

Unit Testing mit Vectorcast und fortlaufende Integration von Testergebnissen mit Jenkins bei der Ottobock Healthcare GmbH

Anwenderbericht

Torsten Wender, Embedded Testing | München, 22. Juni 2017



Gliederung

- Unit Testing bei Ottobock
 - * Top 5 Anforderungen an Unit Test Tool
 - * Lösung mit Vectorcast
- Testintegration bei Ottobock
 - * Top 5 Anforderungen an CI Tool
 - * Lösung mit Jenkins

Unit Testing bei Ottobock

Top 5 Anforderungen an Unit Test Tool

1. Tool soll effiziente C++-Testentwicklung ermöglichen

- Tabellarische Testdateneingabe, Stubs, Testdebugging, etc.
- Kontrollflussprüfung, Zusichern von Übergabeparametern
- Umgang mit Abstrakten Klassen/Templates

2. Tool soll Coverage-Messung ermöglichen

- Nachweis der durch Regulatoren geforderten Testabdeckung

Unit Testing bei Ottobock

Top 5 Anforderungen an Unit Test Tool

3. Tool soll mit verschiedenen Compilern funktionieren
 - Konfigurationsdatensatz-Verwaltung
 - Trennung von Konfigurationsdaten und Test-Input-Daten
4. Tool soll mit verschiedenen Targets funktionieren (Testausführung)
 - Windows-Simulator, Launchpad, Platinen aus Eigenfertigung
5. Tool soll Dokumentation in regulatorisch-relevantem Format erzeugen

