Ideal Test Systems & Target Development Test Process

Tony Chang

June 9, 2010

www.huawei.com

HUAWEI

Contents

- “Test” Fundamental Responsibilities & Activities
- Product Delivery Test Quality Indicators
- Product Test Coverage
- Ideal Test Systems
- Target Development Test Process
  - Pre-Testing Preparation
  - Development Testing
  - Continuous Integration
  - DTTP – Design, Test, Tools Partnership
  - Continuous Regression

“Test” Fundamental Responsibilities & Activities

- Build up the Test case Baseline & Coverage management system
- Develop new Feature Test Design & Plan
- Develop the Solution Integration Test Strategy
- Develop & Execute new Test cases – manual & automated
- Promote Test Automation technology
- Fault Identification, Trouble Report & Retest
- Test Environment Management

“Product Quality Assurance” is our #1 mission

Product Delivery Test Quality Indicators

- **All test cases have been executed & regressed with 95%+ Pass rate**
- **Non-Critical/Gating remaining issues**
- **Manageable number of Minor un-resolved issues**
- **Provide remaining issues Patch Plan**

BUT:

- Number of Test cases cannot direct reflect the Product’s Quality
- Number of bugs found during the Product Verification cycle cannot reflect the Product’s Quality as well

The key factor to evaluate the Product Test Quality is the “Product Test Coverage”

Product Test Coverage

- **Requirements-based Test Coverage**
  - **Test the product/Software behavior against the every claim made in the Requirements & Design specifications**
- **Functional Test Coverage**
  - **Test the product/Software behavior against the Requirements, Design specifications, Customer Scenarios and examines what the program accomplishes from various areas:**
    - Level 1 new feature basic function， PQRS Test Coverage
    - Level 2 new features' interaction Test Coverage
    - Level 3 new & old functional interaction Test Coverage (N-1)
- **Regression Test Coverage**
  - **Product Test case Baseline**
- **Code Coverage**
  - **Is a measure used in software testing to describe the degree to which the source code of a program has been tested and via the analysis process we'll also be able to:**
    - Find areas of a program not exercised by a set of test cases
    - Create additional test cases to increase coverage
    - Determine a quantitative measure of code coverage, which is an indirect measure of Quality
    - May indentify redundant test cases that do not increase coverage

