Company name	Type of document		
Volvo Group Truck Technology	Instruction		
Name of document	**Issue**	**Reg. No.**	**Page**
PVT Handbook	1.1		1 (35)
Issuer (dept, name, phone, location)	**Sign**	**Date**	**Info class**
Johan Adoson, BF59990, +46 31 32 27814, AB3N	JA	2013-05-14	SC
Approved by (dept, name, phone, location)	**Sign**	**Date**	**Valid**
Jessica Erlandsson, BF59990, +46 31 3221513, AB3N	JE		
Receiver (dept, name, geogr plac)			

PVT HANDBOOK

To Verify and Validate Functions in a truck from a driver perspective

Purpose with this document:

- Be a guideline for new personnel involved in PVT
- Be a reference for all personnel involved in PVT
- Be a source of information for customers and other stakeholders
- To secure that a PVT is performed in a harmonized way

This document is under constant update and any printed copy may be out of date.
To find the latest issue of this document, please visit: https://teamplace.volvo.com/sites/3p-EEE-CV/default.aspx

Any feedback on this document is most welcome, send it to: johan.adoson@volvo.com

Contents

1. What is TFV/PVT?

Truck Function Validation (TFV) is a team performing Performance Validation Tests (PVT) on trucks within AB Volvo. TFV verify and validate truck functions in a truck from a driver perspective. TFV belongs to Complete Vehicle Organization which is a part of Volvo Group Trucks Technology (Volvo GTT).
Main tasks, Responsibilities, Authorities etc. can be found here: link

1.1. Roles

The following roles are present within the TFV organization

Role	Type	Primary objective	Comment
PVT Engineer	Local	Lead a PVT in a truck incl participate in the testing	Support PVT development (Processes, Methods and Tools)
PVT Driver	Local	Perform a PVT in a truck	Support PVT development (Processes, Methods and Tools)
PVT Planner	Local	Plan all PVT, Coordinate tests and resources	Manager, Coordinator and/or delegated
PVT Tester	Local	Support PVT testing with truck knowledge expertise	
PVT Troubleshooter	Local	Troubleshoot issues and support PVT Engineers	
PVT Coordinator	Local	Local responsible for PVT	All different roles
PVT Developer	Global	Develop PVT tests	
PVT Truck Function Specialist	Global	Develop PVT Processes, Methods & Tools (PMT)	
PVT Tools	Global	Develop tools for PVT and their users	
PVT Global Coordinator	Global	Coordinate sites including startup	
Customer	Local	Order a PVT and receive the report	

To handle Personnel, Budget, and PMT's a manager controls the team.
More details are available in each role description: link

1.2. What TFV do

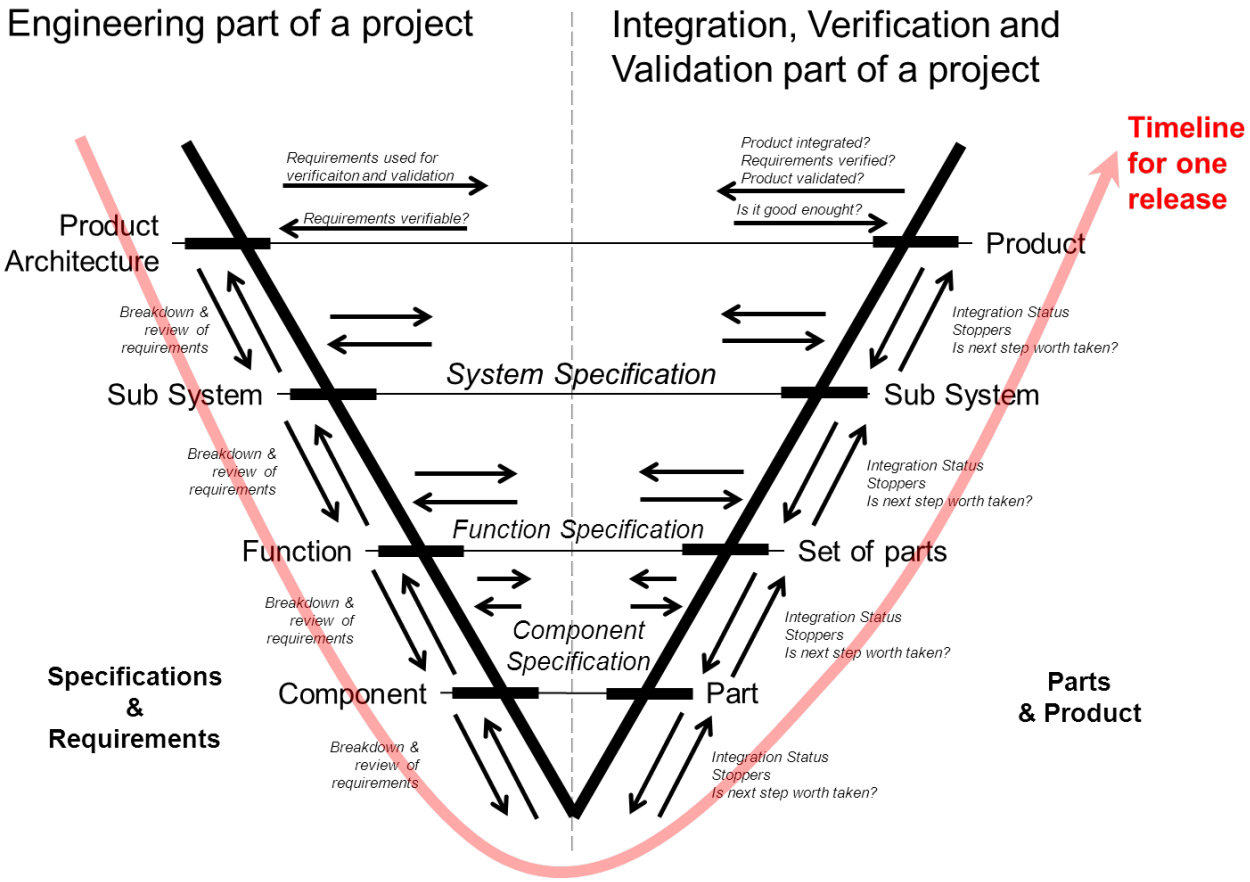
TFV mission is to correspond to several very demanding drivers. This means that TFV must test as wide as possible, as deep as possible and during different circumstances to cover all the different drivers in the world that will buy or use an AB Volvo truck. To do this in a structured way PVTs are performed. The information generated from each PVT is the reason why TFV exists.

1.3. Who are the PVT customers?

A PVT has two main customers, the ones ordering the test which usually is a project. The other customer is the end-customer of the truck, the buyer/user. Since this can be rather conflicting in interest it is important that TFV act as professional as possible when representing both. TFV's mission is to measure a truck's functionality from a driver perspective and address the issues the PVT find as accurate as possible. The main focus on all TFV activities is the **product**, not the project (AB Volvo is not selling projects to its customers). TFV must understand that there can be forces driving the project other than quality and functionality but TFV must also be very clear to their project customer that the driving force for a PVT is an absolute, accurate, objective and representative measurement of a product.

1.4. The V-cycle

The V-cycle is one of many different models explaining the interfaces and interactions when performing a project. Examples of other models are waterfall, incremental and Scrum. The main purpose with the V-model is to show that for every step in detailing the product specification, there is a corresponding step in verifying and validating the developed product. For every step in detailing the specifications the organization will add requirements and these requirements needs to be checked on corresponding level, since the cost in verifying requirements in a prototype is increased by factor 10 for each step.



A few requirements in "Product Architecture" will derive to multiple when passing through the design phase of a project. All these detailed requirements cannot be verified in the end product, it is not even possible for some of the detailed requirements. Requirements on a certain level need

to be verified on the same level in order to be able to work in parallel and synchronize deliveries (both specifications and parts). The level that is responsible for the verification can also secure that the requirements are verifiable when the parts are mature enough to be integrated in the next level. Having this level-approach when integrating the parts into a product makes it also easier to find stoppers and loop a re-delivery on local level instead of re-delivering the complete product.

TFV/PVT is working in the top of the verification/validation leg (right leg) with the product. This means that several other integration & verification activities must have been performed before the PVT can start.

2. Truck Functions

Truck functions and the testing of them is a wide concept. Not only the logical part of a function needs to be accounted for but also component quality, installation, environmental aspects, road conditions, fatigue in material etc.

2.1. What is a Truck Function?

Within AB Volvo, there are several definitions describing what a function is. The latest definition of a function from a PVT perspective is: ***-"A task, action or activity triggered by human, vehicle or environmental stimuli, that realizes or creates a result that is perceivable for, and fulfills a need of a user"*** [1]

The purpose of defining a function is to have a behavior description independent of technology for a wanted behavior. Whenever new technology is available to save cost or increase quality there is no new need of explaining the functionality. New or raised feature-level in functionality is however requiring an update in the functional description.

The growth in functionality in the past years and the accessibility of cheap microcontrollers makes it easy to assume that functionality is electrical and electronic components. This is wrong! Electronic components are one technical solution for the wanted behavior. Functions are component independent.

2.2. What is a distributed system?

At present, Electronic Control Units (ECU's) are realizing the wanted functionality and the trend is to add more software into the trucks to save cost, increase quality and raise feature-levels.

When ECU's first were introduced into the automotive it was to control the engines to make the internal combustion better, to decrease the amount of mechanical parts and to be able to present more information to the driver in the instrument cluster.