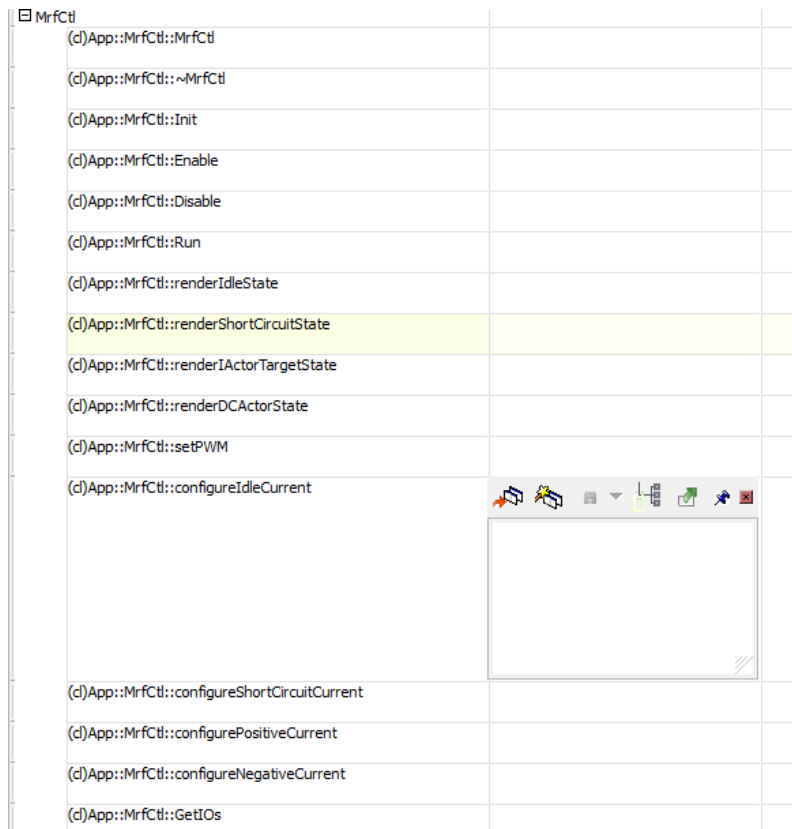
Umsetzung bei Ottobock mit Vectorcast

1. Effiziente C++ Testentwicklung (Tabellarische Testdateneingabe)

USER_GLOBALS_VCAST				
Timestamp				
<<SBF>>				
<<GLOBAL>>				
OBOS::Kernel::HeapTracker::m_heapLocked	bool			
OBOS::Time::TimeStamp Instance	ClassPtr	OBOS::Time::TimeStamp	<<choose a subclass>>	
OBOS::Time::TimeStamp	SubClass	OBOS::Time::TimeStamp::TimeStamp()		
OBOS::Time::TimeStamp::TimeStamp				
m_year	unsigned int	2017		
m_month	unsigned char	05		
m_day	unsigned char	24		
m_dayOfWeek	unsigned char	3		
m_hour	unsigned char	11		
m_minute	unsigned char	32		
m_second	unsigned char	5		
OBOS::Time::TimeStamp::m_daysMonth	user			
(d)OBOS::Time::TimeStamp::GetDifferenceInMinutes				
tod2	ClassPtr	OBOS::Time::TimeStamp	<<choose a subclass>>	
OBOS::Time::TimeStamp	SubClass	OBOS::Time::TimeStamp::TimeStamp()		
OBOS::Time::TimeStamp::TimeStamp				
m_year	unsigned int	2017		
m_month	unsigned char	05		
m_day	unsigned char	24		
m_dayOfWeek	unsigned char	3		
m_hour	unsigned char	11		
m_minute	unsigned char	32		
m_second	unsigned char	5		
OBOS::Time::TimeStamp::m_daysMonth	user			
return	unsigned long			
Stubbed Subprograms				

Umsetzung bei Ottobock mit Vectorcast

1. Effiziente C++ Testentwicklung (Stubbing)



Umsetzung bei Ottobock mit Vectorcast

1. Effiziente C++ Testentwicklung (Kontrollflussprüfung)

The screenshot displays the VectorCAST IDE interface. On the left, the 'Subprograms' pane shows a list of functions under the 'UUT' (Unit Under Test) category for 'Beeper.cpp'. The functions include various methods of the 'Beeper' class, such as 'SetFeedbackConfig', 'GetFee...', 'Init', 'IsActive', 'QueryI...', 'SetFee...', 'SetMute', 'SetPow...', 'setupF...', 'start', 'stop', and '~Beeper'. Below these, under the 'Stubbed Subprograms' category, are 'AbstractBaseDevice.h', 'AbstractFeedback.h', and 'AbstractHilDevice.h'. On the right, the 'Control Flow' pane shows a list of three entries, all pointing to 'Beeper.cpp.(c)OBOS::Feedback::Beeper::SetFeedbackConfig'.

Control Flow	
1	Beeper.cpp.(c)OBOS::Feedback::Beeper::SetFeedbackConfig
2	uut_prototype_stubs.OBOS::Feedback::AbstractFeedback::S
3	Beeper.cpp.(c)OBOS::Feedback::Beeper::SetFeedbackConfig

Umsetzung bei Ottobock mit Vectorcast

1. Effiziente C++-Testentwicklung (Abstrakte Klassen, Templates)

- Instanzierbare Klassen werden automatisch generiert, kein UserCode notwendig

SoundPin	ClassPtr	OBOS::IO::vcast_concrete_IGPIO
OBOS::IO::vcast_concrete_IGPIO	SubClass	...e_IGPIO::vcast_concrete_IGPIO()
OBOS::IO::vcast_concrete_IGPIO::vcast_c...		