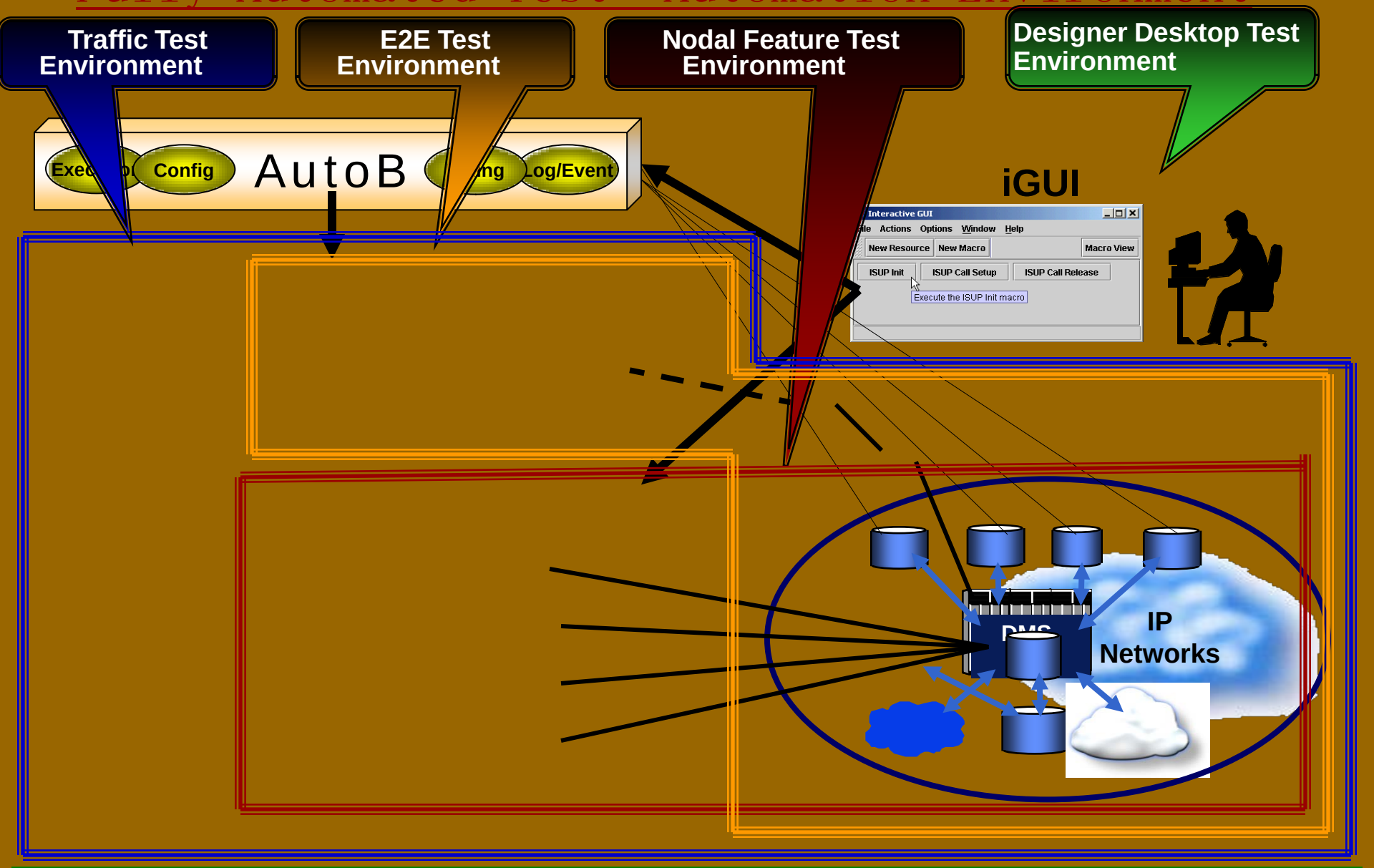
Ideal Test Systems

www.huawei.com

Ideal Test Systems

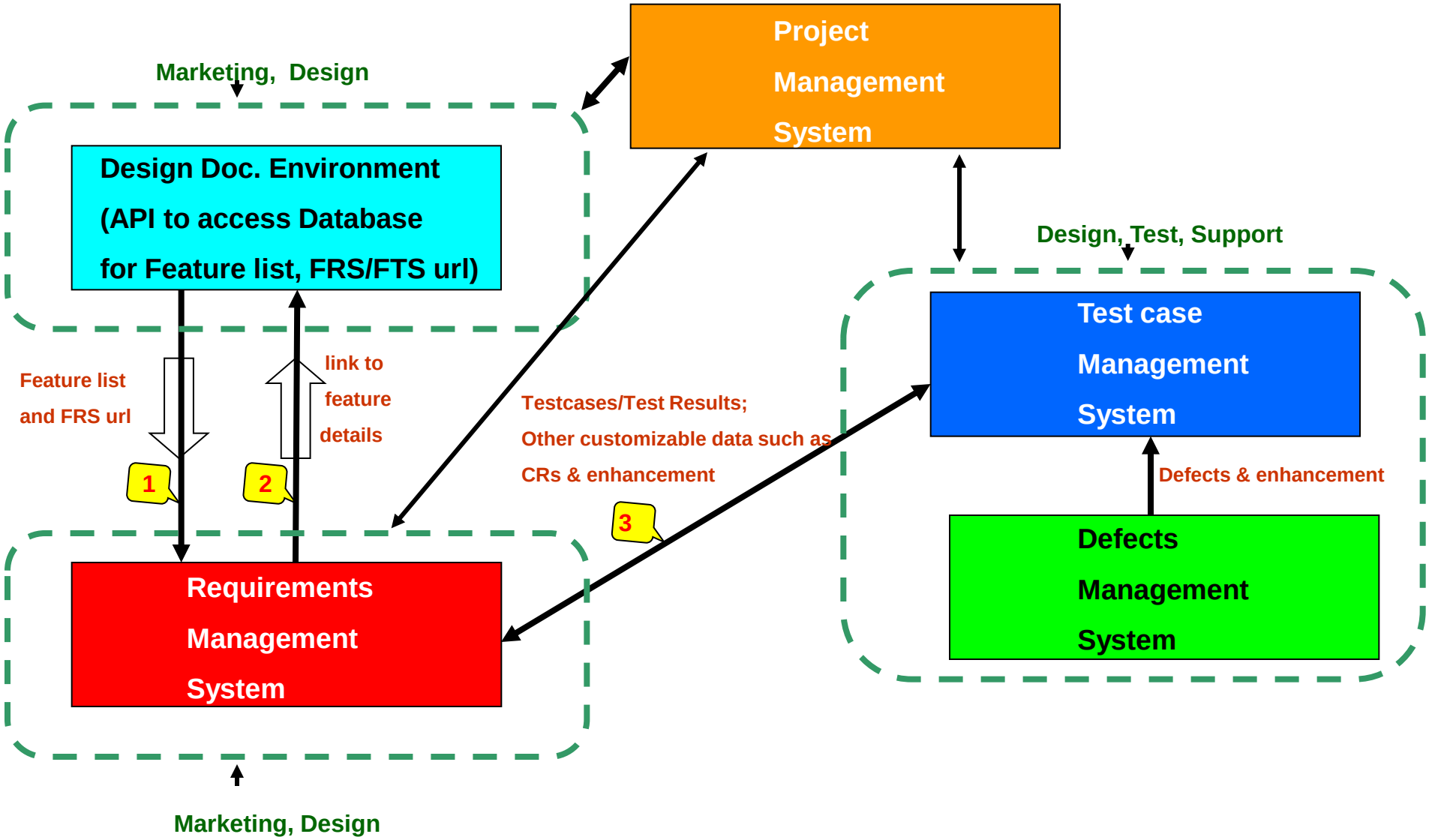
- ☑ **Fully Automated R&D Test Automation System：** Code，Module，Story，Feature，System， Customer Acceptance Test
- ☑ **Fully Integrated R&D Management System**：Requirements，Documentations，Test，Project，Production，Engineering，Beta/CAT
- ☑ **Measureable Product Test Cases Coverage**
- ❑ **Measureable & Visible Products Quality Process:** Requirements，Development，Test，Project，Production，Engineering，Beta/CAT
- ❑ **Customized Test Case Execution Strategy**
- ❑ **Commercialized Test Environment & Tools**：Product Solution Integration ，Networks Solution Integration，Customer Solution Integration Center

Fully Automated Test Automation Environment



* Single Framework for Nodal, E2E and Traffic
* Test Case Sharing
* Global Lab Sharing
* Designer Desktop Testing

Fully Integrated R&D Management System



Test Case Coverage/Management Objectives & Benefits

Objectives

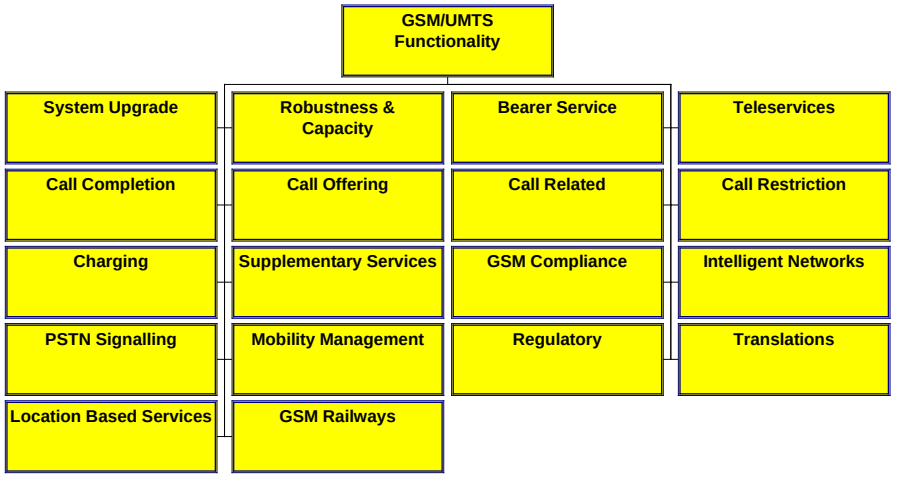
- **Ensure Product Test case Baseline Coverage**
- **Enforce Test case and Test Coverage ownership**
- **Improve Test Case Management (Quality, Efficiency, Coverage, Maintenances …)**

Why do we need to implement TCM ?

