

3mdeb software services

Regression Testing Solutions



3MDEB

ul. Burgaska 9D/10
80-287 Gdańsk, Poland
TAX ID: PL2530164147

CONTACT

Piotr Król, 3mdeb CEO
+48 880 673 344
contact@3mdeb.com



[Website](#)



[About Us](#)



Table of Contents

1. [Service description](#)
2. [Key features](#)
3. [Service options](#)
 - [Full regression testing](#)
 - [Custom regression testing](#)
 - [One-time regression testing](#)
 - [Time-based regression testing](#)
4. [Specification](#)
 - [QA technologies](#)
 - [Regression test suite tree](#)
 - [Spreadsheet report](#)
 - [HTML log files](#)
 - [Bug management](#)



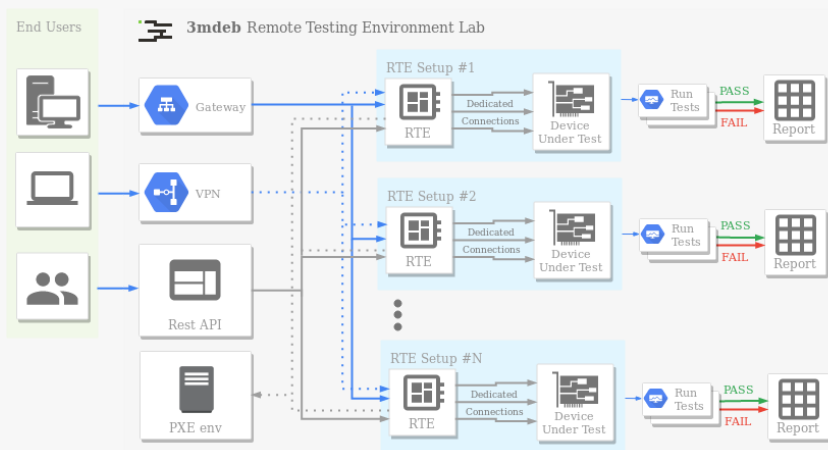
Service description

Regression testing is a concept of validation practice which brings the answer to one of the most frequently asked questions:

Does the new revision of the product is working as expected?

With regression testing, we can confirm, that a recent version of the hardware and/or embedded firmware has not adversely affected existing features.

Regression Testing Architecture



Regression testing is required whenever:

- hardware schematics have been changed
- embedded firmware code has been modified
- new features have been implemented
- there is a change in the embedded firmware framework version (for example coreboot version)
- there is a change in the binary components that are part of the system, but are developed by an external company (for example FSP)



Key features

- Fully automated, fast and accurate validation
- Rigorous dependency analysis (requirements vs. test-cases)
- Full (over 100 tests) or partial (on clients demand) testing package
- RobotFramework infrastructure
- Detailed HTML logs and spreadsheet reports attached
- Bug management support
- Time comparison summaries*
- Optional CI integration
- Affordable pricing

* available only with time-based regression service

```
CBMEM1.1 Coreboot booting time after warmboot
.
(0) Coreboot booting time: 2.571331 s (2571331.0 ns)
(1) Coreboot booting time: 2.569106 s (2569106.0 ns)
(2) Coreboot booting time: 2.566676 s (2566676.0 ns)
(3) Coreboot booting time: 2.570862 s (2570862.0 ns)
(4) Coreboot booting time: 2.568222 s (2568222.0 ns)
...
Coreboot average booting time: 2.5692394 s

CBMEM1.1 Coreboot booting time after warmboot | PASS |
-----
CBMEM1.2 Coreboot booting time after system reboot
...
(0) Coreboot booting time: 2.568462 s (2568462.0 ns)
(1) Coreboot booting time: 2.528046 s (2528046.0 ns)
(2) Coreboot booting time: 2.521855 s (2521855.0 ns)
(3) Coreboot booting time: 2.521665 s (2521665.0 ns)
(4) Coreboot booting time: 2.517512 s (2517512.0 ns)
...
Coreboot average booting time: 2.531508 s

CBMEM1.2 Coreboot booting time after system reboot | PASS |
-----
Performance.Cbmem | PASS |
3 critical tests, 3 passed, 0 failed, 0 skipped
3 tests total, 3 passed, 0 failed, 0 skipped
=====
Performance.kernel-load-pfSense
=====
KLD1.0 ATA UDMA coldboot test | SKIP |
Platform doesn't have (m)SATA with installed pfSense.
-----
KLD1.1 ATA UDMA warmboot test | SKIP |
Platform doesn't have (m)SATA with installed pfSense.
-----
KLD1.2 Get disk and system info | SKIP |
Platform doesn't have (m)SATA with installed pfSense.
```



Service options

Client is obligated to choose the type of regression testing:

- [Full](#) or [Custom](#)

and the period of time for validation:

- [One-time](#) or [Time-based](#)

receiving a customized validation package.