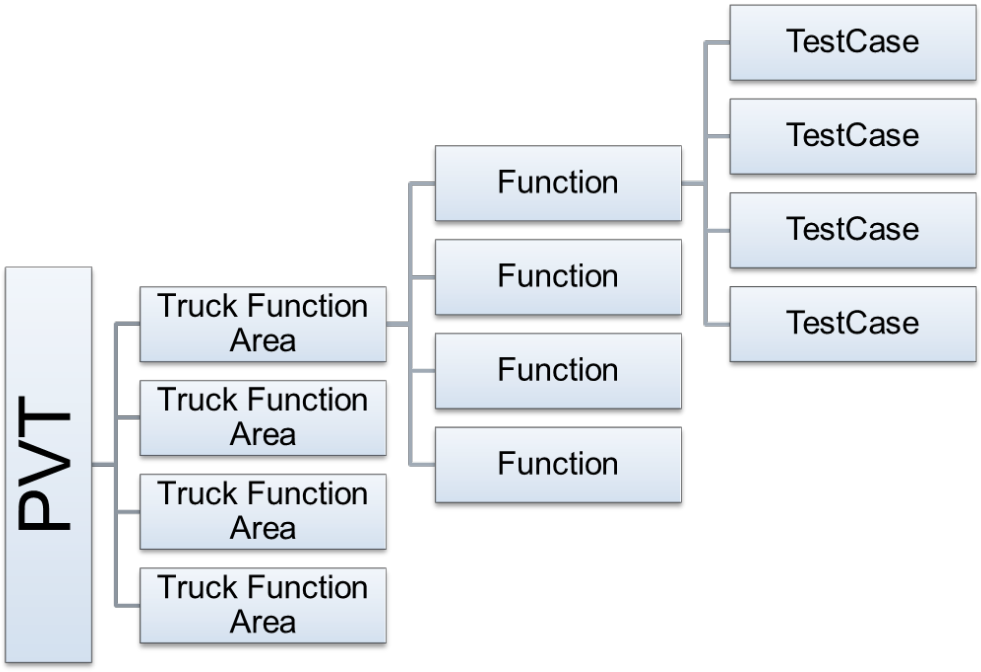
During time, the interaction increased and the complexity grew to add more features into the product or to save cost in exchanging expensive mechanical solutions. Functionality was grouped together to handle all sub-functionality like exterior light, electronically controlled suspension or bodybuilder.

In order to save raw material like copper in wiring harness and to be able to add more features into the product next step is to make a distributed system where the geographical allocation of the functionality is important. The "brain" of the functional system is no longer allocated in the cab but spread out in the complete truck to reduce the amount of wiring harness needed. Functionality that handles activities in the geographical front or rear of the truck is allocated together in one ECU instead of having multiple ECU's with part of the functionality. The logical rules of the functionality are more important to define since it is shared inside the distributed system.

[1] IFR = Integration of Functional Requirement project, IFR

2.3. How do TFV structure the Truck Functions?

There is no centralized, common way of expressing the functionality of a truck inside AB Volvo. The different engineering departments are working with functionality in a relative way; only what concerns them when defining the scope of a project or to handle the workload inside their departments. To be able to measure the product as absolute as possible, TFV has defined an own list, developed from experience and "current situation". It shouldn't be the testing organization defining the product, but this is the current situation since all other lists of functions are relative an organizations responsibility and workload and not harmonized centrally.
The most extensive list available when TFV started was EE department's list of E2EF [2].
TFV has further evolved this list into Truck Functions (TF) and categorizes these into Truck Function Areas (TFA).



The TFA's are mainly for categorization, to be able to handle all available and future truck functions. Even if the result of testing can be presented on TFA level there is for the moment no purpose with it. TFA are for the moment used for finding a function fast and for administrative purposes (generating statistics for evaluation of test objects).
The TF's are used for a variety of purposes; defining a trucks relative functionality content, categorize testcases, report results on and calculating projects test coverage in reference to used trucks. **An experienced driver should easily understand the purpose of the function when reading the Truck Function name**.
On each TF, one or several testcases are connected. These testcases have a description and an expected result which the requested test can connect a result on. The testcase can be generic (suits all available products) or specific (suits some Variants in the Variant Family). Functionality is normally not expressed in single Variants so logical rules and combinations has to be added to evaluate if a testcase is valid to be executed in a truck.

[2] E2EF = End To End Function

The test system structure is kept in a database environment and the structure supports traceability, analyses and evaluation of the data. Responsible for the structure is the PVT Truck Function Specialist.

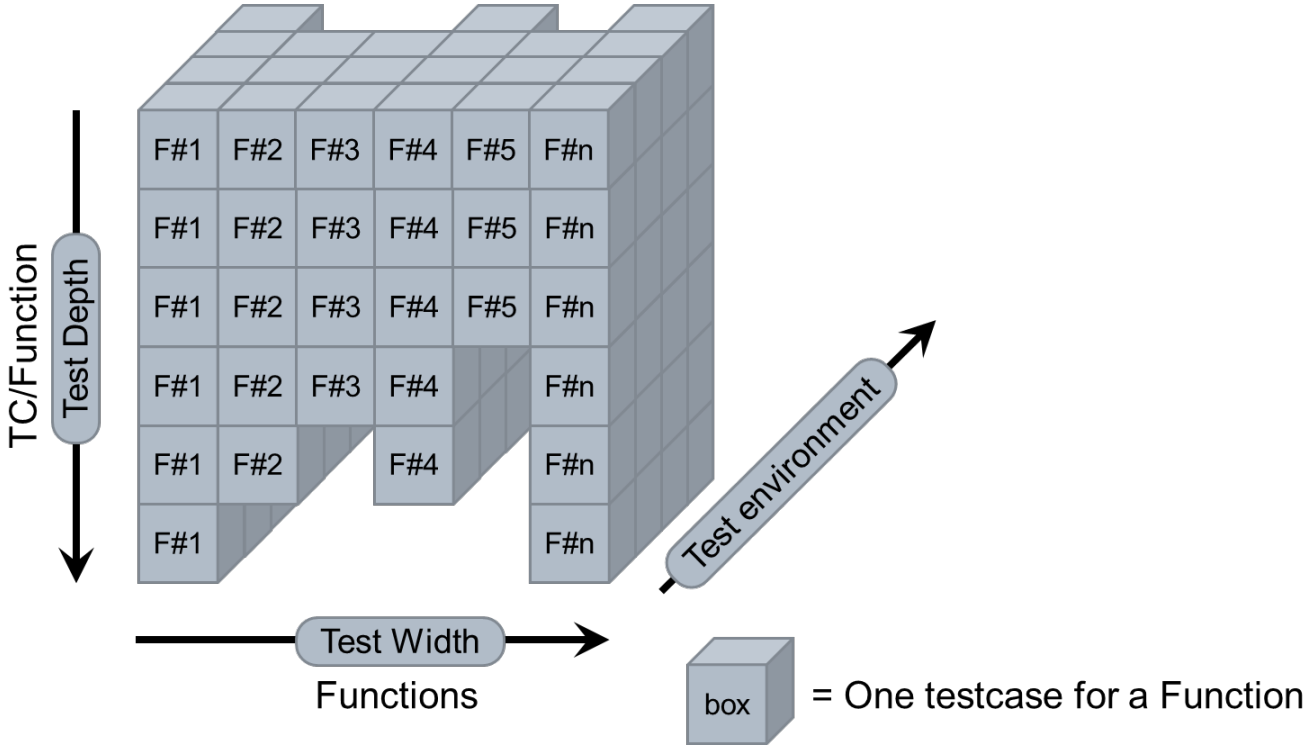
2.4. How do TFV test a Truck Function?

A Truck Function is covering all possible combinations of that particular function. Together with all Variants of a truck it is almost an endless possibility of combinations that affects the functionality. The workload of testing all available combinations is too heavy and would require a fleet of trucks not possible to handle. The combinations of Variants that are important are therefore stated in the testcases in order to cover the functionality.
On each TF there are several testcases connected and stated if there is a certain Variant combination required for that testcase. With this structure both function availability and all the variants of the function is covered. At present there are no centralized requirements on the functionality defined for complete truck so the testcases are developed from experience, known problems, Variant and Variant Families and requested testing from engineering departments (which in some cases defines functional requirements on a complete truck level).

A Truck Function is never sold alone to a customer without a complete content so TFV are always trying to test the complete width of the product, which is called **Test Width** and is defined as all Truck Function's available in one or several test object (One test object cannot cover all available Truck Functions so additional test objects are needed to get the best test width as possible).

TFV are using the testcases for each Truck Function to exercise, challenge and to create statistics that can be used for analysis and conclusions. The more testcases the test execute; the more challenged is the Truck Function. The number of possible testcases executed in each Truck Function is called **Test Depth**.

The 3rd dimension in testing a Truck Function is the environment factor. A Truck Function working as expected in sunny weather, in Swedish road conditions, with good batteries, immediately after workshop service or with a certain cargo can of course be affected and even malfunction by haze or fog, road conditions in Spain, poor batteries, high mileage or another cargo. This is called **Test Environment** and is handled through executing the same testcases in different environment.



The testcases for a Truck Function is a checklist to secure that the PVTs are covering all parts of the functionality and to generate as detailed raw data and statistics as possible. This is called sequential testing.
However, the test object is normally not presenting the intermittent errors in the system when the testcases are executed. To be able to handle the intermittent errors that occur during the sequel testing the possibility to generate OFD's (Other Faults during Driving) is available. OFD covers intermittent errors, unwanted behavior, exploratory testing and validation issues if not stated in the testcase.

2.5. Differences in Verification and Validation

TFV/PVT is covering both verification and validation aspects of the truck.
Verification: -***"Are we building the product right?"*** (I.e. as specified)
Validation: ***-"Are we building the right product?"*** (I.e. are the specifications right?)