- **Enforce the corresponding primes to improve Test case quality and integrity**
- **Vehicle to ensure & improve Product Test Coverage & Code Coverage**
- **Vehicle to proper upgrade the Regression Test cases to the next Release**
- **Vehicle to ensure the effectiveness of “Agile”, “Iteration”, “RBT” & “DFT”**
- **Vehicle to ensure the System Integrity & implementation of “VBS”**
- **Efficient Test cases Execution & Maintenance**
- **Efficient Quality management (Eg. CR/Patch Analysis)**
- **Vehicle to build up Product Test Knowledge base**

TCM and Test Coverage Improvement

Test Case Management -- by Subsystems



Test Case Clean up Process

Manageable Subsystems:

- **Test cases were grouped into manageable Subsystems**
- **Test cases executed on TICC per subsystem to understand the failures**
- **Failures analyzed and test cases corrected**
- **Test cases executed on TICC as frequently as possible, results analyzed and test cases cleaned up after every run to improve results**

Coding Standards:

- **Test case coding standards published and used to write and correct the test cases**
- **TEP coding standards published and adopted to improve test case execution results on TICC**

Test Case Ownership -- Subsystems

- Organize Product related Test cases into manageable pieces called **Subsystems**
- Subsystems represent functional areas. Assigning **Owners** for each Subsystem
- Subsystem Owners/Primes responsible for documentation, execution, analysis, debug, creating and maintaining TP/TC
- Each subsystem is further broken down and composed by multiple **Funcnode**. Each Funcnode is composed by multiple **Funcleaf**, which is the smallest functionality unit in regression.
- Subsystem Owner assigned, Design Manager identified

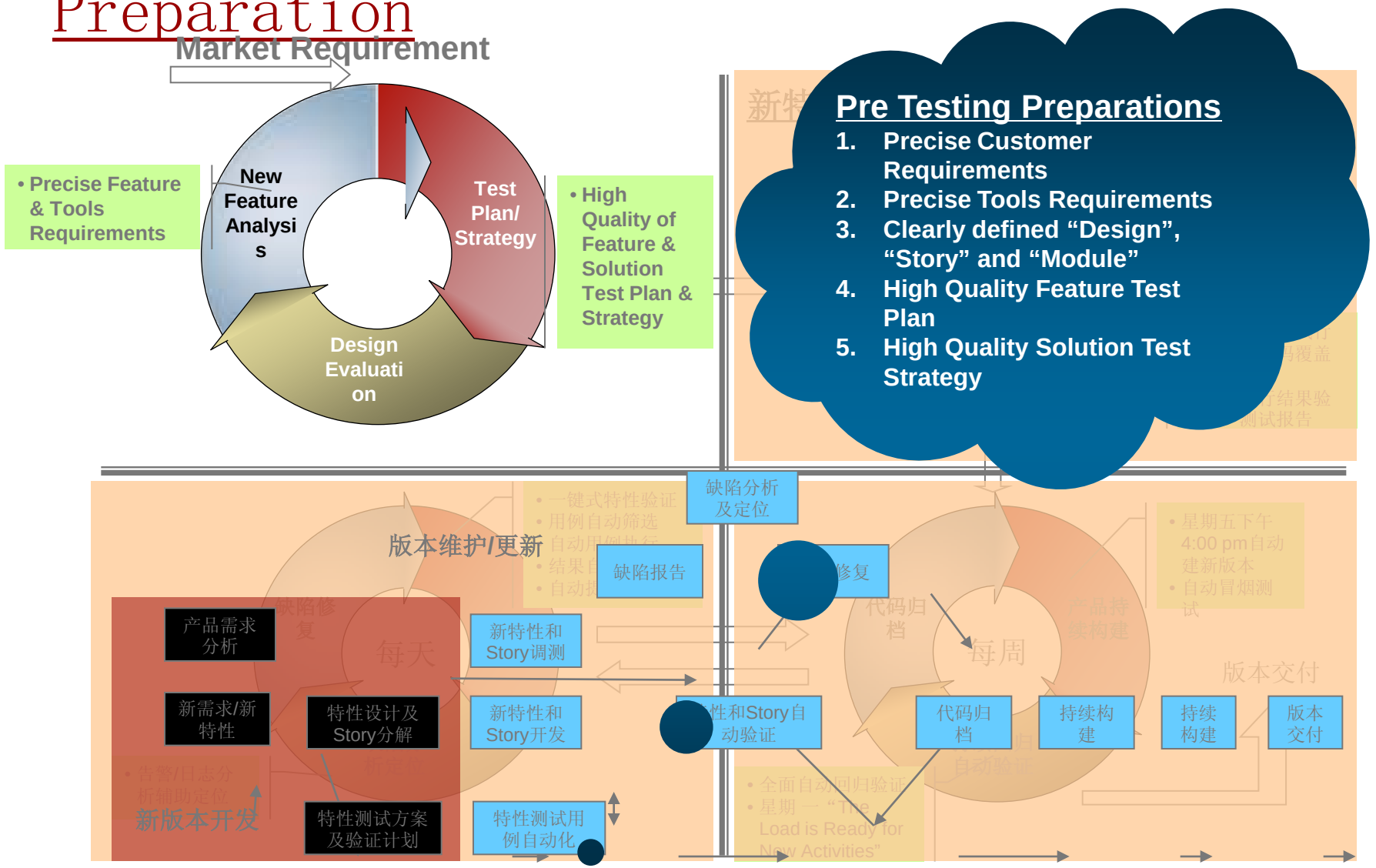
Test Coverage Improvement

- The Subsystem Owners of the identified functionality will research the coverage improvement and write test cases to plug the holes in their subsystem coverage
- TCM test plan captures the coverage and functionality improvement
- Identify the design subsystems and modules for that test subsystem
- Update test strategy document with test coverage/ Software module information and update document with code coverage results