Price:

The fee is proportional to the selected portion of the regression suite and depends on the quantity and complexity of the requirements of new tests - the price should be determined by contacting the company.



Service options

Full regression testing

OSFV regression suite testing service is a full automation tests package offered by the validation team. The full suite consists of 118 tests, but the regression suite database is still growing and future tests will be also included in this package. Tests cover respectively hardware and embedded firmware related features listed in the [specification](#) section.

* certain test-cases requires special features enabled, which could be not supported by clients hardware or embedded firmware.

Custom regression testing

Instead of re-executing the entire regression tests suite, clients can customize regression test suite for their needs, creating a personalized testing package:

- selecting portion of the full regression tests
- requesting new tests
- categorizing test cases (**Reusable** and **Obsolete** test cases)
- prioritizing test cases

One-time regression testing

Simple, one-time based ([Full](#) or [Custom](#)) regression testing service.

Time-based regression testing

Time-based regression testing is a cyclical service performed by the validation team. Depending on the needs, regression tests can be run each week, month or custom period of time specified by the client. This kind of testing approach has the highest detection of product malfunctions, relying on re-testing whole regression test suite.



Specification

QA technologies

Technologies that are supported by Regression Testing Solutions can be part of the validation package requested by the client:

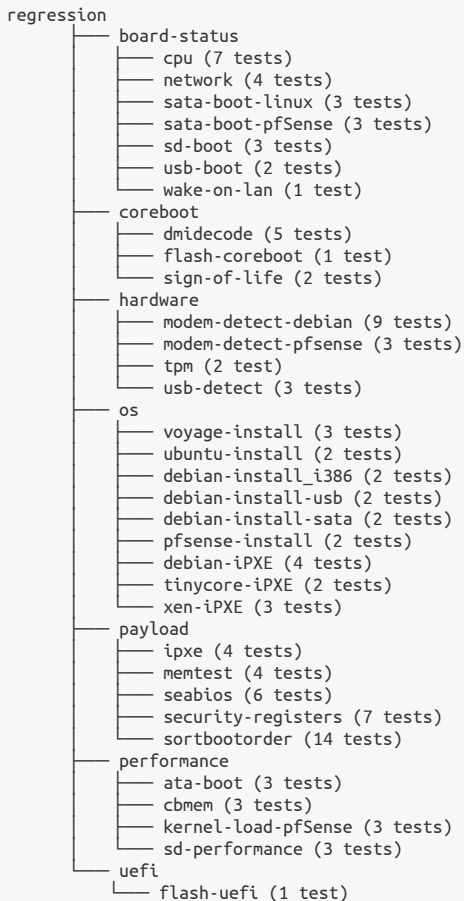
- Installation validation
 - Linux based systems (Debian, Ubuntu, Voyage)
 - BSD based systems (pfSense, OPNsense)
- OS performance validation
- Embedded Firmware validation (coreboot, U-Boot, Zephyr, mbedOS, bare metal)
- Hypervisors validation (Bareflank, Xen, Intel ACRN)
- Preboot Execution Environment (PXE) over network validation
- Hardware compatibility testing (storages, modems, security modules)



3mdeb validation rack



Regression test suite tree





Spreadsheet report

Each service option includes thorough validation report document and regression tests results in the spreadsheet form. The report consists of tests details with percentage coverage, graphical and text summary and issues description between the current and previous releases.