Verification is to test and check if the specified functionality in the product is working.
Validation is to check if the specified functionality is something that the end-customer wants or expects.
There are no requirements on complete vehicle level that is tagged for functional verification so the verification is hard to perform with a formalized method.
If the company in the future adds requirements on truck functions these should be connected to one or several testcases but until then, the verification part that TFV perform is informal and based on input from the driver manual, experience and meetings with engineering representatives.
The validation part is a representation from an end-customer who is very demanding. When validating the truck TFV relies on the experience from the PVT drivers. They are the ones with the knowledge of how the trucks are used in reality. An engineer with a truck driving license and a little

imagination can perform validation in some aspect but can never replace the experience from a professional driver.

The mix of structured testing with testcases and the informal, exploratory testing performed with experience generates error reports, validation issues, statistics and the possibility to predict as **information** to the customer. Quality can never be tested to a good level; it can only be developed to a good level. In order for this to happen, information is needed and that is the main service that a Verification and Validation organization is providing.

3. Testcases – purpose and meaning

3.1. What is a testcase?

A testcase in PVT is a behavior-description for certain functionality with a description on the expected result.
Main pre-condition when executing a testcase is that the functionality is working (it has already been verified) and PVT evaluates how well the functionality works. If the functionality is not working, this needs of course also to be reported.

The purpose of the testcase is the following:

1. Keep track of the formal, sequential testing; making sure that no regression is implemented in the new release.
2. Evaluate if the expected result for a certain behavior is present (test/verification)
3. Evaluate if behavior is present, how well does it work. (validation)

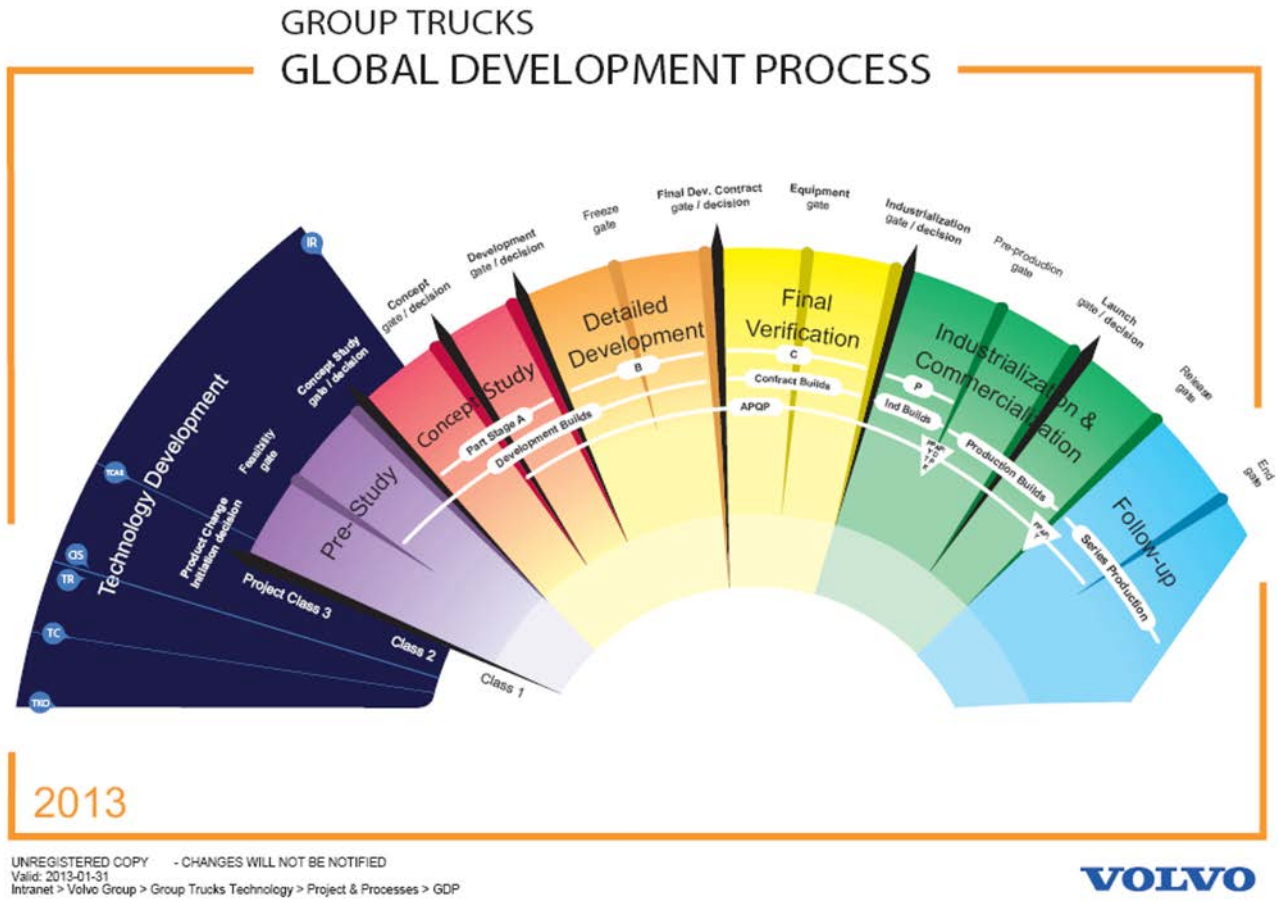
The testcase in a PVT is the smallest piece when building the test.
The testcase, after execution, is always having a result.

3.2. Developing and maintaining testcases

Each site has a designated Site Editor for testcase development. This role together with the PVT Developer, are handling all testcases, requesting new, updating old, sending for review, stating what productclass and/or Variant the different testcases should be sorted on and initialize a translation for other languages.
Documentation and description in the workflow are handled and explained in PVT Manager.

4. Performance Validation Tests in a project

All AB Volvo projects are following the Global Development Process (GDP). From the Pre-study phase to the Follow-up phase there are Builds (trucks) available.
Development Builds (a.k.a. Mules) are prototypes for verifying and/or validating concepts, installations, functionality, features, parts, services or aftermarket issues. They are normally hybrids of different technology and built for one particular purpose.
Contract Builds are the first prototypes of the complete scope of the project including all changed parts.
Industrialization Builds are the first builds manufactured in the factories where the different production lines have been tuned.
Production Builds are series of builds where the complete chain of logistics is tuned.
Series Production Builds is normal production builds.

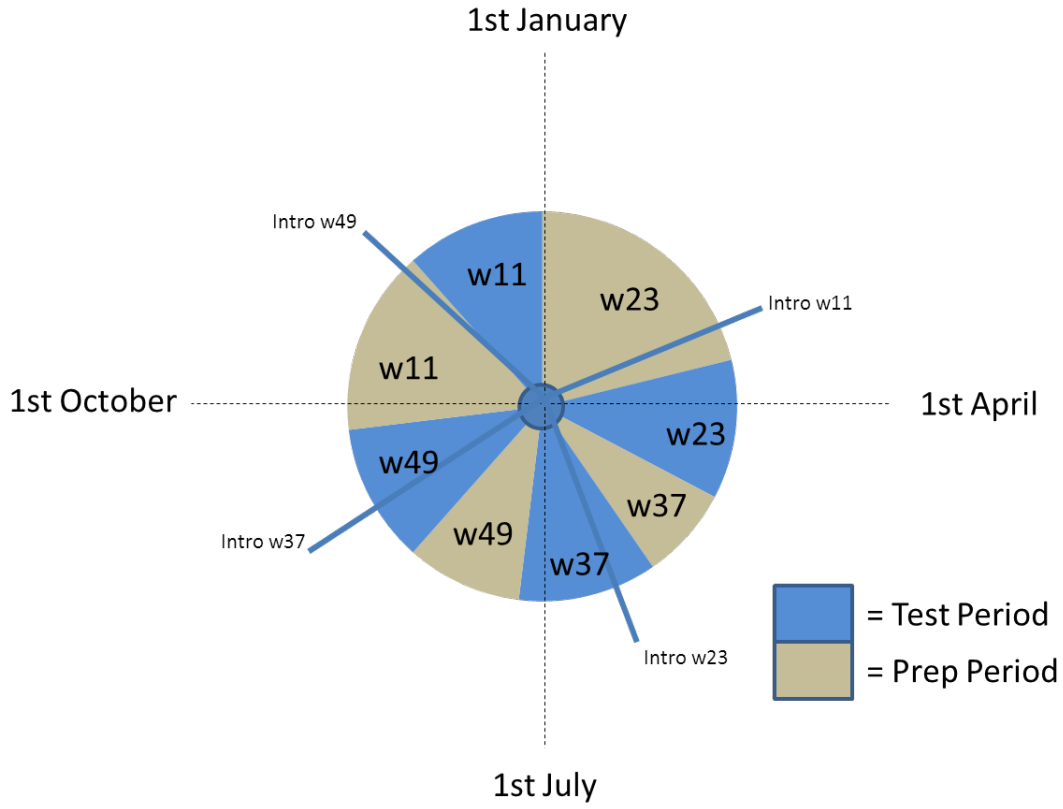


All these different kind of Builds can be tested with a PVT. If the project is developing functionality it is mandatory to perform PVT during the project lifeline.

4.1. SP-intro, projects and minimum PVT requirements on a project

The SP-Intros are introduced as 4 releases during a year instead of having individual releases for each project. The main purposes of introducing SP-intros is to prepare manufacturing for new DCN (Design Change Note, the way Volvo express a change in the current product), to have a synchronized and tested platform and to easier express changes in the product to each market.

These SP-intros affects TFV a lot and it is very important that TFV finds the pace to handle this. If not, control will be out of hand and TFV cannot secure the status of each intro. So when performing a PVT it is on a product towards a certain SP-intro and not a single project, especially late in the project time plan.
The minimum requirement on a project is at least to be a part of a PVT Total (for verification) and a PVT Europe (or similar for other markets) (for validation). The result of the testing is risk-management within the project and no particular minimum requirement is stated here.