Target Development Test Process

www.huawei.com

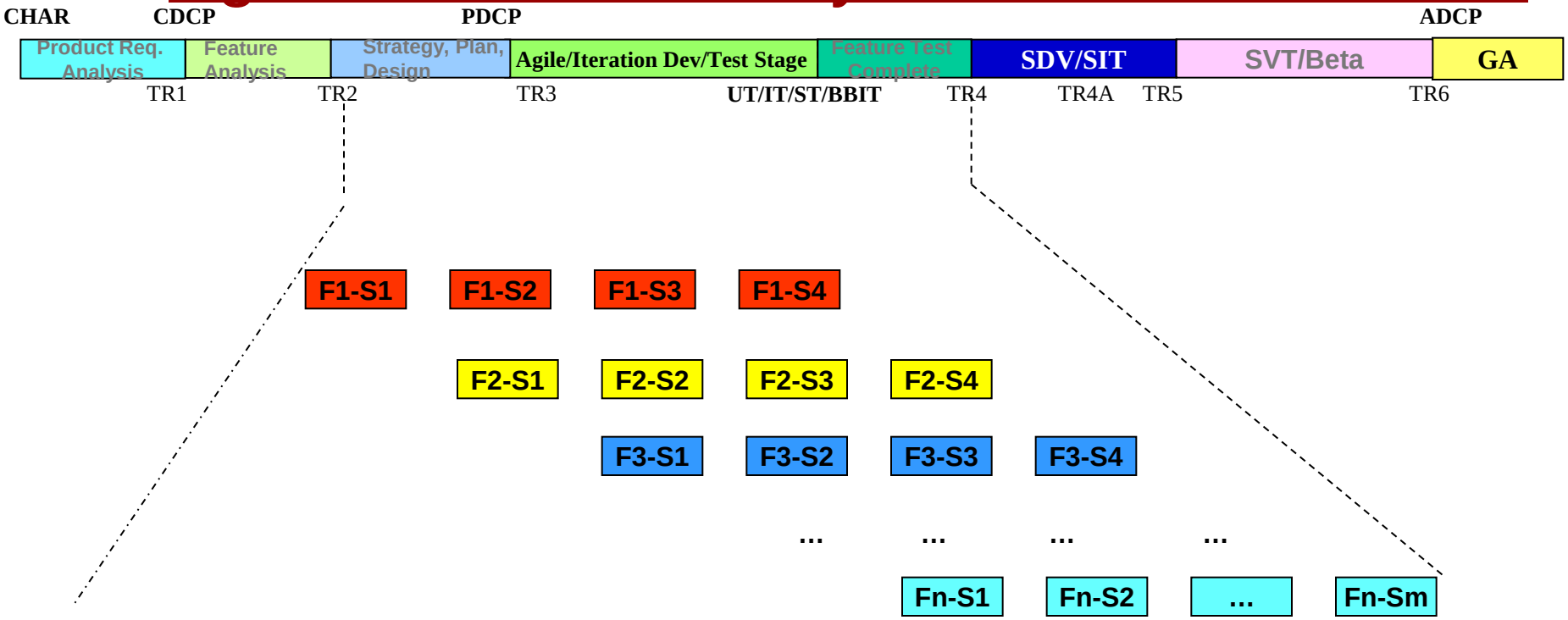
Target DI Process – Pre Testing Preparation



The Importance of Feature Test Plan

- Tester’s most critical deliverable during the test cycle
- Can evaluate the tester’s feature knowledge
- Can evaluate the new feature’s test coverage
- Pre-define the test progress & evaluation point
- Ensure the Requirements Traceability’s coverage
- Provide the guideline for DTTP (Design, Test, Tools Partnership)
  - define the “Capability Test cases” & schedule with designer
- Decide which test cases will be used for Regression
- Identify which test cases’ results can be shared with Data testing
- Lift up the testers test capability & product knowledge

Agile & Iteration Story-Based Test Process



Feature 1	Story 1	Story 2	Story 3	Story 4	Remaining	Total
White box TC	10	5	10	15	10	50
Black box TCs	10	15	20	25	30	100

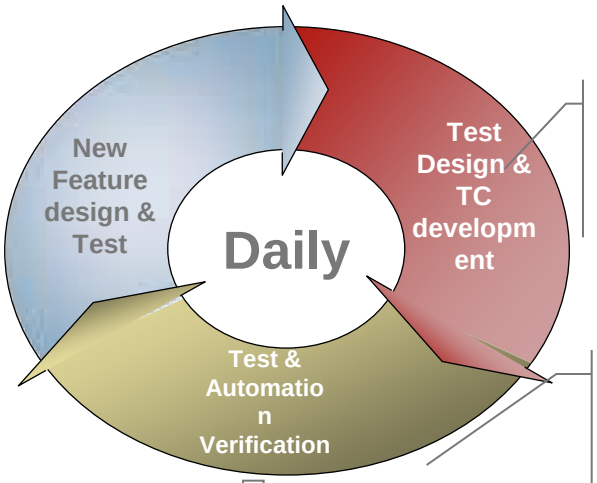
- Both the designers & testers are owning the features together – “DTTP, Design Test Tools Partnership”
- Clear metric to measure each Story’s progress & Status
- Story 1—4 needs to be completed Coding & Testing prior to TR4 (completed the Agile designer Testing/LLT)
- Remaining Test cases will need to be completed at TR4 (completed the Feature testing)

Target DT Process – Development Testing

New Feature Devel. Testing

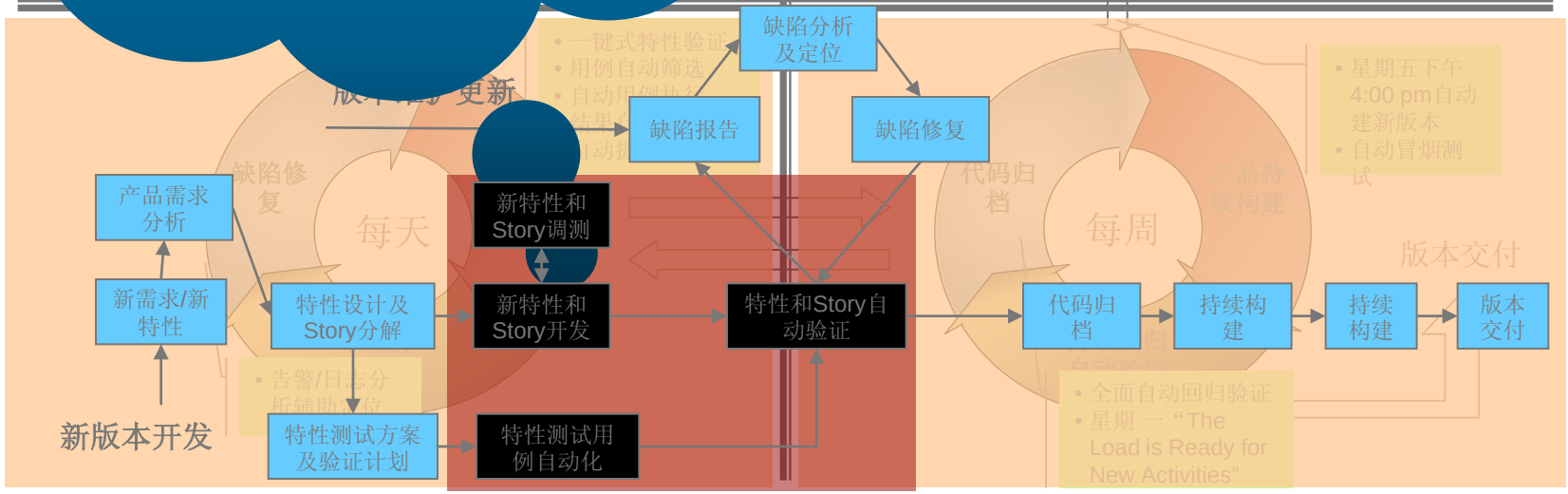
1. Integrate “White box” test tool
2. “Grey box TCs” generation & Execution based on Test Modeling & Record Playback
3. Auto-Results & Log Analysis
4. DTTP—Design, Test, Tools Partnership, early Feature Test Automation
5. Designer Desktop Testing for Capacity