- Spezialisierungen können „gewählt“ werden, wenig User Code notwendig

VcNotificationManager		
(c)OBOS::Notification::NotificationManager<OBOS::Notification::NotificationState_t, 8U>::AddNotification		
(c)OBOS::Notification::NotificationManager<OBOS::Notification::NotificationState_t, 8U>::RemoveNotification	✓	
(c)OBOS::Notification::NotificationManager<OBOS::Notification::NotificationState_t, 8U>::GetStateOfNotification	✓	
(c)OBOS::Notification::NotificationManager<OBOS::Notification::NotificationState_t, 8U>::SetStateOfNotification		
(c)OBOS::Notification::NotificationManager<OBOS::Notification::NotificationState_t, 8U>::SearchNotificationToPlay	✓	
(c)OBOS::Notification::NotificationManager<OBOS::Notification::NotificationState_t, 8U>::SearchNotificationsWithSeverity	✓	
(c)OBOS::Notification::NotificationManager<OBOS::Notification::NotificationState_t, 8U>::HasNotificationsWithSeverity	✓	
(c)OBOS::Notification::NotificationManager<OBOS::Notification::NotificationState_t, 8U>::getIndexOf	✓	
(c)OBOS::Notification::NotificationManager<OBOS::Notification::NotificationState_t, 8U>::removeNotificationAt	✓	
(c)OBOS::Notification::NotificationManager<OBOS::Notification::NotificationState_t, 8U>::NotificationManager	✓	

Umsetzung bei Ottobock mit Vectorcast

2. Messung der Testabdeckung

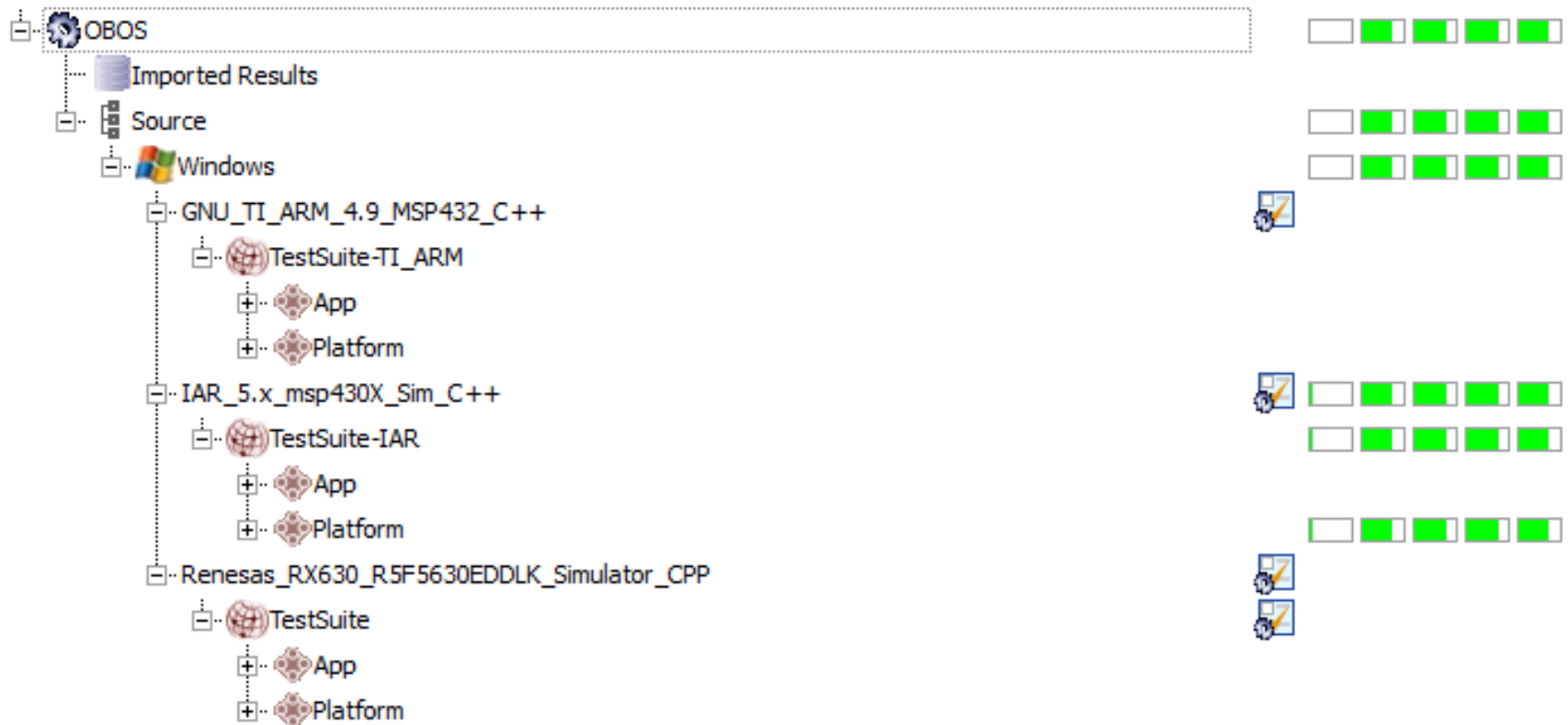
```

10  0      (T)      OBOS::Time::TimeStamp::GetDifferenceInMinutes
10  1      *        uint16_t minYear;
10  2      *        uint32_t thisMinutes;
10  3      *        uint32_t thatMinutes;
10  4      *        (static_cast<void>(0U));
10  5      ( ) (F)    if (
10  5.1    ( ) (F)    this->m_year < tod2->m_year)
                        {
10  6                        minYear = this->m_year;
                        }
                        else
                        {
10  7                        *        minYear = tod2->m_year;
                        }
                        // convert time of day in minutes
10  8      *        thisMinutes = this->GetMinutesSinceYear(minYear);
10  9      *        thatMinutes = tod2->GetMinutesSinceYear(minYear);
                        // The greater value is the minuend, the smaller value is the subtrahend
10  10     ( ) (F)    if (
10  10.1   ( ) (F)    thisMinutes > thatMinutes)
                        {
10  11                        return thisMinutes - thatMinutes;
                        }