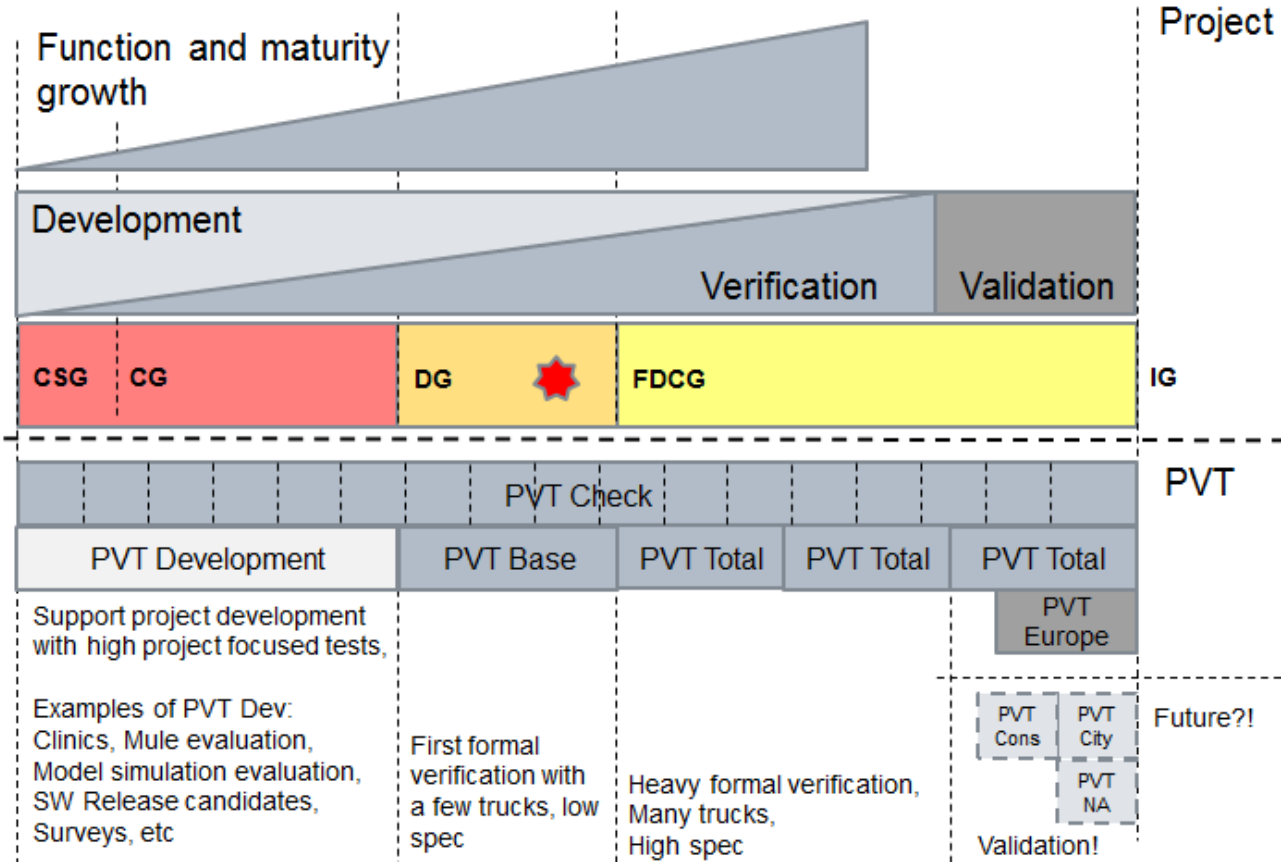


With the given Intro's from the company; w11, w23, w37 and w49 the Preparation and Testing periods can be visualized.

Intro	Notes
W11	Prep-period is long so additional activities can be performed (like educations, test updates, method development etc.) Test-period is during Christmas & New Year holidays and put high demands on planning. In northern hemisphere (v.v. southern) the climate is cold makes warm climate testing hard (e.g. high temp, rain).
W23	Prep-period is very long so additional activities can be performed (like educations, test updates, method development etc.) Test-period involves some holidays (Easter, Whitsunday etc.)
W37	Prep-period is short and planning needs to be accurate In northern hemisphere (v.v. southern) the climate is hot makes cold climate testing hard (e.g. low temp, icy roads etc.).
W49	Prep-period involves summer vacation, planning is crucial

4.2. Chain of rising tests

To get the most value for spent money and effort for a certain time in a project the PVT must be balanced. There is no need and a waste of money performing a high-cost validation test early in the project. In contrary, it can be counter-productive to execute tests that expect a mature product in an immature one. The engineering organization is not ready to handle the faults that are found since their focus is to further design the product instead of fine-tuning implemented design.



In the beginning of the project TFV experience and knowledge is used to verify and validate the project changes into the product and when the project changes are mature enough the PVT moves from development testing to formal verification and validation (somewhere around the red star). This change from “good cop” to “bad cop” when one day supporting development and finding solutions to the next day be annoying and demanding can cause some friction in the project organization. The communication between the customer and TFV is vital during this change and it is important that this change occurs, so the project isn’t stuck in development testing. This change is of course also important to have in mind on a personal level. Each role in TFV is one day supporting a project development, the next day representing the end-customer. Depending what personality the person in the role has, it can be an uncomfortable situation, so personal preparations is needed for the change.

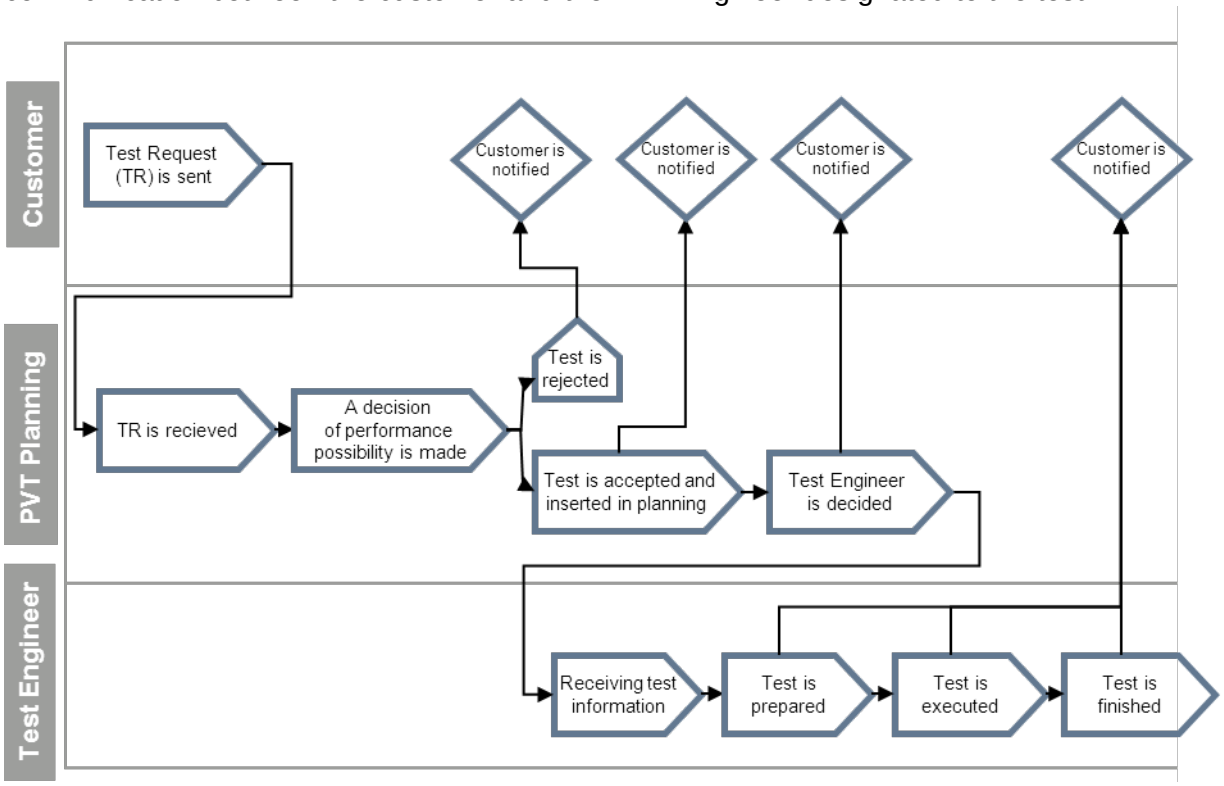
5. To run a Performance Validation Test

Independently of what test is executed the principles are the same.
1. Preparation phase
2. Execution phase
3. Conclusion phase

Any delays in each phase affect the next phase and normal behavior is to move each deadline forward in corresponding time. The customer must be informed.
A PVT does not catch up time from delays caused by others.

