UnitedHealth Group

Latest Trends in Testing

Ajay K Chhokra

Introduction

- Software Testing is the last phase in software development lifecycle which has high impact on the quality of the final product delivered to the customer.
- Even after being a critical phase, it was not given the importance as it actually deserves. The schedule constraints and slippage carry forwarded from the previous phase also make the testing phase more torrent.
- History reveals that the situation has changed with time, wherein testing is now visualized as one of the most critical, phase of software development. This makes software testing a discipline which demands for continuous and systematic growth.
- Testing today is about Transformation and Adaptation of Latest Testing trends to provide value adds to business
- Testing in today’s era is not just about **Quality Assurance but also Business Assurance**

Establishing and Enabling Testing Trends

Current Test Challenges

Tight Budget

Demanding Customer

Schedule Pressure

Cost reduction without loosing quality

Test Optimization

UnitedHealth Group

Optimizing your Testing Practice for Tomorrow

Test optimization is the discipline of adjusting a testing process so as to optimize some specified set of parameters without violating some constraint.

The most common goals are :

- Minimizing Cost

  (Minimize number of test cases)
- Maximizing Throughput, and/or Efficiency

  (Maximize Coverage and uncovers most of the defects)
- Minimizing Time to Market

  (Reduces the overall test cycle time and thus the Product cycle time)

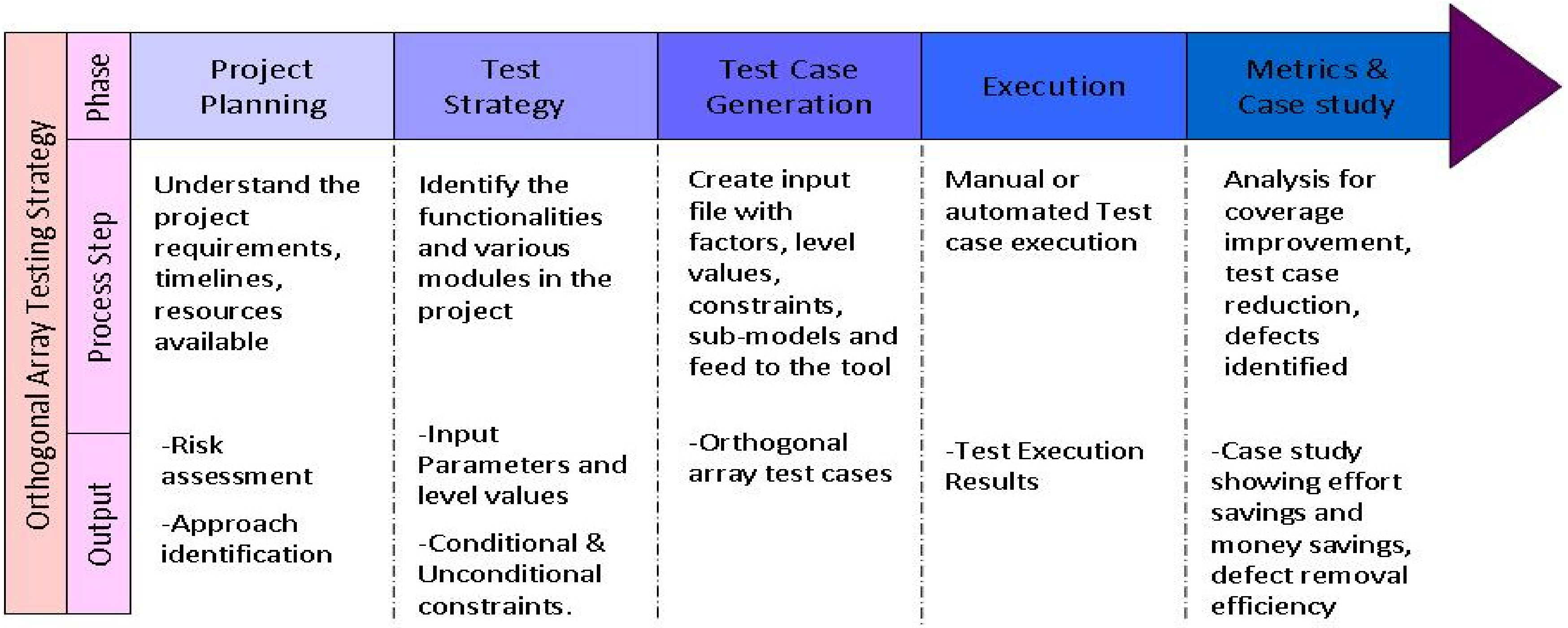
This is one of the major quantitative tools in decision making .