```

Umsetzung bei Ottobock mit Vectorcast

3. Unterstützung verschiedener Compiler (Konfigurationsdatensatzverwaltung)



Umsetzung bei Ottobock mit Vectorcast

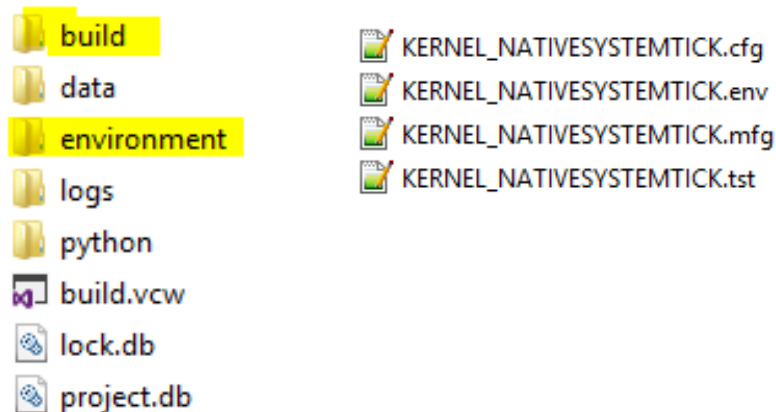
3. Unterstützung verschiedener Compiler (Konfigurationsdatensatzverwaltung)

Name	Compiler	
Environment Variable:		
Whitebox	YES	
Coverage type:	STATEMENT+MDC	
Parent Lib:		
ADA		
C/C++		
Compiler hierarchy string:	IAR_5.x_msp430X Sim_C++	
Preprocessor/Compiler		
Preprocessor file:	?.i	
Defined variables:	_MSP430F5438A __data20= NDEBUG	
Define flag:	-D	
Include flag:	-I	
Compile command:	icc430 -debug -e -eccc++ -core=430X --double=32 --data_model...	
Preprocessor command:	icc430 -e -eccc++ -preprocess=nlc . -preinclude \$(VCAST_SOURC...	
Linker		
Output file flag:	-o ^	
Object file extension:	.r43	
Executable file extension:	.d43	
Linker command:	xlink	
Linker options:	"\$(VCAST_SOURCE_ROOT)OBOS\\$(VCAST_BRANCH_NAME)\platfor...	
Green Hills intex utility command:		
Green Hills integrate file:		
Debug		
Debugger command:	execute_debug.bat	
Command line debugger	<input type="checkbox"/>	
Misc		
Preprocess options		
Header extensions		
C extensions		
C++ extensions		
Language mode	.cpp	
Parser flags:	-w --c++ -iar -iar_version=msp430 --type_info_in_std	
Maintain relative include directories		
Collapse expanded header files	COLLAPSE_SYSTEM_HEADERS	
Mixed C/C++		
C preprocessor command:	icc430 -e -preprocess=nlc .	
C compile command:	icc430 -debug -e -core=430X --double=32 --data_model=medium	
C parser flags:	-w -c -iar -iar_version=msp430	