Feature Development Testing

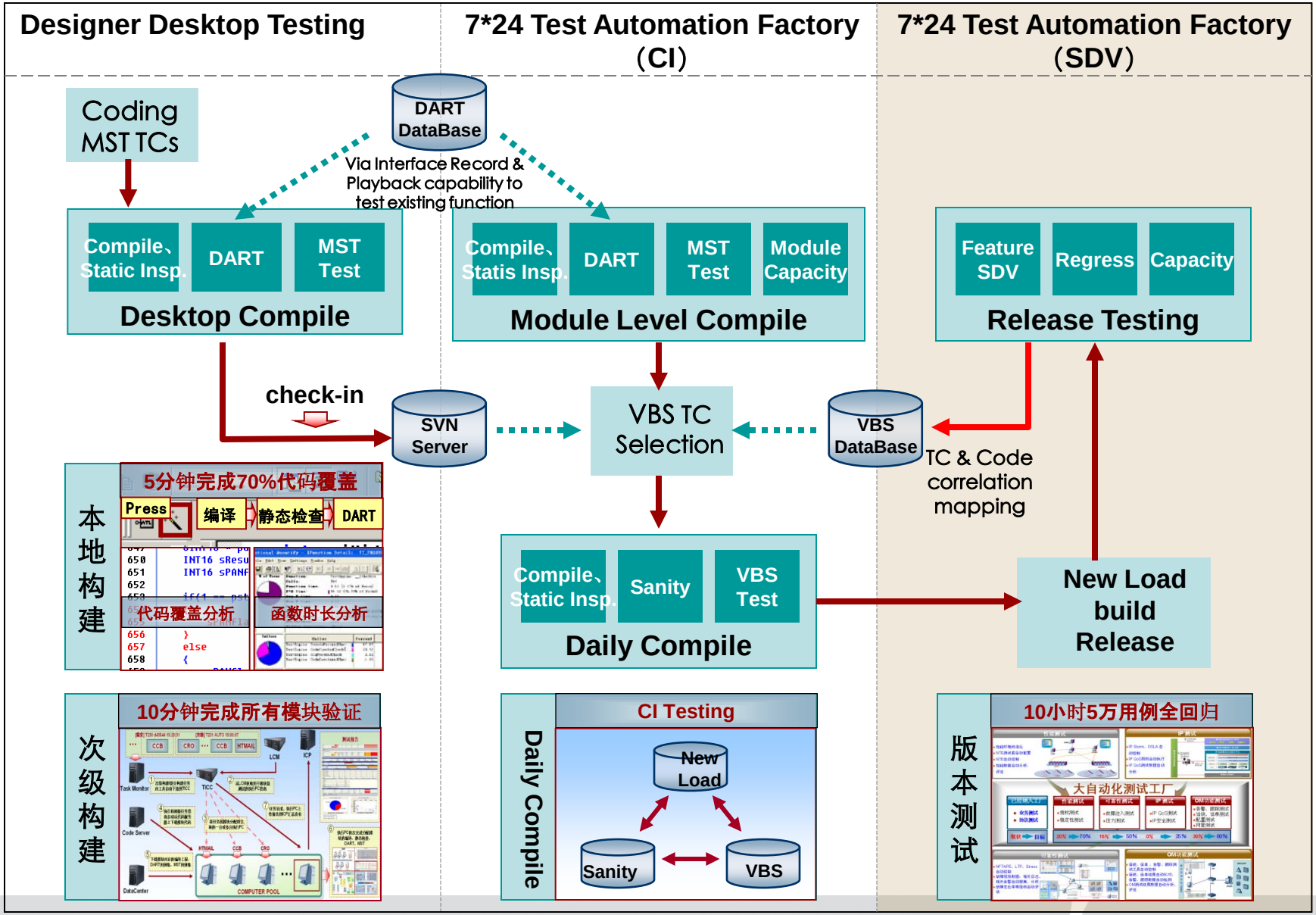


- Auto-TC Generation for “Module/Grey box” testing

- MBT
- Fully Automated Test Automation Environment



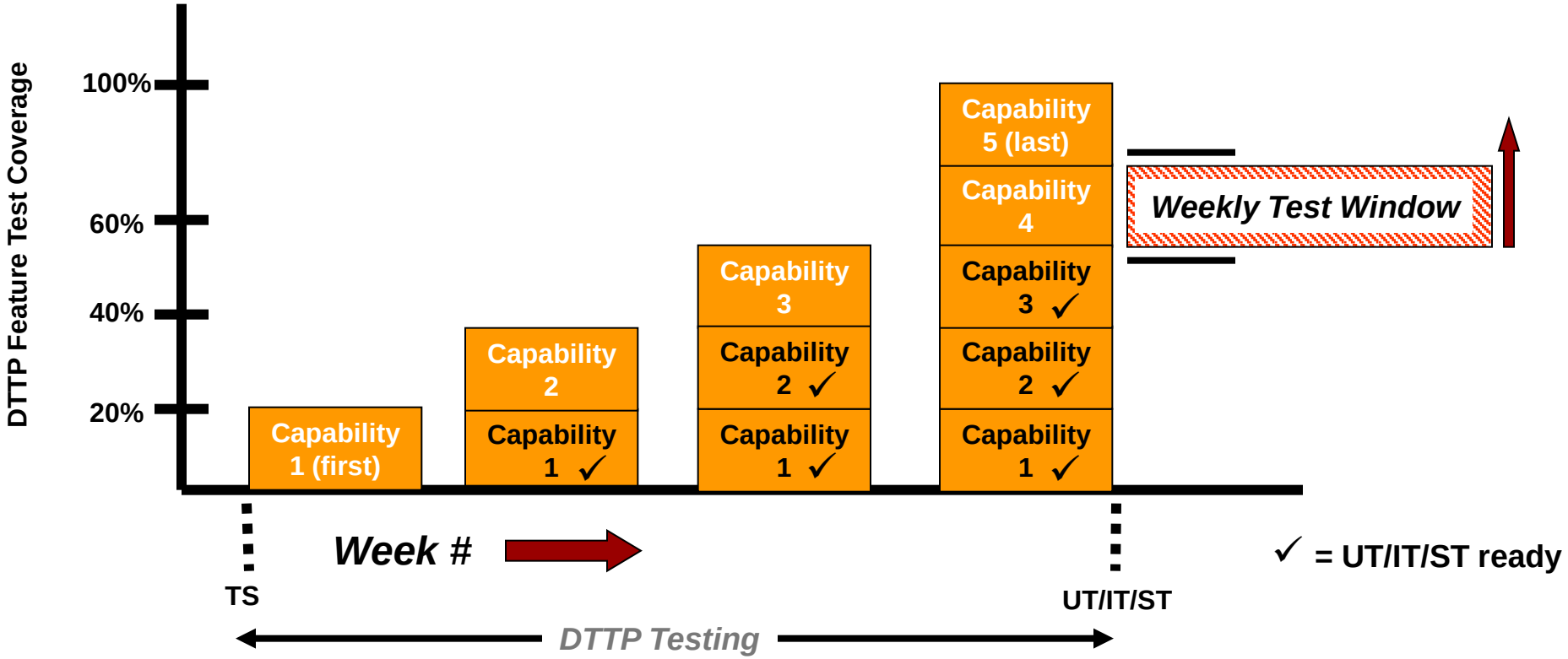
Fully Automated Designer Test Automation Environment