Test Optimization is a discipline to achieve Optimum Cost for testing without compromising on the Quality of Testing.

Orthogonal Array Testing Strategy

UnitedHealth Group

- **The Orthogonal Array Testing Strategy (OATS)** is a systematic, statistical way of testing pair-wise interactions. It provides representative (uniformly distributed) coverage of all variable pair combinations. This makes the technique useful for testing of software components.
- Orthogonal Array is a technique which helps to reduce the test cases to an optimal number while the test coverage is maintained at an adequate level.



Orthogonal Array Testing Strategy					
Phase	Project Planning	Test Strategy	Test Case Generation	Execution	Metrics & Case study
Process Step	Understand the project requirements, timelines, resources available	Identify the functionalities and various modules in the project	Create input file with factors, level values, constraints, sub-models and feed to the tool	Manual or automated Test case execution	Analysis for coverage improvement, test case reduction, defects identified
Output	-Risk assessment -Approach identification	-Input Parameters and level values -Conditional & Unconditional constraints.	-Orthogonal array test cases	-Test Execution Results	-Case study showing effort savings and money savings, defect removal efficiency

Orthogonal Array Testing Strategy

UnitedHealth Group

Benefits

- It guarantees the testing of all pair-wise combinations of all the selected parameters.
- It helps in reducing cost and testing cycle time.
- It helps you to arrive at complex combination of input variables.
- It is simpler to generate and is less error prone than manually generated test cases.
- It helps to deliver a system faster and cheaper and enables our company to remain competitive in today’s market.

Best Practices:

- Select the right values to test with. OATS can fail when you go wrong in selecting the parameters and level values.
- Have a clear understanding of the requirements. OATS can fail when you don’t know how the variables interact and what the dependencies between the parameters are.
- Pay attention to important combinations. OATS fails when highly critical combination gets too little attention

Risk Based Testing

Risk Based Testing (RBT) is the technique used to prioritize the development and execution of tests upon the impact and likelihood of failure of the functionality or aspect being tested.

- Focuses test activities on tests that will mitigate the greater risk.
- Creates an optimized test suite with fewer test cases, and the ability to uncover most of the bugs.
- Increases test effectiveness while potentially resulting in fewer tests, thus reducing cost while maintaining or increasing test effectiveness

Risk Based Testing					
Phase	Project Planning	Test Strategy	Test Case Preparation	Execution	Metrics & Reporting
Process Step	-Understand problem statement, -Risk identification in the application	- Identify business critical functions	-Create / Identify test cases based on prioritized functionality	Manual or automated Test case execution of business critical test cases and test scripts.	-Metrics generation includes - Requirement coverage -Test coverage - DRE
Output	-Risk assessment -Approach identification	-High priority functionality.	-Test suite of prioritized test case ready	-Test Execution Results	-Metrics provides confidence on the product

Risk Based Testing

UnitedHealth Group

Benefits

- High priority requirements and test cases are tested early in the testing life cycle .
- High priority defects are detected early in the life cycle
- Increases test effectiveness while potentially resulting in fewer test cases thus reducing costs.
- Quantitative report of software risks involved helps in better decision making
- Injects confidence into the product by testing the critical aspects of the project.
- Focused approach on critical user requirements at testing phase results in easier acceptance of software by the customer

Best Practices:

- Have a clear understanding of the business critical functionality depending on risk, severity and criticality.
- Create and prioritize business critical test cases in the test planning phase by taking into account probability of failure and impact on the application
- In the test execution phase, execute the test cases having high and medium priority and then go for rest of the test cases if the time permits.

Decision Table Based Testing

Decision table helps us look at the “complete” combination of inputs and select test cases based on logical conditions required to test the system requirements.

Decision Table Technique						
	Phase	Project Planning	Test Strategy	Test Case Generation	Execution	Metrics & Case study
	Process Step	Understand the project requirements, timelines, resources available	Identify the functionalities and various modules in the project	Identify the parameters and possible values for the parameters, constraints, and feed to the tool available	Manual or automated Test case execution	Analysis for coverage improvement, test case reduction, defects identified
	Output	-Risk assessment -Approach identification	-Input Parameters and possible values -Conditional constraints.	-Decision Grid with all the possible combinations based on the input provided	-Test Execution Results	-Case study showing time & effort savings and money savings, defect removal efficiency

Decision Table Based Testing

Benefits

- It ensures adequate coverage for test design.
- It can be extensively used in those systems where input parameters can be categorized in if-then-else logic.
- Test conditions and test coverage review becomes easier.