Umsetzung bei Ottobock mit Vectorcast

3. und 4. Unterstützung verschiedener Compiler und Targets

- Trennung von Konfigurationsdaten und Test-Input-Daten
- Trennung von versionsüberwachten Inhalten und Compilaten
- Speicherung als editierbare Textdateien



Umsetzung bei Ottobock mit Vectorcast

5. Dokumentation in regulatorisch-relevantem Format (Requirements Gateway)

- [-] [Polarion] GK2
 - ...GK2-100 Korrosion - Funktionsfähigkeit
 - ...GK2-103 CleaningReinigung
 - ...GK2-105 Stoß und Schlagfestigkeit
 - ...GK2-107 Geräuschpegel
 - ...GK2-108 Einsatzbedingungen
 - ...GK2-111 Funktionelle Haltbarkeit
 - ...GK2-117 Lebensdauer
 - ...GK2-119 Robustheit beim Transport
 - ...GK2-122 Bedienbarkeit
 - ...GK2-124 Produktsegment
 - ...GK2-125 Kosmetik
 - ...GK2-131 Modumschaltung
 - ...GK2-132 einfache Aktivierung
 - ...GK2-133 Absichtserkennung durch Bewegungsmuster
 - ...GK2-134 Abschalten der Absichtserkennung
 - ...GK2-139 Sensoren
 - ...GK2-141 Gehen - Standphasensicherung
 - ...GK2-142 Extension
 - ...GK2-145 Winkelbereich
 - ...GK2-148 Akku-Ladeintervall
 - ...GK2-149 Fail Safe
 - ...GK2-152 Akku
 - ...GK2-154 Feedback
 - ...GK2-158 **GK2-154 Feedback**
 - ...GK2-160 The knee joint must provide feedback to the end user.
 - ...GK2-162
 - ...GK2-164
 - ...GK2-168 Datenlogger/Schrittzähler
 - ...GK2-170 Strukturfestigkeit
 - ...GK2-172 ME-Geräte Norm 60601 bestehen
 - ...GK2-182 Überlastschutz
 - ...GK2-184 Austauschbare Komponenten
 - ...GK2-186 Diagnosemöglichkeit
 - ...GK2-188 Vermeiden plötzlicher Eigenschaftsänderungen
 - ...GK2-190 Akku-Lebensdauer

Specify options for interfacing with the requirements subsystem.

Subsystem profile: Polarion

Python script name: polarion_gateway

Polarion server: http://dedusrv210:8080/polarion/ws/services/

Polarion user: testmanager

Polarion password: ☒ Save password (in cleartext)

Project id: GK2

Get Fields

Testintegration bei Ottobock

Top 5 Anforderungen an CI Tool

1. Tool soll konfigurierbare Testautomatisierung leisten
 - parametrisierte Testausführung auf virtuellen Maschinen
2. Tool soll auf Commit-Events im Produktionscode oder Testcode reagieren
 - Unterstützung mehrerer SCMs (TFS, SVN, Git)
3. Tool soll Testergebnisse aufzeichnen
 - Grafische Darstellung der Ergebnisse
 - Grafische Darstellung des Ergebnis-Trends
 - Berechnung der Ergebnis-Differenz zu vorherigen Run