DTTP (Design, Test Tools Partnership) Objectives:

- Pull forward the Test & Tools teams engagements at the early stage of the Requirements & Test Analysis cycle
- Ensure the Feature Quality during the TR3 & TR4 interval
  - Identify the Quality issues at the early stage of the release cycle
  - Shorten the Issues Resolution time during the development & test cycle
- Shorten the overall development & test cycle due to the parallel testing
- Enforce the Communication & Partnership among Design, Test & Tools members
- Enforce the Early Test Automation from the beginning of the Release Cycle
- Accommodate the implementations of Iteration, Agile & Early Test
- Lift up the Capability of the Test & Tools teams

The March to UT/IT/ST:



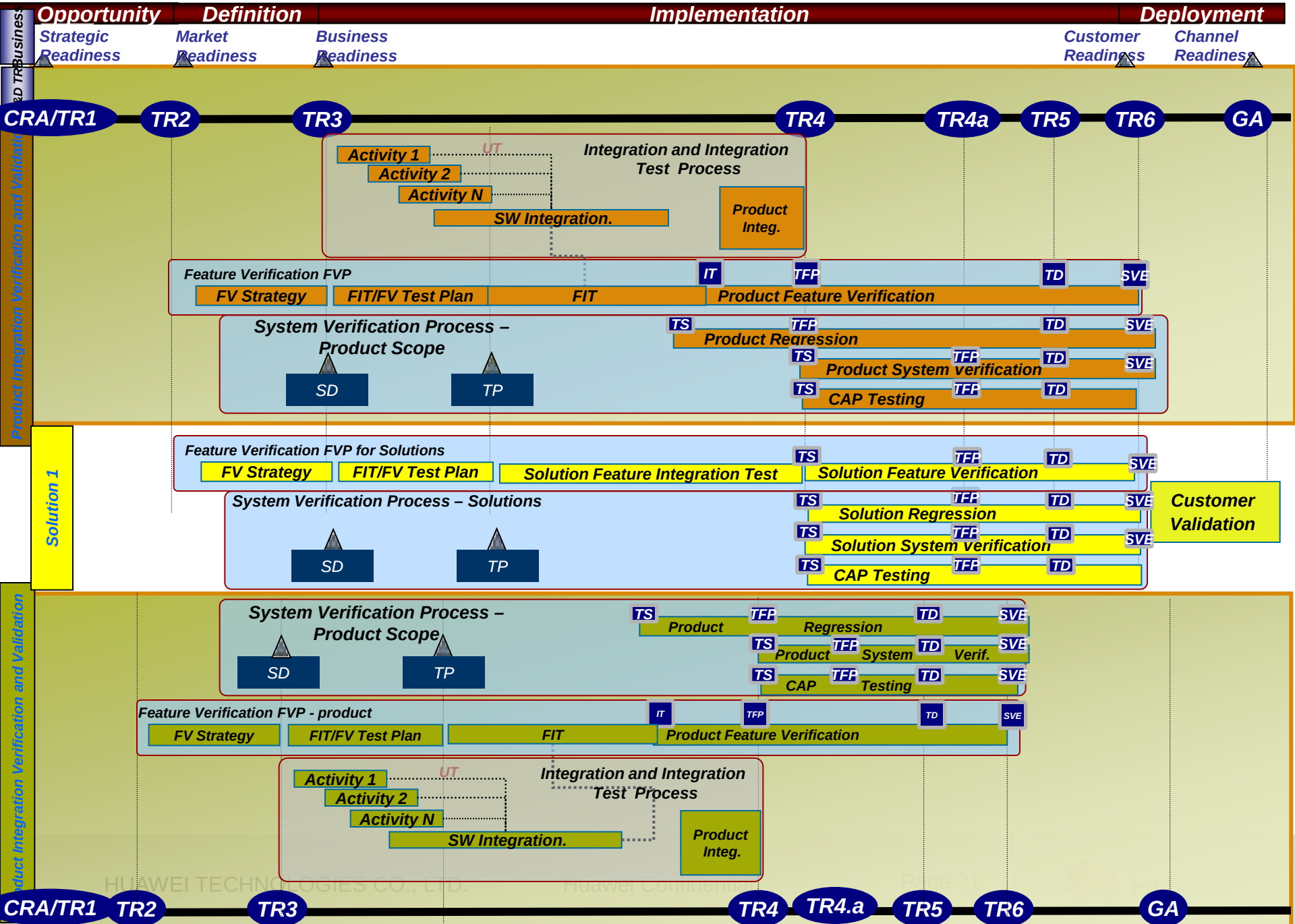
Timely contributions from each group (Design, PV, Tools) required to make capability completion a success!