Best Practices:

- Have a clear understanding of the requirements of the functionality to be tested.
- Divide the functionality into small modules having maximum of 5 parameters.
- Select the parameters and possible values for the parameters carefully as the test cases will be dependent on what you select
- Identify all the constraints on the parameters based on your understanding of the functionality and generate the decision grid.
- Use a tool if available for better accuracy to generate the decision grid based on input values and constraints.

CTDP - CIT
(Common Test Data Preparation - Common Integration Testing)

UnitedHealth Group

- The purpose of this CIT-CTDP is to optimize the test data prep and reutilization of testing data between different application.
- We basically use this technique where we have similar kinds of data that is being used by different applications or where we see that there is data flow from one application to another.

Implementation

CTP can be implemented between any two applications having dependency on each other like Upstream/downstream relationship.

- come up with one common test plan, DTP focusing commonality of business condition
- jointly review the test plan and divide between the two areas for data preparation and execution

CTP can be implemented between any two applications not having dependency on each other

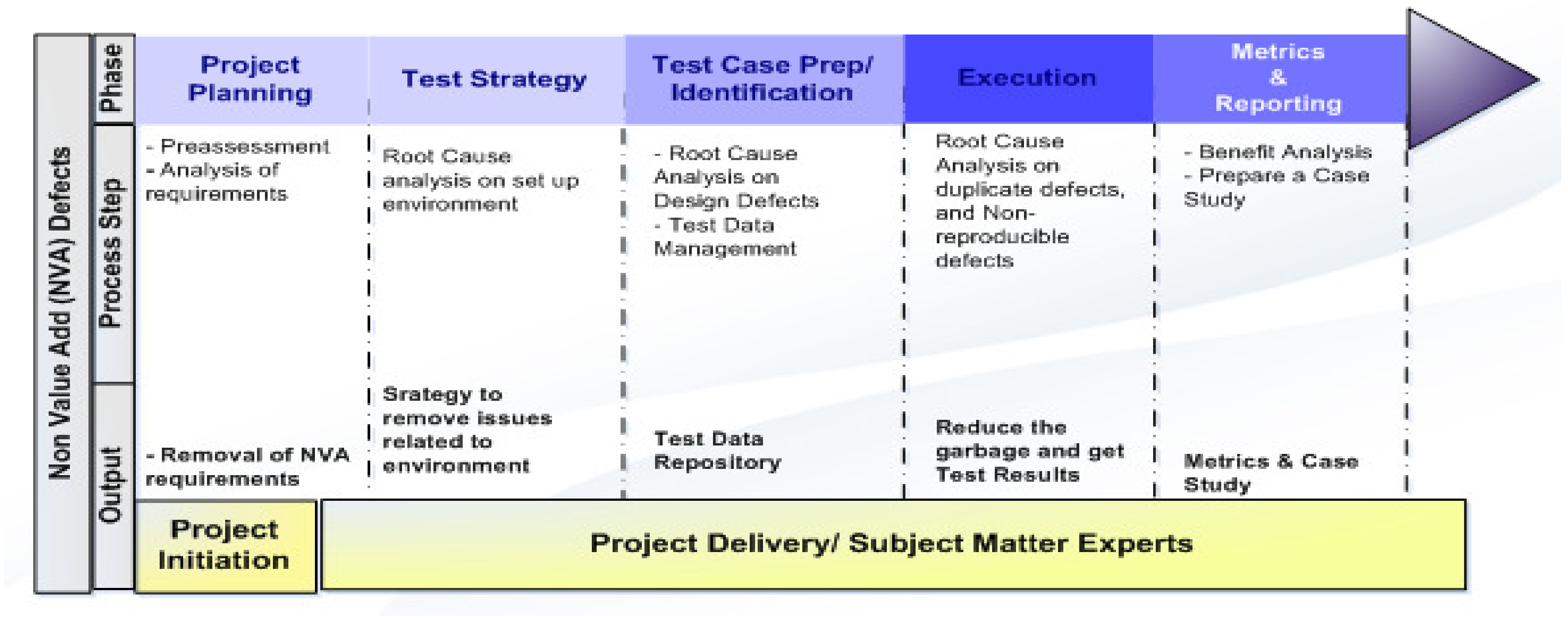
- come up with their DTP independently
- jointly review them and seek data
- remove overlapping cases; add integration scenarios & run test plan independently

Benefits

§ This CIT-CTDP will optimize the effort that you are putting in test data preparation.
§ It will save hours and reduce re-work which we are doing for different application undergoing similar testing.
§ Better validation and verification of Common Application functionalities.