Testintegration bei Ottobock

Top 5 Anforderungen an CI Tool

4. Tool soll Coverage-Ergebnisse aufzeichnen

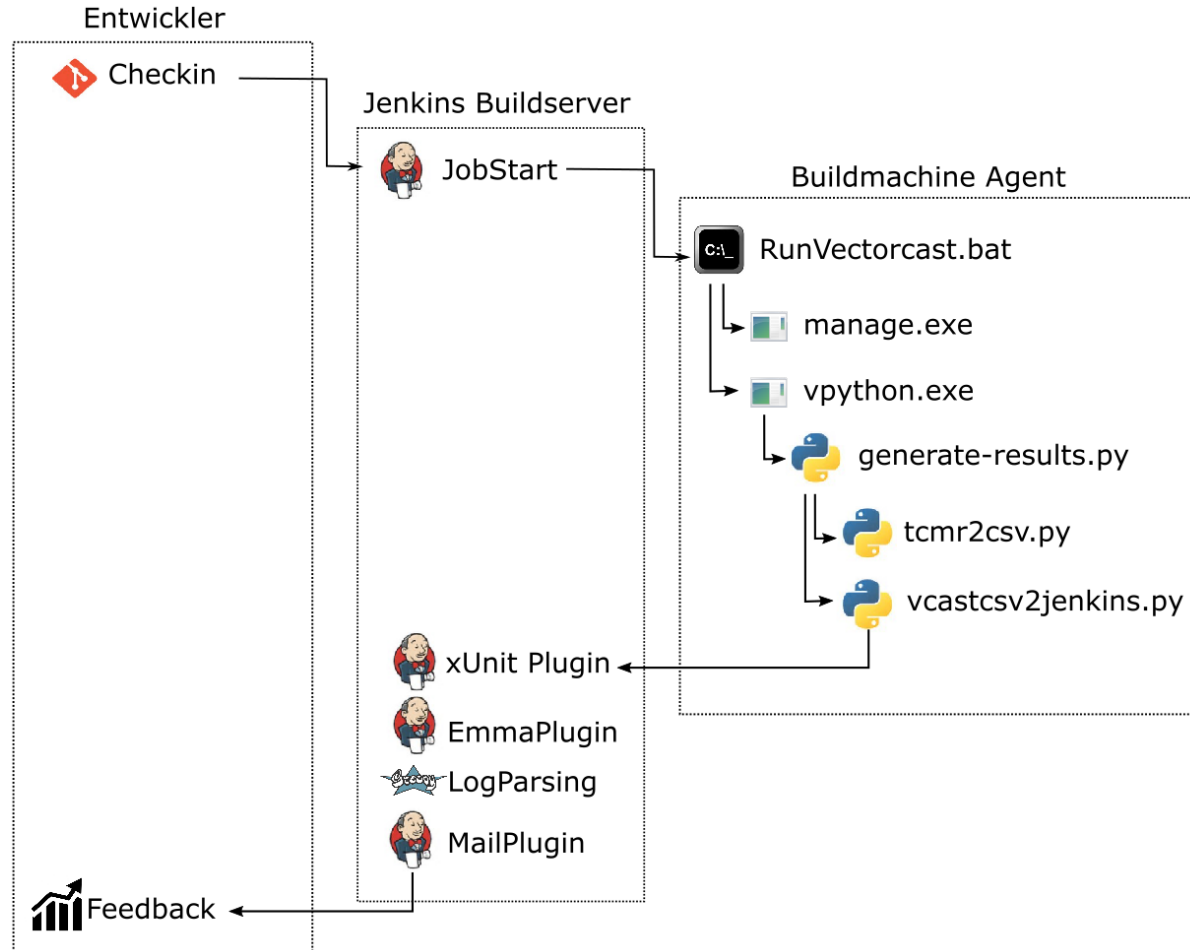
- Grafische Darstellung der Ergebnisse
- Grafische Darstellung des Ergebnis-Trends