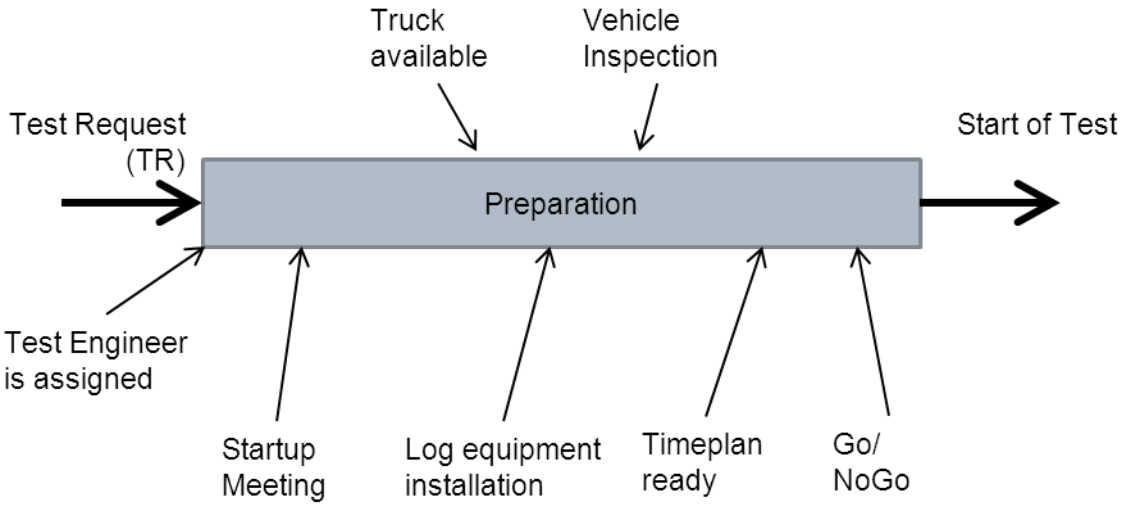
5.1. Test Request (TR)

The Test Request is the input to the test. This is a desire from a customer and makes it possible to handle resources available in the team. As it is a desire the TR will sometimes change during the communication between the customer and the PVT Engineer designated to the test.



TFV offer the customers a number of different tests which is easy to express in a Test Request (TR). These tests are standardized and part of the formal verification and validation TFV wants to perform before any project enters Serial Production. Customer has sometimes other needs; taskforce testing, limited budget or a sanity check in a vehicle early in the project phases and TFV then offer a PVT Development Test which is a completely custom-made test which puts a very high demand on how the TR is written. The TR is an agreement between the Customer and TFV Team.

5.2. Preparation phase



The Test Request (TR) is the foundation of the test, that's why it is important that it is clear, understandable and complete. If information is missing synchronization is needed between designated PVT Engineer and the Customer, especially if the test is a PVT Development Test. The minimum time from receiving the TR to test can start is stated in each test.
When designated PVT Engineer has reached a certain amount of control it is time for a startup meeting to synchronize all stakeholders.

5.2.1. Generic agenda for Start-up Meeting

- Intention for test – focus areas
- Deviations from standard test – Test Request
- Testcases - Information/discussion about required test cases and mileage collection.
- Truck - vehicle specification supports the test, time plan, equipment, updates, integration status, availability before and after test, loads and trailers etc.
- Request for logging equipment (CAN/LIN, temperatures, cameras etc.)
- Required routes/topology
- Required load weights and trailer specifications (Dymo, EBS etc).
- Reporting (if PROTUS, which RON's? Especially for maintenance-issues)
- Road Release
- Criteria's for stopping/aborting the test with consequences
- Other stakeholders that wants/needs to follow the test (other projects, Feature Leaders etc)
- Miscellaneous

The startup meeting will generate several actions before the test can start and the PVT Engineer needs to monitor these and check progress if they are solvable before the start of the test. Any deviation needs attention and any judgment that the test cannot start according to plan needs to be communicated to the stakeholders (especially the Customer).

Preparations can be divided into 3 categories

1. Test – TR special requests, testcode, drivers, troubleshooters etc.
2. Testobject (Vehicle) – Truck, Campaigns, HW/SW-versions, external equipment etc.
3. Log equipment - Breakout harness, M-loggers, Memorators etc.

Checklists are available here.

5.2.2. Go/NoGo

For every major step in the process to prepare and execute the test the PVT engineer should perform a sanity check if the test is worth the effort. This sanity check is called “Go/NoGo”. Several aspects should be taken into account; Product Maturity, Equipment, Resource availability, Value for the customers, test value etc.
Every possible mean should be used to turn a “NoGo” into a “Go” with kept test value for the customer. If a NoGo is un-avoidable the designated PVT engineer needs to explain why there is a NoGo present and present an alternative to the customer (with support from all stakeholders).
A Go/NoGo can be just a state of mind at the PVT Engineer, a discussion between stakeholders or a formal meeting. The degree of severity is controlling the need.
When sending expeditions outside the normal surroundings a Formal Go/NoGo meeting should be held for all concerned stakeholders.
Example of generic agenda if a formal meeting is needed:

1. Performed activities so far
2. Result of PVT Check and other findings
3. Driver’s opinion
4. Decision! Go? NoGo?

5.2.3. Preparation phase ready

When all preparations are made, all stakeholders are informed and “all systems Go!” the test can be started. This includes:

1. PVT Engineer and designated PVT Drivers (including Driver Teamleader) has an agreement of the plan
2. PVT Engineer has informed the manager of the test and resources used.
3. PVT Engineer and the customer are in sync

5.3. Execution phase

During execution the test object is untouchable!
The testobject is completely owned and controlled by the PVT Engineer. If the customer wants to use the truck for other purposes it is a new agreement between the customer and the PVT Engineer but normal behavior is to abort the test. This includes changing HW, SW or changing parameters. In some circumstances the test can be temporary stopped and started again with the delay time added to the deadlines.

The execution in sequence

1. Gather vehicle information and reset vehicle data e.g. vehicle mileage, engine clock and reset all Diagnostic Fault Codes (DTC’s)
2. Report start of test in different systems (Refiner, PROTUS diary etc)
3. Execute a PVT Check to find stoppers and get an overall status of the vehicle. Go/NoGo?

4. Collect Vehicle Data
5. Execute the next test-sequence
    a. Execute the test routines connected in calendar time (such as Daily inspection)
    b. Execute the test routines connected to the sequence (such as mini-expeditions).
6. Collect Vehicle Data
7. Repeat step 5 and 6 until all sequences are made.
8. Notify the customer that the test is done (but not closed).

During execution phase the PVT Engineer:

- Inform the PVT Drivers of important findings in other trucks.
- Is at least one step ahead in the planning and informs the drivers of the next steps
- Handles any deviations from the plan, e.g. work shop visits
- Gather information and log-files and writes PROTUS reports
- Keep track of important events and statements for later reporting
- Keep track of resources and budget and inform manager if anything deviates significantly from the plan.
- Report the status of the test, if requested by the customer
- …

During execution phase the PVT Driver:

- Report error, faults or issues when executing testcases or finding OFD's with the support of the logging equipment installed
- Informs the PVT Engineer of errors in vehicle, equipment or testcases
- Follow up any errors that needs further explanation with the PVT Engineer
- Performs Daily Inspections on the vehicle to find any errors
- Perform Weekly Inspections on the vehicle to find any errors
- Change trailer and load according to plan
- …

5.3.1. PVT Expeditions

Some of the tests performed involve sending a vehicle far away from the normal support organization, typically PVT Haulage and PVT Europe. This put a high demand on the preparation of these tests. Sending an expedition does not only stress the support organization but also add cost (personnel, hotels, ferries, tolls etc.) and the test value must always be considered to the cost. Sending expeditions are the final step in the product development progress and if the product is not mature enough there is no need of an expedition. The inconvenience being abroad with a company-classified vehicle should not be underestimated and the support from the PVT Engineer to the PVT drivers is vital. Whenever planning an expedition it should be natural to first check the functionality and the maturity at in normal test conditions, then have a formal Go/NoGo-meeting with the stakeholders (especially the drivers that will perform the expedition) and decide if the effort is worthwhile or if the risk and cost is too high.

5.4. Conclusion phase

When every planned and adapted activity is executed and all needed data is collected (but not processed) the test is done. Now the vehicle can be used for other purposes including made

available for engineering departments for fault tracing (depending of agreement with the customer in the TR).
The conclusion phase includes the following activities:

- Finalize reporting and generate the report that is stated in the TR. (if ER is requested, request a registration number)
- Debrief the PVT Drivers on feedback on test execution, testcases, vehicle status, planning, lessons learned etc. This requires some sort of documentation. Until a system for this is available, this must be handled manually by every test engineer. Documentation must be kept centrally. TBD: PVT Knowledge & Experience Management
- Handle the equipment from the test (uninstall logging equipment etc)
- Deliver the vehicle to the next receiver according to the TR.
- Provide support to fault tracing to engineer if vehicle is kept inside the TFV organization.
- Address issues in testcases and suggestions of new ones to the PVT Test Developer
- Address other issues and educational need to the PVT Global Coordinator
- Report Test Closed to the customer.

5.4.1. Generic agenda for Debriefing meeting

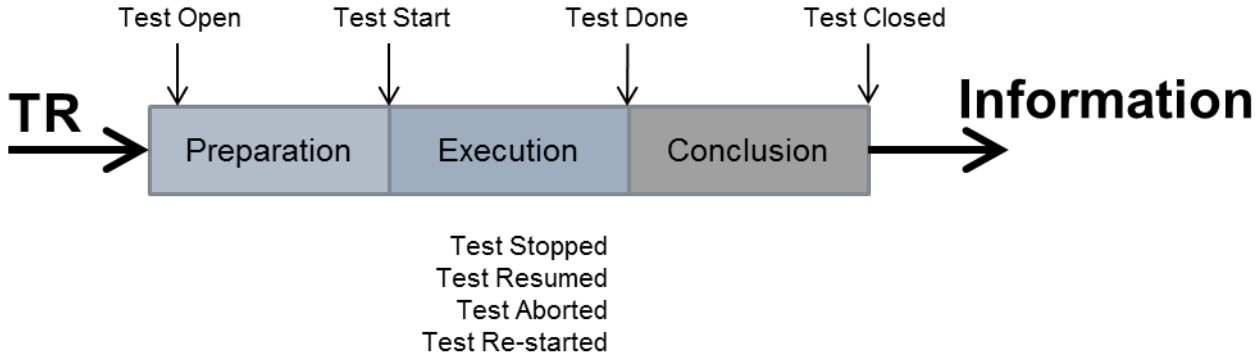
Testobject evaluation

- Testcase walkthrough – Every TC has a result, no questions on “comments” in testprotocol.
- Testcase inconsistency – Any inconsistent result? Reason? Document why. Compile result.
- Most severe issues – Supports “Italian flag” and Conclusion
- Overall status of the truck – Supports “Italian flag” and Conclusion
- Conclusion - What is the conclusion of the test that needs to be stated in the report?