Non Value Add Defects

UnitedHealth Group



Non Value Add (NVA) Defects
Phase
Process Step
Output
Project Planning
Test Strategy
Test Case Prep/ Identification
Execution
Metrics & Reporting
- Preassessment
- Analysis of requirements
Root Cause analysis on set up environment
- Root Cause Analysis on Design Defects
- Test Data Management
Root Cause Analysis on duplicate defects, and Non-reproducible defects
- Benefit Analysis
- Prepare a Case Study
- Removal of NVA requirements
Srategy to remove issues related to environment
Test Data Repository
Reduce the garbage and get Test Results
Metrics & Case Study
Project Initiation
Project Delivery/ Subject Matter Experts

Non Value Add Defects

UnitedHealth Group

Benefits

- Reduces the time consumption of Dev & QA to track and investigate invalid defects.
- Reduces the delay in attention and resolution of valid defects and maximize product quality.
- Reduces the number of duplicate defects, test error, test data change and the WAD defects in the application.
- Reduces project cost with optimum resource utilization.

Best Practices

- After requirement analysis use the inspection method to identify the defects which doesn't add any value.
- After each execution cycle observations are made and efforts are to be put in to reduce the garbage.
- NVA defect analysis should be done once in a while and when you feel the NVA% is higher than the industry/company standards.

UnitedHealth Group

Automation
-Right and Adaptable

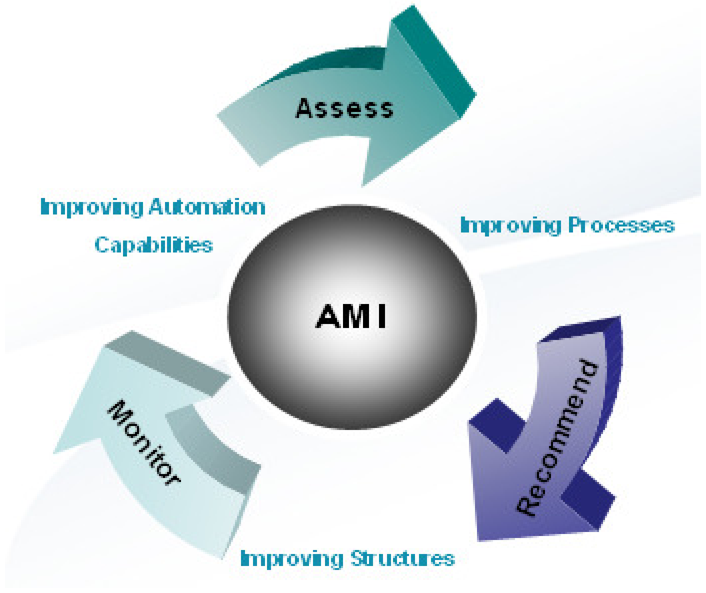
Automation – Focus on Right Automation

UnitedHealth Group

Adapt to Automation Maturity Assessment:

- *Automation Maturity is a process improvement approach that helps application test teams assess and improve their automation capabilities.*

- *By performing a gap analysis assessment, a test team can determine the current state of their regression and identify the opportunities and scope to improve and advance their automation capabilities*



Focus on Right Framework

UnitedHealth Group

- Optimize Distributable Frameworks
  - Optimized automation – *Performance Tuning,*
  - Uniform Coding standards
  - Pretested, Preconfigured, Packaged and Certified
- Reusable Component Libraries
  - Increase Quality and Reliability – *Tested and verified every use*
  - Reduction of development time – *Build once use many times*
  - Optimized for Reusability and Repeatability
- Automation Framework Training
  - Promotion of Operational Efficiencies
  - Enterprise Governance
  - Vetted Automation Best Practices
- Metrics
  - Framework should enable ROI driven Metrics

Automation Frameworks-Keyword Driven

UnitedHealth Group

- What:
  - Easy-to-understand “**Keywords**” to describe the actions to be performed on the application under test.
  - Excel sheets to create and manage steps in test cases and data drive the steps
- Why:
  - Enables non-technical personnel to create automated tests
  - Helps to construct automation scripts with common easy to use Excel interface
- Effort:
  - Automation Expertise in Keyword module development for Application specific functionality
  - Automation Expertise with modular deployment plan an strategy for specific architecture and test actions