5. Tool soll Ergebnisse unmittelbar an Entwickler zurückmelden

- frühe Fehlererkennung
- testgetriebene Entwicklung

Testintegration bei Ottobock

1. Konfigurierbare Testautomatisierung



Testintegration bei Ottobock

1. Konfigurierbare Testautomatisierung - Hauptskript

```
echo.  
echo #####  
echo #####      Building Environments      #####  
echo #####  
echo.  
%VECTORCAST_DIR%\manage --project %homedir%%projname%.vcm --level %level% --status  
%VECTORCAST_DIR%\manage --project %homedir%%projname%.vcm --level %level%  
--release-locks  
%VECTORCAST_DIR%\manage --project %homedir%%projname%.vcm --level %level%  
--build-execute --incremental --output %projname%.vcm_manage_report.html  
echo.  
echo #####  
echo #####      Creating Reports      #####  
echo #####  
echo.  
%VECTORCAST_DIR%\vpython %VECTORCAST_DIR%  
\python\vector\apps\Jenkins\generate-results.py %homedir%%projname%.vcm --level  
%level% --jenkins  
%VECTORCAST_DIR%\manage --project %homedir%%projname%.vcm --level %level%  
--full-status=%projname%.vcm_full_report.html  
%VECTORCAST_DIR%\manage --project %homedir%%projname%.vcm --level %level%  
--full-status > %projname%.vcm_full_report.txt  
%VECTORCAST_DIR%\vpython %VECTORCAST_DIR%\python\vector\apps\Jenkins\getTotals.py  
%projname%.vcm_full_report.txt
```

Umsetzung bei Ottobock mit Jenkins

1. Konfigurierbare Testautomatisierung - Eingabe

Publish xUnit test result report

Check-Version N/A (default)

Check-Version N/A (default) Pattern

Skip if there are no test files ☒

Fail the build if test results were not updated this run ☒

Delete temporary JUnit files ☒

Stop and set the build to 'failed' status if there are errors when processing a result file ☒

Record VectorCAST Coverage Data (Legacy Format)

Folders or files containing VectorCAST XML reports



Specify the path to the VectorCAST XML report files, relative to [the workspace root](#)

- If you left this field blank the plugin will look for files matching the pattern: `**/coverage*.xml` in the workspace.
- Or you can enclose the search specifying a list of files and folders separated by semicolon.
- Or use an Ant Fileset pattern.


Control build stability using threshold ☐


Specify whether build should fail if actual coverage is less than defined by threshold

Health reporting

	Statement	Branch	Basis Path - this one	MC/DC	MC/DC Pair
	<input type="text" value="90"/>	<input type="text" value="90"/>	<input type="text" value="90"/>	<input type="text" value="90"/>	<input type="text" value="90"/>
	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>

Configure health reporting thresholds.

For the  row, leave blank to use the default values (i.e. 100, 70, 80, and 80 for class, method, block and line respectively).

For the  row, leave blank to use the default values (i.e. 0, 0, 0, 0).

