verify the ETL data transformations between EX and DWH | Show log | All 32 tests passed | 23 sec | Show Tests |
| 2020-04-09 03:07:00 | | Lloyds.MDC.AutomationTests.Tests.IntegrationTests3_DWH_DM<br>The tests compare the actual data (DeletedFlag=0) between DWH and DM | Show log | All 34 tests passed | 23 sec | Show Tests |
| 2020-04-09 03:07:24 | | Lloyds.MDC.AutomationTests.Tests.ValidationTests | | All 37 tests passed | 6 sec | Show Tests |

**Five most slowest tests**

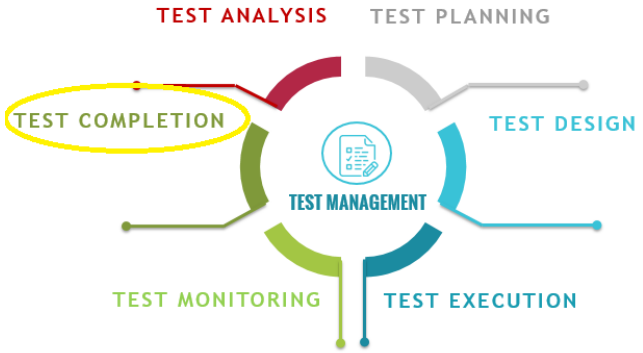
| Time | Status | Class | Test | Duration |
|---|---|---|---|---|
| 2020-04-09 02:54:36 | PASSED | IntegrationTests1_MDC_EX | Compare1to1SBF_MDC_EX(Form_001) | 7 min 5 sec |
| 2020-04-09 02:37:39 | PASSED | IntegrationTests1_MDC_EX | Compare1to1LCR_MDC_EX(Form_002) | 4 min 25 sec |
| 2020-04-09 02:28:55 | PASSED | IntegrationTests1_MDC_EX | Compare1to1LCR_MDC_EX(Form_003) | 4 min 11 sec |
| 2020-04-09 02:42:35 | PASSED | IntegrationTests1_MDC_EX | Compare1to1LCR_MDC_EX(Form_004) | 2 min 36 sec |
| 2020-04-09 02:35:09 | PASSED | IntegrationTests1_MDC_EX | Compare1to1LCR_MDC_EX(Form_005) | 2 min 23 sec |

## Activities

- Check whether all defects closed
- Create change requests or product backlog items for unresolved defects
- Create test summary report
- Analyze lessons learned
- Using the information gathered to improve test process maturity

## Artifacts

- Test summary report
- CR, PBI for unresolved defects
- Lesson learned records



# Software Testing Process

## TEST ANALYSIS

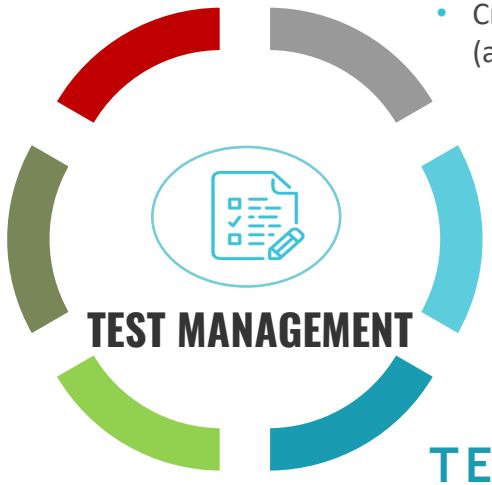
- SRS, Source-to-target elements mapping, Naming conventions, dataflow diagram, High level design documents finalized
- Acceptance criteria defined
- Outstanding questions clarified
- Defects in documentation identified and fixed
- Features to be tested defined and documented

## TEST PLANNING

- Testing objectives, risks defined and published
- Overall approach of testing, testing strategy defined and published
- Testing effort estimated (initial draft at least)
- Test data, test environment defined
- Automation scope/CI defined
- Create, confirm and publish test plan/automation testing framework (architecture)

## TEST COMPLETION

- Sprint backlog verified if all defects closed
- Retrospective meeting conducted
- Lessons learned analyzed
- Information to improve test process maturity gathered
- Test summary report created and shared




## TEST DESIGN

- DQ artifacts specified, created and published (checklists, test scenarios, test cases, etc.)
- Test scripts, automation test scripts created
- DQ artifacts reviewed/signed-off by the customer (PO, etc.)
- Test environment established; test data prepared

## TEST MONITORING

- Test results analyzed against specified coverage criteria
- Test cases, bug reports analyzed in order to determine whether additional tests are required
- Level of component or system quality assessed based on test results
- Test result report created and shared

## TEST EXECUTION

- Smoke test executed after deployment
- Test cases, test scripts executed
- Automated tests executed
- Defects reported and verified after fix
- Test cases, scripts updated
- Traceability matrix updating with test results

<epam>

# Q & A