DTTP Benefits

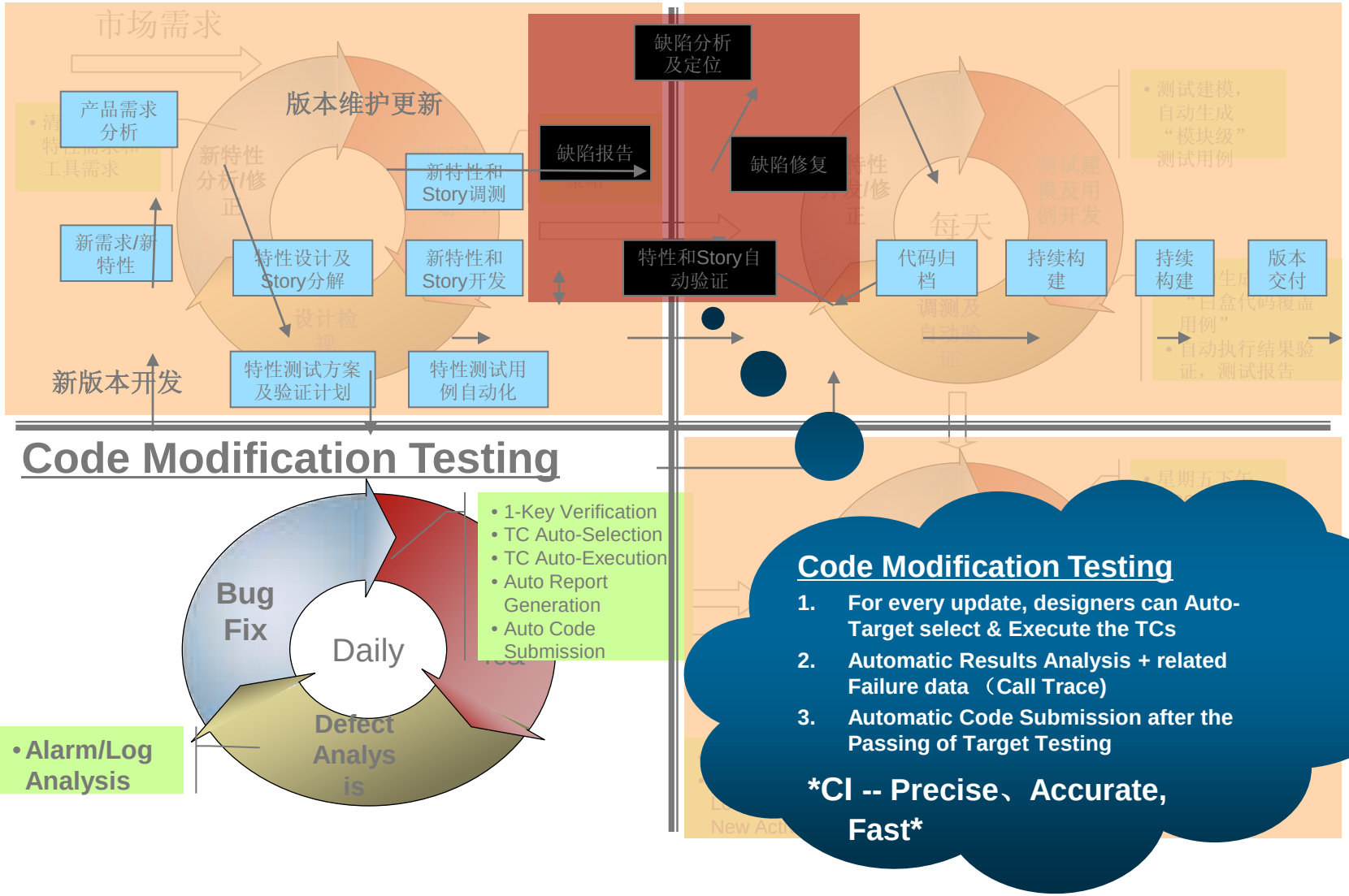
Ensure Product Early Quality、Shorten Test Cycle & TTM

- **Pull forward the Test & Tools development activities**
  - Early Test
  - Enable & Support the designer's testing
  - Support the Agile Test Strategy – early Test Automation
- **Lift up the Product Early Quality – design & test in-sync**
  - Code Ready
  - Tools Ready
  - Test cases/Scripts Ready
- **Doing the Test Automation from the beginning**
- **Shorten the Development Test Cycle & TTM**
- **Lift up the capabilities for Testers & Tools Developers**

Solution Verification Process Reference – Clear Milestone Expectations



Target DT Process - Development Testing



Validation Before Submission

1

•Software Updates/Increments
•(New Features or CRs Updates)

Feature Updates

•Design Prime compiles and creates modules/patches
• Submit the Request to VBS process

2

CRs Updates

• CRs designers compile and create modules/patches
• Submit the VBS process

2

TMS

3

Test Cases Selection based on TCs & S/W Mapping

TCs Optimization

Compile & Pkg Test cases

Retrieve Test cases

AutoBrain

4 Test Execution

Test Scripts

Execute Test cases

Monitor Status

Test Reports

5 Result Analysis

Validate Billing/Log

Generate Results

Billing/Log Files

Expected Result

Result Files

Collect Billing/Log

Match Test case

6

Module Integrity Suite pass at 100%?