Test evaluation

- How has the work proceeded?
- How has the information been handled?
- What is the “lessons learned” from this test?
- What is the individual learning’s from this test?

5.5. State-machine and nomenclature



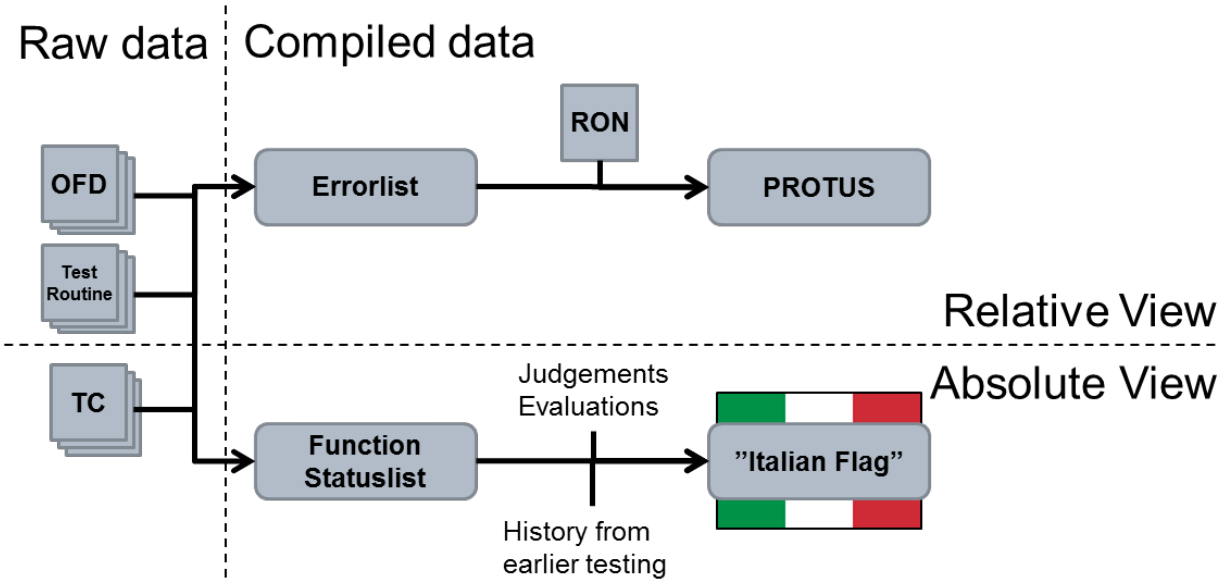
Test	Definition
Open	Designated PVT Engineer has started to work with preparations. Cost and workload is generated.
Start	All preparations made and testing activities are started
Done	All testing activities are done and no more testing is planned
Closed	All activities are finished and no more work or cost is expected.
Stopped	All testing activities are temporary stopped to evaluate the situation.
Resumed	All testing resumed after a stop
Aborted	All testing activities are cancelled. New TR is required.
Re-Started	The test is for some reason re-started (from the beginning).

6. Results

Result is generated in 3 different ways during the execution of a PVT.

1. Testcase - TC – Formal, structured functional testing with a Description and an Expected result.
2. Other Faults during Driving - OFD – All other faults detected during the test.
3. Testroutine – Formal activity or routine other than a TC, e.g. DTC readout, Daily inspection etc.

These 3 containers of information are the foundation in the result and the raw data that is generated during a test. This raw data must then be processed, compiled and analyzed for different views and conclusions. The raw data must be kept in its origin to be able to compare and re-use it later when new methods and tools are available.



6.1. Results from testcases

When a testcase has been executed it can have the following result:

Result	Description	Comments
OK	The expected result was fulfilled	No explanation needed. If small remarks or deviations are present, these must be commented.
NOK	The expected result was not fulfilled.	Explanation needed. Examples: What is not working or what is not working so well? Link to Problem report.
NA	The testcase is not possible to execute.	Explanation needed. Examples: TC not supported in vehiclespec, Function turned off in vehicle, Special request in the Test Request etc
NT	The testcase was not tested	Explanation needed. Examples: Dangerous testing, lack of time, no knowledge how to execute the testcase etc

Except the result itself, the signature of the executer must also be documented for later evaluation of the test-result.

6.2. Other Faults during Driving (OFD)

OFD (other Faults during Driving) is all faults detected when not executing a testcase or a testroutine. It could be a pop-up appearing when it is not supposed to or any deep digging in particular functionality.

6.3. Result from Testroutines

A Testroutine should always have a result. It could be as simple as “Performed” (like a “DTC readout”) or a complete checklist with different results to be followed (like a “Sleep-over”).
These are not necessary to report in detail when writing the formal report (e.g. ER) but we need to keep track of them internally.

6.4. Raw test-data to compiled test-data

The raw testdata (Result from- Testcase, -OFD’s and -Testroutine’s) created during testing needs to be analyzed and compiled after each test. The data can be contradictory and additional issues can be found during this analyze. The compilation is also a way to evaluate the different areas within the test (organization, work performance, interactions between personnel, customer satisfaction, Test Request fulfillment etc).
The compilation is preferable done in a system to be able to document the conclusions made, but until the system support this, manual compilation is required (e.g. Excel).
The responsibility of the compilation is the test engineer.

6.5. “Italian flag” and other judgments/evaluations

The compiled data is the result of the test and satisfies the test request. But projects do normally have several tests within its time plan and the project management is interesting in trends, evaluations and judgments between the different tests. 2 approaches are mentioned below:

1. Comparison testcase by testcase. Pretty strait forward.
   Pros: Easy. Cons: data is not compiled, hard to read, can be misunderstood if tests have been performed in to different test objects.
2. “Italian flag”, 3 areas of green white and red. Green is problems that has been fixed or showing progress. White is smaller issues or issues with no progress. Red are severe problems and/or new problems.
   Pros: Understandable. Cons: Subjective judgment. Things can fall between.

“Italian flag” has proven to be a good reporting/evaluation method, appreciated by the project management and it is recommended to continue to perform this when several tests are planned. It is also recommended to back up this evaluation with test statistics, but to do this in an efficient way; the result needs to be stored in a database structure. Until then, this recommendation is not mandatory.

6.6. Status reports

6.7. PROTUS reports & RON

PROTUS is the formal faultreporting system. Even though it is in some ways a stiff and obsolete system, it is widely used and there is a complete organization handling the PROTUSs written.
Erratic behavior and missing functionality is reported as L-PROTUSs.
Validation issues are reported as P-PROTUSs.
The PROTUS-system keeps track of all faults in the truck that is addressed to the engineering departments. Even if the faults are reported as symptoms it is vital that we provide as much information as possible when writing PROTUS-reports.
To be able to write a PROTUS-report, a RON (Responsible Object Number) must be associated.
The different RON's for each test must be provided from the Test Requester before the test starts.
This includes a maintenance RON where non-project concerned issues can be addressed.

6.8. Engineering Report (ER)

4 weeks after Test Done
Important events during test execution

6.9. The different view possible

???

7. Vehicle & Variants

7.1. Product Classes

7.2. Variants

7.3. Trailers & Dolly's

Payload

7.4. Inspections

Delivery Inspection
Daily Inspection
Weekly Inspection

7.5. Road Release & Road Approval

8. KPI's

KPI Stakeholders

8.1. KPI 1 - The number of performed test cases for different tests

8.2. KPI 2 - Deliveries on time – measured with ER to be written within four weeks after test finishes

8.3. KPI 3 - The number of status 21 PROTUS reports caused by TFV team

9. PVT Tools

9.1. E-FACTS

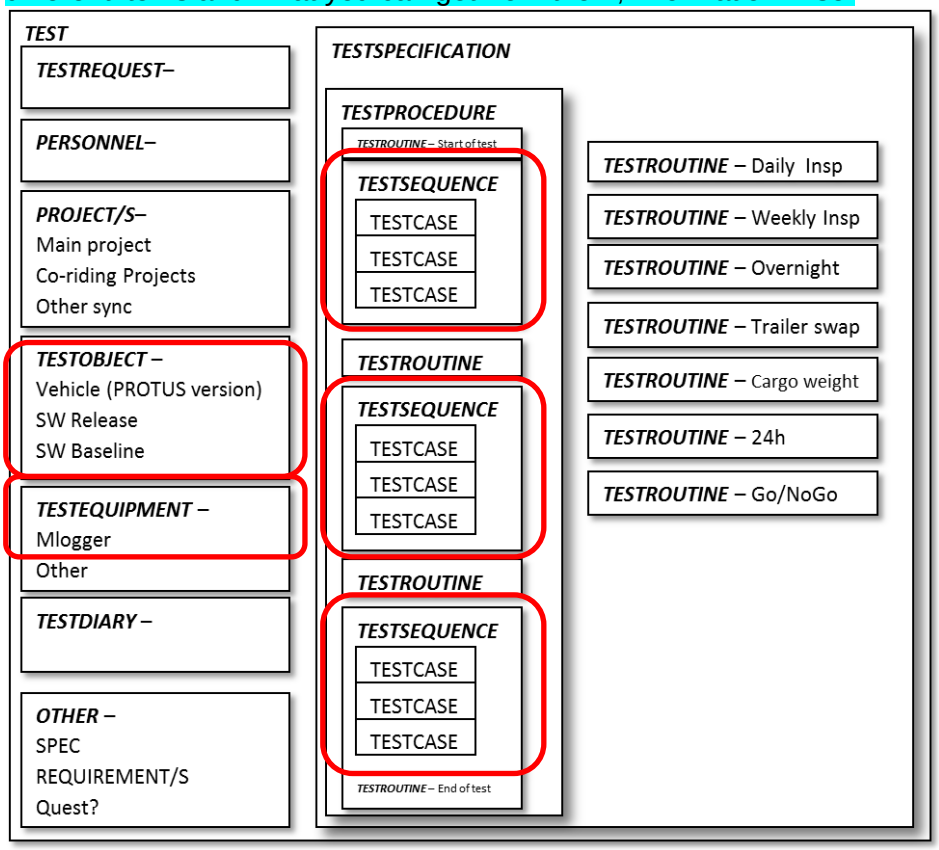
9.1.1. Log equipment

9.2. Tools during testing

9.3. Other tools

10. Meta-model of information

A Meta-model is an explanation how information is connected to each other. Information, such as result of a testcase execution, is connected to that testcase. It is also connected to the testobject it has been executed in as well as what test has been executed and the driver that has executed it. All this information (both direct and indirect) is connected to each other and the Meta-model is explaining how. These bricks of information that are connected to each other are called Items. On a deeper, more detailed level the Meta model also explains the dependencies between the different items and what you can get from them, information-wise.