Test ID	Description	v4.8.0.7	v4.9.0.1	v4.9.0.2	v4.9.0.3	v4.9.0.4
SEC1.0	Pressing Z in sortbootorder MUST cause entrance to secret menu	PASS	PASS	PASS	PASS	PASS
SEC1.1	Typing w <serial> in secret menu MUST write serial	PASS	PASS	PASS	PASS	PASS
SEC1.2	Pressing r in secret menu MUST read serial	PASS	PASS	PASS	PASS	PASS
SEC1.3	Pressing e in secret menu MUST erase serial	PASS	PASS	PASS	PASS	PASS
SEC1.4	Pressing s in secret menu MUST show lock status	PASS	PASS	PASS	PASS	PASS
SEC1.5	Pressing q in secret menu MUST leave to main menu	PASS	PASS	PASS	PASS	PASS
SEC1.6	PXE booted Debian MUST see serial numbers written in security registers	PASS	PASS	PASS	PASS	PASS
USB1.1	50x coldboots to check if SeaBIOS detects USB sticks properly	PASS	PASS	PASS	PASS	PASS
USB1.2	50x warmboots to check if SeaBIOS detects USB sticks properly	PASS	PASS	PASS	PASS	PASS
USB1.3	50x resets to check if SeaBIOS detects USB sticks properly	PASS	PASS	PASS	PASS	PASS
VOY1.1	Boot into Voyage installer	PASS	PASS	PASS	PASS	PASS
VOY1.2	Login into Voyage installer	PASS	PASS	PASS	PASS	NOT TESTED
VOY1.3	Perform installation	PASS	PASS	PASS	PASS	NOT TESTED
UBU1.1	Install Ubuntu 16.04 LTS	PASS	PASS	PASS	PASS	PASS
UBU1.2	Boot into installed Ubuntu 16.04 LTS	PASS	PASS	PASS	PASS	PASS
DNI1.1	Install Debian i386	PASS	PASS	PASS	PASS	PASS
DNI1.2	Boot into installed Debian i386	PASS	PASS	PASS	PASS	PASS
DNI1.3	Install Debian amd64	PASS	PASS	PASS	PASS	PASS
DNI1.4	Boot into installed Debian amd64	PASS	PASS	PASS	PASS	PASS
DNI1.5	Install Debian amd64 on SATA storage	NOT TESTED	NOT TESTED	NOT TESTED	PASS	PASS
DNI1.6	Boot into installed Debian amd64 on SATA storage	NOT TESTED	NOT TESTED	NOT TESTED	PASS	PASS
DEB1.1	Debian from IPXE 4.14.y	PASS	PASS	PASS	PASS	PASS
DEB1.2	Debian from IPXE 4.15.y	PASS	PASS	PASS	PASS	PASS
DEB1.3	Debian from IPXE 4.16.y	PASS	PASS	PASS	PASS	PASS
DEB1.4	50x reboots to Debian 4.14.y booted over IPXE	PASS	PASS	PASS	PASS	PASS
PFS1.1	pfsense 2.4.2 installation on (m)SATA disk	NOT SUPPORTED	NOT SUPPORTED	NOT SUPPORTED	NOT SUPPORTED	NOT SUPPORTED
PFS1.2	Booting into installed pfsense	NOT SUPPORTED	NOT SUPPORTED	NOT SUPPORTED	NOT SUPPORTED	NOT SUPPORTED
TCL1.1	Boot to Core 6.4 booted over IPXE	PASS	PASS	PASS	PASS	PASS
TCL1.2	50x reboots to Core 6.4 booted over IPXE	FAIL	PASS	PASS	PASS	FAIL
XEN1.1	Boot Xen from IPXE	PASS	PASS	PASS	PASS	PASS
XEN1.2	Verify if IOMMU is enabled	PASS	PASS	FAIL	FAIL	PASS
XEN1.3	100x reboots to Xen 4.8	FAIL	FAIL	FAIL	FAIL	FAIL
SD1.1	Reading speed from SD card should exceed 20MB/s	PASS	PASS	PASS	PASS	PASS
MDM1.1	HUAWEI, WLE200NX, TL-WN722N modem detection pfsense coldboot	NOT SUPPORTED	NOT SUPPORTED	NOT SUPPORTED	NOT SUPPORTED	NOT SUPPORTED
MDM1.2	HUAWEI, WLE200NX, TL-WN722N modem detection pfsense cvramboot	NOT SUPPORTED	NOT SUPPORTED	NOT SUPPORTED	NOT SUPPORTED	NOT SUPPORTED
MDM1.3	HUAWEI, WLE200NX, TL-WN722N modem detection pfsense reboot	NOT SUPPORTED	NOT SUPPORTED	NOT SUPPORTED	NOT SUPPORTED	NOT SUPPORTED



HTML log files

RobotFramework infrastructure generates comprehensive log files from each test cases run by the regression suite. Due to the HTML format, after opening files in the web browser, logs become fully interactive. The client can check whether validation procedure was executed according to the requirements by expanding selected test-cases or read in detail error information about malfunction of the tested feature.