No

•Correct issues
•Build it into new modules/patches

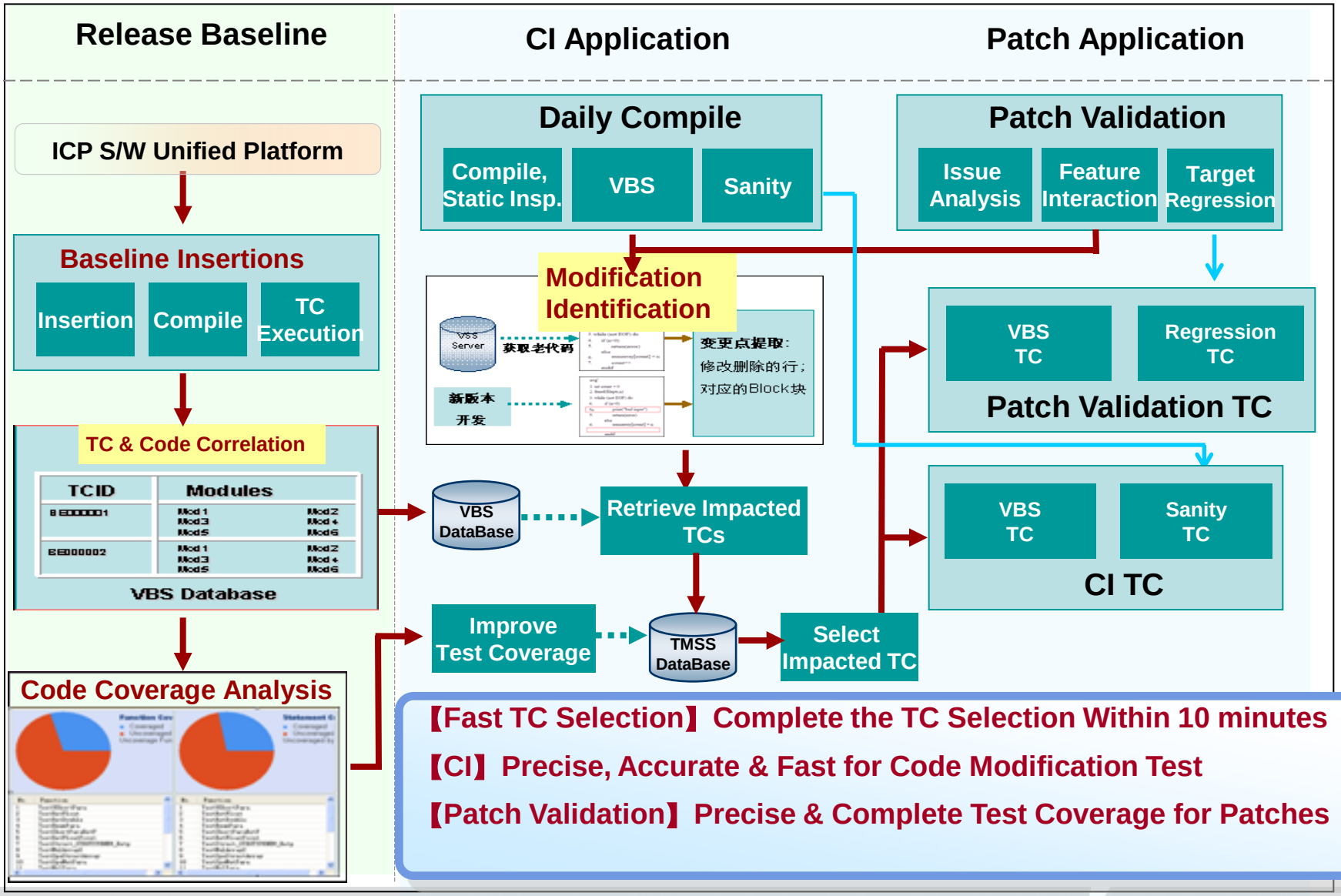
2

Yes

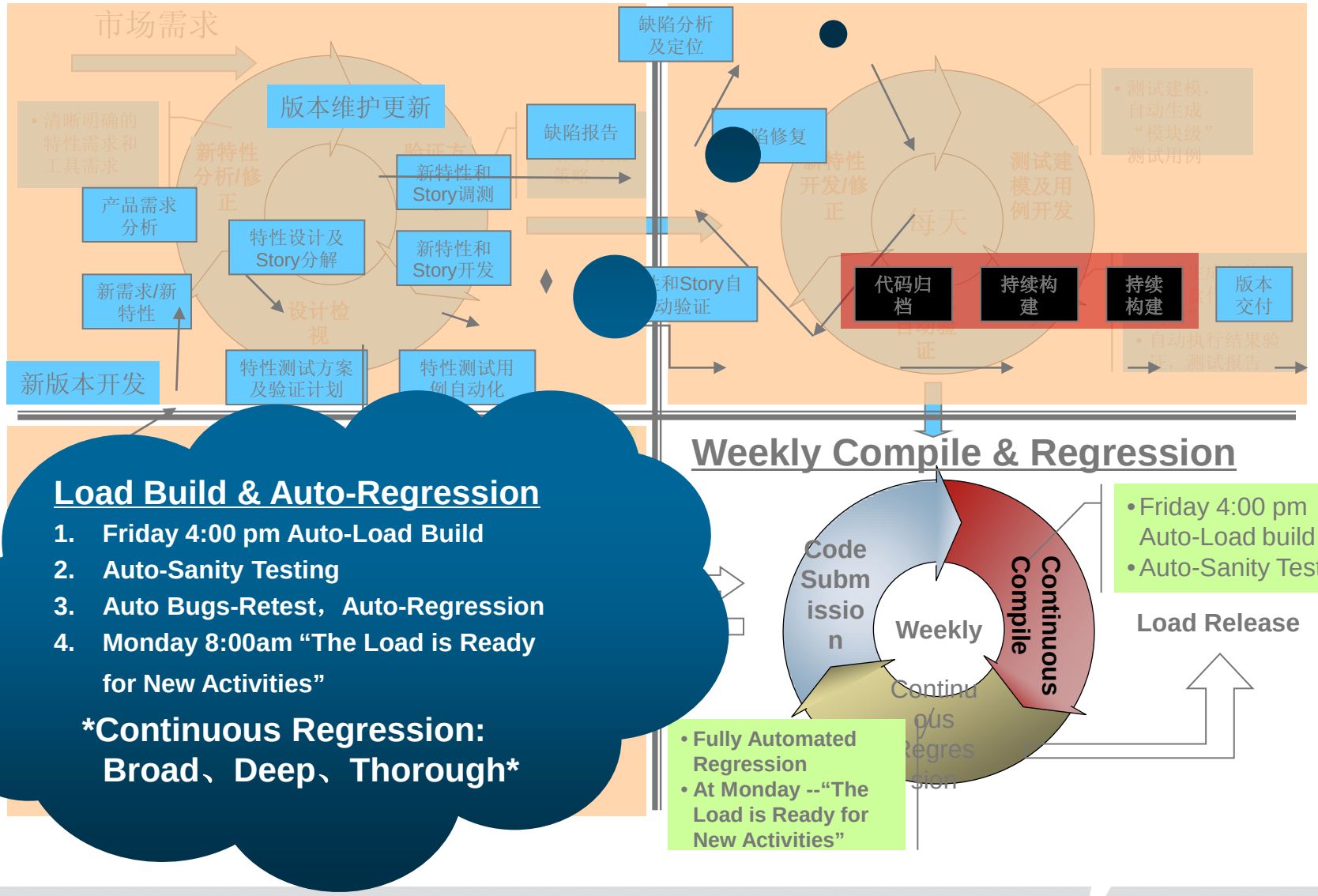
•Submit Module to Weekly S/W Load build
•Put Test Case ID's in the update text

7

VBS Applications – Continuous Integration & Patch Validation



Target DT Process - Development Testing



Load Build & Auto-Regression

1. Friday 4:00 pm Auto-Load Build
2. Auto-Sanity Testing
3. Auto Bugs-Retest，Auto-Regression
4. Monday 8:00am "The Load is Ready for New Activities"

Continuous Regression: Broad、Deep、Thorough

Thank You

www.huawei.com