ARCHITECTURAL

- INTERFACE Keywords:
  - “INPUT”
  - “VERIFY”
  - “CLICK”
  - “GET VALUE”
  - “WAIT”

APPLICATION

- ACTION Keywords:
  - “LOGIN”
  - “LOGOUT”
  - “PAY CLAIM”
  - “FIND CLAIM”
  - “VALIDATE PAYMENT”

COMMON

- EXTERNAL Keywords:
  - “COMPARE STRING”
  - “GET TIME”
  - “GET DATE”
  - “GENERATE NAME”
  - “CREATE TEXT FILE”

Automation Frameworks- Modular

UnitedHealth Group

- What:
  - Uses Architectural and Common modular libraries of functions to facilitate quick development of Application specific scripts and modular libraries
  - A Modular Library framework divides an application into modules which are used to build tests and facilitates reuse and manageability of code base
- Why:
  - Building out scripts becomes a matter of compilation of highly reusable tested functions from all modules
  - Enables even greater modularity, maintainability and reusability as more modules are created
- Benefit:
  - High degree of reuse insures stability of scripts and lowers development time
  - Highly distributable – Lowering cost of testing

ARCHITECTURAL Modules

- Web, Mainframe, Dot Net, ZK, Powerbuilder , Visual Basic

APPLICATION Modules

- Critical Business Functionality, Common Interface validation approach, Error

COMMON Modules

- Reporting, Data Management, Database validation, Quality Center, Date, Time, String, File, Database, Directories, Random name generation

Cross Browser Testing

UnitedHealth Group

- **Scalability**:

  – To validate the same functionalities on different browser/versions like Internet Explorer, Mozilla Firefox, Google Chrome & Safari without any framework change

- **Highly extensible**

  –Generic solution which can be extended for other web applications with creation of objects repositories and few customizations in the framework

  –For incorporating new browser/versions/operating systems

  –Addition and Deletion of keyword related functions

- **Capability:**

  –Framework capable of running on multiple browsers (IE, Chrome, Firefox, Safari).

  –Validations (Broken Links, Image, text, button, pdf file, pop up validation.

  –Selective execution and Data driven with multiple combinations of data.

  –Automation solution can be deployed for regression, GUI and functional testing.

  –HTML Reporting @ Test Case and Test Step Level.& automatic mail delivery with result to stakeholders.

  –Error and Exception handling ,Test Scripts can be executed in unattended mode.

  –Centralize storing of framework (shared drive) allows less maintenance for end users

- **Compatibility**

  –Framework compatible with Windows and MAC operating system as per implementation

- **Technical knowledge**:

  Focus on Frameworks that require

  – Zero technical knowledge required for user to write and execute test cases

  –Easy to learn

CBT Automation Tools

- Tool – Selenium 2.0

  (Selenium RC and Web Driver)

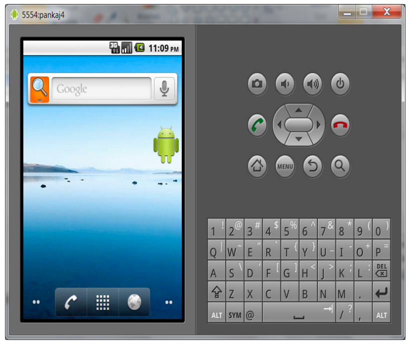
- Browser – Internet Explorer (6,7,8,9), Chrome 12.0 , FireFox 3.6, Safari 3

Mobile Testing automation

UnitedHealth Group

Mobility Framework

- Extensible
- Easily customizable and adaptable for web applications
- Execution on Chrome/FireFox to simulate different mobile platforms and Android Emulators
- Easy to learn- Even a manual tester can implement automated tests
- HTML Reporting @ Test Case and Test Step Level.
- Error and Exception handling
- Test Scripts can be executed in unattended mode.
- Outlook Email Integration



Mobile Automation Tools

- Tool – Selenium 2.0, Sikuli, Zapfix etc
- Browser – Chrome 12.0 , FireFox 3.6 etc
- Mobile Emulator – Android 2.1,2.2 etc

UnitedHealth Group

Enable QA's - Aligning QA with Business

Quality Assurance To Business Assurance

UnitedHealth Group

- QA enablement
- Focus on Business Needs
- Deliver Business Value