Test Execution Log

<div><div></div><div>SUITE</div></div> Performance	00:16:23.428
Full Name: Performance	
Documentation: performance suite	
Source: /home/artur-rs/Desktop/osfv/open-firmware-rte/performance	
Start / End / Elapsed: 20190206 00:57:38.214 / 20190206 01:14:01.642 / 00:16:23.428	
Status: 10 critical test, 3 passed, 7 skipped, +0 failed 10 test total, 3 passed, 7 skipped, 0 failed	
<div><div></div><div>SETUP</div></div> Prepare platform \${fw_file}	00:01:38.918
<div><div></div><div>SUITE</div></div> Ata-Boot	00:00:01.039
<div><div></div><div>SUITE</div></div> Cbmem	00:14:40.910
Full Name: Performance.Cbmem	
Source: /home/artur-rs/Desktop/osfv/open-firmware-rte/performance/cbmem.robot	
Start / End / Elapsed: 20190206 00:59:18.796 / 20190206 01:13:59.706 / 00:14:40.910	
Status: 3 critical test, 3 passed, 0 failed 3 test total, 3 passed, 0 failed	
<div><div></div><div>SETUP</div></div> BuiltIn.Run Keywords Prepare Test Suite, Check If IPXE Is Enabled	00:00:17.649
<div><div></div><div>TEARDOWN</div></div> keywords.Log Out And Close Connection	00:00:00.033
<div><div></div><div>TEST</div></div> CBMEM1.0 Coreboot booting time after coldboot	00:04:44.042
Full Name: Performance.Cbmem.CBMEM1.0 Coreboot booting time after coldboot	
Start / End / Elapsed: 20190206 00:59:36.478 / 20190206 01:04:20.520 / 00:04:44.042	
Status: PASS (critical)	
<div><div></div><div>KEYWORD</div></div> BuiltIn.Set Suite Variable \${iterations}, 5	00:00:00.002
<div><div></div><div>KEYWORD</div></div> \${average} = BuiltIn.Set Variable 0	00:00:00.001
<div><div></div><div>KEYWORD</div></div> BuiltIn.Log To Console \n	00:00:00.000
<div><div></div><div>FOR</div></div> \${INDEX} IN RANGE [0 \${iterations}]	00:04:44.037
<div><div></div><div>KEYWORD</div></div> \${average} = BuiltIn.Evaluate \${average}/\${iterations}	00:00:00.001
Documentation: Evaluates the given expression in Python and returns the results.	
Start / End / Elapsed: 20190206 01:04:20.518 / 20190206 01:04:20.519 / 00:00:00.001	
01:04:20.518 TRACE Arguments: ['12.843094/5']	
01:04:20.519 TRACE Return: 2.5686008	
01:04:20.519 INFO \${average} = 2.5686008	
<div><div></div><div>KEYWORD</div></div> BuiltIn.Log To Console \nCoreboot average booting time: \${average} s\n	00:00:00.000
<div><div></div><div>TEARDOWN</div></div> BuiltIn.Run Keyword If Test Failed Check CPU temp	00:00:00.000
<div><div></div><div>TEST</div></div> CBMEM1.1 Coreboot booting time after warmboot	00:05:02.261
<div><div></div><div>TEST</div></div> CBMEM1.2 Coreboot booting time after system reboot	00:04:36.891
<div><div></div><div>SUITE</div></div> kernel-load-pfSense	00:00:01.125
<div><div></div><div>SUITE</div></div> Sd Performance	00:00:00.802



Bug management

All malfunctions and bugs are precisely described and linked in the service selected by the client. We are experienced in:

- GitLab
- GitHub
- Atlassian/JIRA Software

Although, any new tracking tool is welcome and will surely become familiar in no time.

NOTE: *If there will be a request for the help with bugs elimination (mainly in the embedded firmware area) we can also provide such support.*