Name	Description
TEST	The overall container for every planned and executed activitites including the result
TESTSPECIFICATION	The container for the activities in the TEST. Contains both sequential and non-sequential activitites
TESTPROCEDURE	The container for all sequential activitites in a TESTSPECIFICATION, Needs to be executed in chronological order.
TESTPROTOCOL	The result of a TEST
TESTSEQUENCE	Set of testcases, Are default sorted but can be sorted individual as well.
TESTROUTINE	A generic container for every planned activity that is not a testcase/testsequence. Could be time-dependent(Such as Daily inspections) or independent (Such as Test start routine or Start Inspection)
TESTCASE	Description, Expected Result and attributes collected to one container, possible to generate structure and statistics
TESTREQUEST	Information from the customer, a desire to perform a TEST. Can be updated during the preparation of the TEST
TESTDIARY	A timebased notepad where all non-planned activities can be documented. Used for personal use, evaluation, knowledge management and formal reporting

The red squares explain what we currently have in our system. Development is ongoing.

11. PROTUS Verification

12. Appendix

12.1. Abbreviation & Terminology

The Volvo Group Wiki for abbreviations & terminology can be found here: link
Unique TFV/PVT abbreviations & terminology in table below:

TBD	

12.2. Stakeholders during PVT (updated 2014-05-14)

Role	GOT	LYS	GSO	CUR	AGO
Mandatory					
Customer	Test Requester	Test Requester	Test Requester	Test Requester	Test Requester
Log equipment	Martin Svennungsson	Jean-Philippe Abeillon	Jeff Hairston	Andre Moscheto	Masahiko Hoshino
PVT Planning	Henrik Biswanger	Julien Rouge	Tim Jarvis	Andre Moscheto	Kouichi Hirata and Hideo Takahashi
Driver planning	Magnus Ludvigsson	Patrick Jacquet	N/A	Andre Moscheto	DRD
Optional					
Testcases	Johan Hallberg	Sylvain Galléa	Roscoe Luce/ Tim Jarvis	Andre Moscheto	Masahiko Hoshino
Site editor for TC's	Tobias Hallor	Sylvain Galléa	Roscoe Luce	Andre Moscheto	Masahiko Hoshino
Methods	Johan Adoson	Julien Rouge	Tim Jarvis/ Roscoe Luce	Andre Moscheto	Masahiko Hoshino
Tools	Johan Adoson	Jean-Philippe Abeillon	Tim Jarvis	Andre Moscheto	Masahiko Hoshino
Log installation	Fredrik Svensson/ Anders Jellbin	Paul Pereira/ Laurent Sanchiz	Tim Jarvis/ Roscoe Luce	Andre Moscheto	Masahiko Hoshino/ DRD
Troubleshooter	Tim Jansson	Thibault Thomas	Roscoe Luce	Andre Moscheto	Hideo Takahashi/ Masahiko Hoshino
PVT Drivers		Jérôme Silvestrin	N/A		DRD
KPI #1 responsible	Johan Adoson		Roscoe Luce		
KPI #2 responsible	Tomas Larsson		Tim Jarvis		
KPI #3 responsible	Tim Jansson		Tim Jarvis		

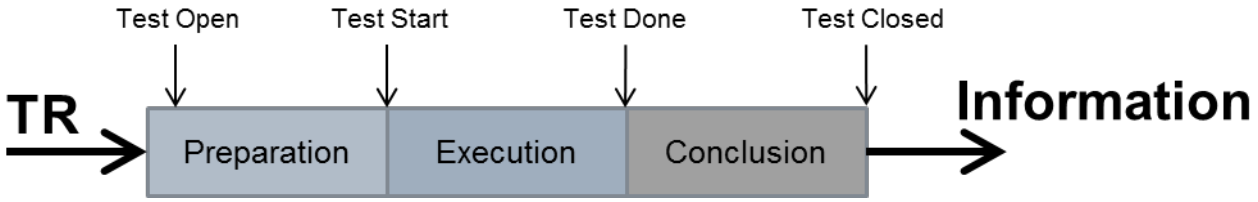
12.4. PVT Current tests

The cost for each PVT is under synchronization and development

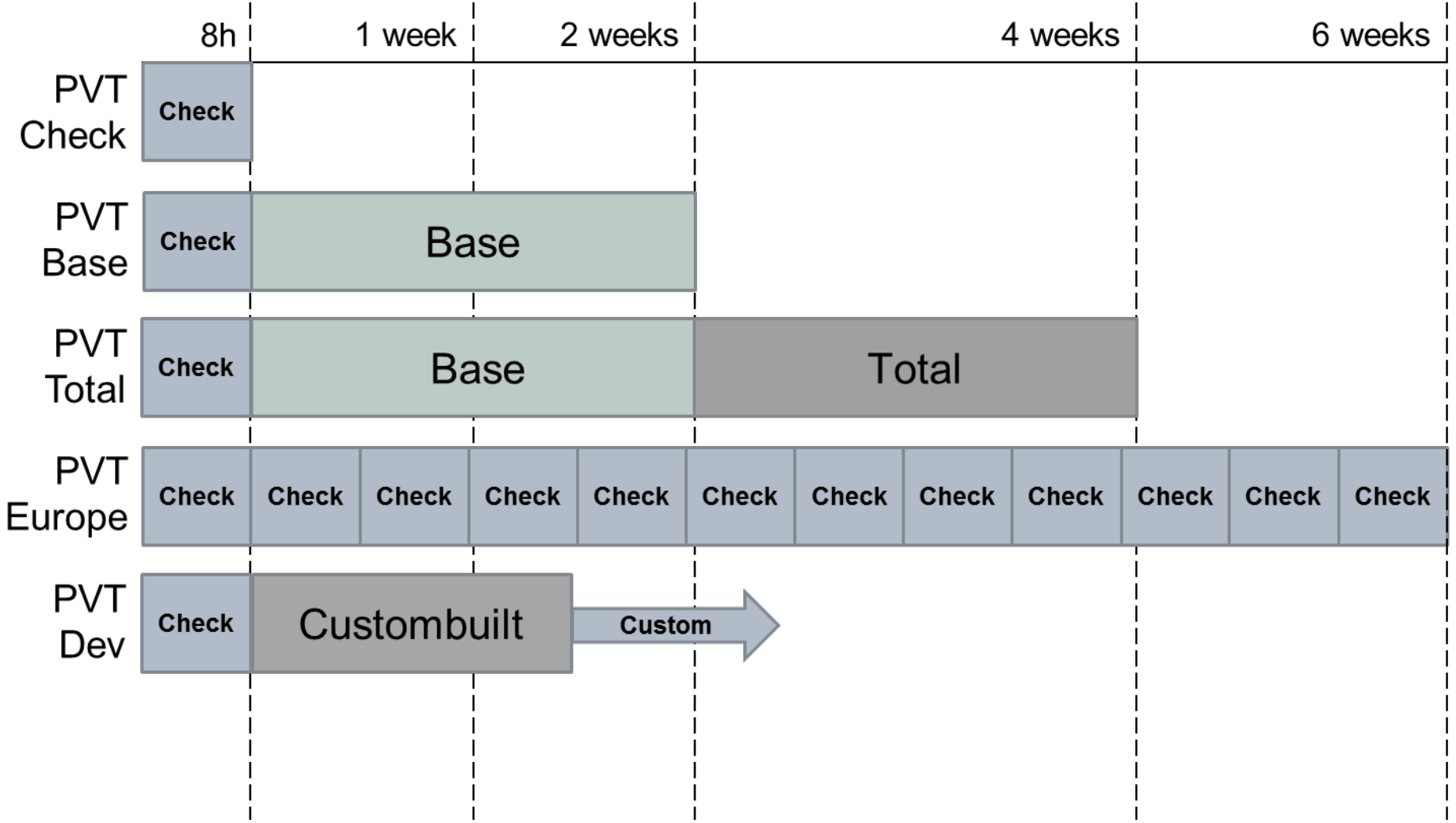
Current tests available are:

Test	Type	Description	Preparation Time **Test Open** to **Test Start**	Execution time **Test Start** to **Test Done***	Conclusion Time **Test Done** to **Test Closed**	Total Time **Test Open** to **Test Closed**	Driver shifts	Report	Comments
PVT Check	Delivery Control	Finding stoppers in all available functions.	2days	8h*			One shift used	Mail + testprotocol in PROTUS	
PVT Base	Verification	Normal usage with 1-driver system and no sleepover.	4w	2w*	4w		2-shift used	ER + PROTUS + Mail	2 drivers during one shift, 1 driver during the other shift. (this is for lead-time saving) The testcode is run twice.
PVT Total	Verification	Extended usage incl 2-driver system and sleepovers.	5w	4w*	4w	13w	2-shift used	ER + PROTUS + Mail	
PVT Haulage	Validation		6w	6w*	4w		1-shift used	ER + PROTUS + Mail	
PVT Europe	Validation		6w	6w*	4w		2 driver system	ER + PROTUS + Mail	
PVT Development	Development test		custom	custom	custom	custom	custom	Custom	

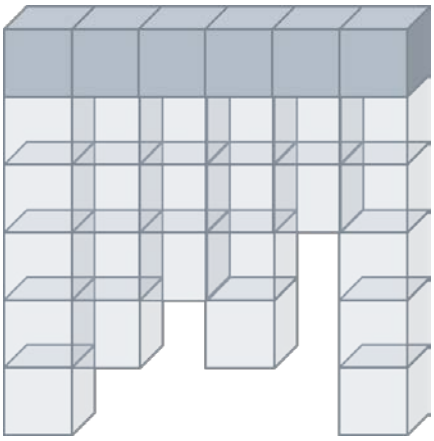
*These figures are reference figures when executing a test with the mentioned driver set-up and a high-spec vehicle. These figures have been empirically being evaluated during long time. If another driver set-up is used, consider both consequences in leadtime as well as test quality in the result. It is the activities in each test that finally determines the needed time. And of course, Many issues = more time needed.



12.5. Execution time and principles



12.5.1. PVT Check



Purpose:
A PVT Check is a delivery control of a truck. Every function available in the truck is checked (but not tested, verified or validated) in order to find any stoppers. This test is used as a checkpoint before digging any further into functionality testing or other tests.

The test requires a very experienced driver to generate fast and accurate results with the discipline not to dig into deep.

The PVT Check is also used for regression testing, if any added functionality has affected the already present functionality.

The PVT Check does not require a Road Release (it is a precondition for RR) but requires a test-track with the ability to drive in high speed, has slopes, gravel road or similar non-friction roads.

Project status: Performed in all phases; as a sanity check, as a delivery control or as a foundation for other PVT's.

Driver focus:
100% of their time in executing testcases

Sequences and Special tests:

1. Collect Vehicle Data and reset testobject (SW/HW, Parameters and DTC's)
2. Perform a PVT Check testsequence.

At Test Done: Collect Vehicle Data and reset testobject.

A "**PVT Check testsequence**" is a set of testcases administrated in PVT Manager. Typically one testcase/Truck Function is designated in a PVT Check. The testcase is on very high level to check if the expected functionality it available or not.

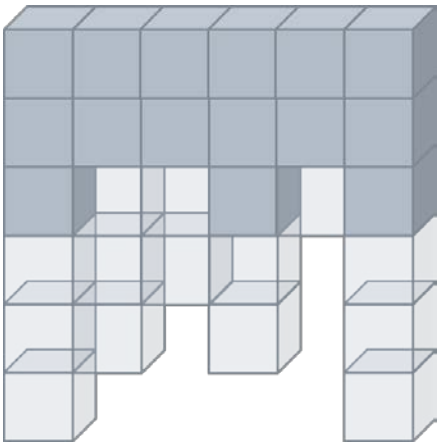
Logging Equipment:
Internal PVT Check requires an M-logger, since the PVT Check is a part of ALL other testing.
External PVT Check does not require an M-logger, but this should be stated in the TR.

Reporting:
Normal reporting after (no reporting during execution)) a PVT Check is,

- Status report in PROTUS
- Status report (in a mail) to the Customer
- All testdata stored according to local agreement (but not on individual HDD's!) ➔ Longterm: Stored in test system.

Vehicle and driver equipment:
Driver bag (see checklist for inventory.)
Special test equipment for PVT Check (connected to each testcase)

12.5.2. PVT Base



Purpose:
A PVT Base verifies and validates all available functions during all normal usage with one driver (but could be executed with 2 to cut leadtime).

The PVT Base is performed in the surroundings of a proving ground and requires a Road Release to be able to drive on public roads.

A PVT Base is a Verification test.

Project status: Performed after C-release when maturity is expected to be high and only minor issues are present in each part.

Driver focus:
½ - ¾ of their time in executing testcases
½ - ¼ of their time in exploratory testing (OFD's)
Testcase progress monitored by the test engineer

Sequences and Special tests:

1. Collect Vehicle Data and reset testobject (SW/HW, Parameters and DTC's)
2. Perform a PVT Check testsequence
3. Collect Vehicle Data (Parameter and DTC's)
4. Go/NoGo
5. Perform a PVT Base testsequence (loop#1)
6. Collect Vehicle Data (Parameter and DTC's)
7. Perform a PVT Base testsequence (loop#2)

At Test Done: Collect Vehicle Data and reset testobject

Daily inspections performed by each driver,
Weekly Inspections performed

"**PVT Base testsequence**" is a number of testcases administrated in PVT Manager where functions are tested as deep as possible from a driver perspective.

Logging Equipment:
Normal logging equipment during a PVT Base is a standard setup of M-logger, logging all CAN-links.
Log files are collected each shift by the PVT driver and handed over to the PVT engineer.
Testcase report is stored and updated in a laptop.

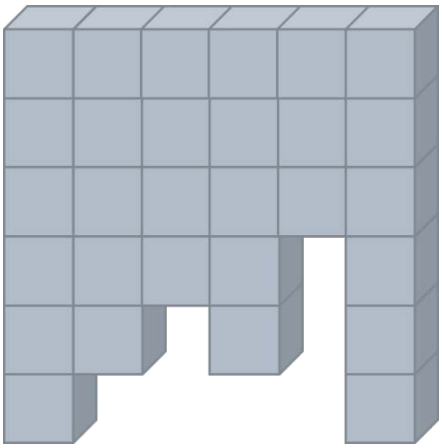
Reporting:
Normal reporting during and after a PVT Base is,

- Status report every week to the Customer

- PROTUS reports written as soon as possible
- ER report finalized and distributed 4 weeks after ***Test Done***
- All testdata stored according to local agreement (but not on individual HDD's!) ➔ Longterm: Stored in test system.

Vehicle and driver equipment:
Driver bag (see checklist for inventory.)
Special test equipment for PVT Base (connected to each testcase)

12.5.3. PVT Total



Purpose:
A PVT Total digs as deep as possible in every available TF in the truck. This means executing every testcase available. The PVT Total is the most thorough test available and includes challenging and exercising the system hard. Even fault injections are performed in areas important to the driver. The PVT Total is performed in the surroundings of a proving ground and requires a Road Release to be able to drive on public roads.

A PVT Total is a Verification test.

Project status:
Performed after C-release when maturity is expected to be high and only minor issues are present in each part.

Driver focus:
½ - ¾ of their time in executing testcases
½ - ¼ of their time in exploratory testing (OFD's)
Testcase progress monitored by the test engineer

Sequences and Special tests:

1. Collect Vehicle Data and reset testobject (SW/HW, Parameters and DTC's)
2. Perform a PVT Check testsequence
3. Collect Vehicle Data (Parameter and DTC's)
4. Go/NoGo
5. Perform a PVT Base testsequence (loop#1)
6. Collect Vehicle Data (Parameter and DTC's)
7. Perform a PVT Base testsequence (loop#2)
8. Collect Vehicle Data (Parameter and DTC's)
9. Go/NoGo
10. Perform a PVT Total (normal) testsequence
11. Collect Vehicle Data (Parameter and DTC's). This is the formal DTC list reported.
12. Perform a PVT Total (extended) testsequence

At Test Done: Collect Vehicle Data and reset testobject

Daily inspections performed by each driver,
Weekly Inspections performed
2-driver system & Sleepover (Miniexpedition) in step 10(normal) or in step 12(backup).

"**PVT Total (normal) testsequence**" is a number of testcases administrated in PVT Manager where functions are tested as deep as possible for a driver perspective. These testcases are provoking the functionality but does not jinx or fault inject the truck system. No abnormal DTC's or behavior is expected.

"**PVT Total (extended) testsequence**" is a number of testcases administrated in PVT Manager where functions are jinxed and injected with faults as well as parameter-change. DTC as well as severe pop-ups are expected. This test will affect the DTC-list for the test so it is important to fetch that list before!

Logging Equipment:
Normal logging equipment during a PVT Total is a standard setup of M-logger, logging all CAN-links.
Log files are collected each shift by the PVT driver and handed over to the PVT engineer.
Testcase report is stored and updated in a laptop.

Reporting:
Normal reporting during and after a PVT Total is,

- Status report every week to the Customer
- PROTUS reports written as soon as possible
- ER report finalized and distributed 4 weeks after ***Test Done***
- All testdata stored according to local agreement (but not on individual HDD's!) ➔ Longterm: Stored in test system.

Vehicle and driver equipment:
Sleep over equipment for designated drivers.
Driver bag (see checklist for inventory.)
Special test equipment for PVT Total (connected to each testcase)

12.5.4. PVT Europe

12.5.5. PVT Haulage

12.5.6. PVT Development

12.6. Checklists

LINK TO EXCEL

12.7. Current Function List and Variant Families effected

TBD

12.8. Revision History

Release	Date	Comment	Author
0.1	2013-03-06	Document created	Johan Adoson
0.2	2013-04-08	First review. Not yet developed areas marked with yellow	Johan Adoson
1.0	2013-04-25	After first review. Updates of structure, spelling, order of word and grammar.	Johan Adoson
1.1	2014-05-14	New and updated information marked with color. Section “Result” detailed Check, Base and Total further detailed Metamodel added	Johan Adoson