Umsetzung bei Ottobock mit Jenkins

1. Konfigurierbare Testautomatisierung - Ausgabe

```
03:02:10 [OB Shell - Vectorcast] [STDOUT] Running: (cl)OBOS::Time::TimeStamp::GetMinutesSinceYear.001
03:02:10 [OB Shell - Vectorcast] [STDOUT] Preparing Test Data
03:02:10 [OB Shell - Vectorcast] [STDOUT] Running Test Case
03:02:10 [OB Shell - Vectorcast] [STDOUT] Retrieved RSP License: "C_TARGET_IAR"
03:02:10 [OB Shell - Vectorcast] [STDOUT] Pre-Run setup for stdout execution
03:02:10 [OB Shell - Vectorcast] [STDOUT] Compiling file Test Case Data
03:02:10 [OB Shell - Vectorcast] [STDOUT] Linking Instrumented Harness
03:02:10 [OB Shell - Vectorcast] [STDOUT] Running Test with command: C:\jenkins\workspace\OBOS\MicroC\Platform\Tools\VC\SupportFiles-IAR\vcast_debug\settings\vcast_debug.Debug.cspy.bat
C:\jenkins\workspace\OBOS\MicroC\Platform\Tools\VC\OBOS\build\16700911\SYSTEM_TIMER_TIMESTAMP\UUT_INST.d43
03:02:12 [OB Shell - Vectorcast] [STDOUT] Building Execution History Data for Event: 1 of 2
03:02:12 [OB Shell - Vectorcast] [STDOUT] Building Execution History Data for Event: 2 of 2
03:02:12 [OB Shell - Vectorcast] [STDOUT] Building Execution Report
03:02:12 [OB Shell - Vectorcast] [STDOUT] Building Execution Report for Event: 1 of 2
03:02:12 [OB Shell - Vectorcast] [STDOUT] Building Execution Report for Event: 2 of 2
03:02:12 [OB Shell - Vectorcast] [STDOUT] Created Execution Report
03:02:12 [OB Shell - Vectorcast] [STDOUT] Updating Coverage Data
03:02:12 [OB Shell - Vectorcast] [STDOUT] Test Execution Complete

03:04:14 Archiving artifacts
03:07:55 Recording fingerprints
03:07:55 [xUnit] [INFO] - Starting to record.
03:07:55 [xUnit] [INFO] - Processing Check-Version N/A (default)
03:07:57 [xUnit] [INFO] - [Check-Version N/A (default)] - 115 test report file(s) were found with the pattern '**/test_results_*.xml'
testing framework 'Check-Version N/A (default)'.
03:08:04 [xUnit] [INFO] - Check 'Failed Tests' threshold.
03:08:04 [xUnit] [INFO] - Check 'Skipped Tests' threshold.
03:08:04 [xUnit] [INFO] - Setting the build status to SUCCESS
03:08:04 [xUnit] [INFO] - Stopping recording.
03:08:04 Emma: looking for coverage reports in the provided path: **/coverage_results_*.xml
03:08:06 Emma: found 112 report files:
03:08:06 C:\jenkins\workspace\OBOS\MicroC\Platform\Tools\VC\SupportFiles-IAR\Reports\xml_data\coverage_results_HAL_ABSTRACT
03:08:06 C:\jenkins\workspace\OBOS\MicroC\Platform\Tools\VC\SupportFiles-IAR\Reports\xml_data\coverage_results_HAL_ABSTRACT
03:08:06 C:\jenkins\workspace\OBOS\MicroC\Platform\Tools\VC\SupportFiles-IAR\Reports\xml_data\coverage_results_HAL_BSLSUPPO
```

Umsetzung bei Ottobock mit Jenkins

2. Commit Trigger für Testausführung

Änderungen

OBOS - Unit Tests v0.0.91.29780 (15.06.2017 13:45:39)

1. fixed /*vcast_dont_instrument_end*/ in bytestreamdistributor.h — [Wender, Torsten](#) / [Microsoft Team Foundation Server/Visual Studio Team Services](#)
2. Added more VectorCast Tests — [Bartels, Christopher](#) / [Microsoft Team Foundation Server/Visual Studio Team Services](#)
3. Fixed if-check in AS5055 Init; Implemented Unit-Tests for Constructor and Init method on AS5055 class in VCast — [Bartels, Christopher](#) / [Microsoft T](#)

OBOS - Unit Tests v0.0.90.29774 (15.06.2017 01:00:01)

1. Finished Unit-Test implementation for Watchdog430 — [Bartels, Christopher](#) / [Microsoft Team Foundation Server/Visual Studio Team Services](#)
2. Mini-Tutorial VCast; edit Watchdog Unitcases (incomplete coverage) — [Bartels, Christopher](#) / [Microsoft Team Foundation Server/Visual Studio Team](#)

OBOS - Unit Tests v0.0.89.29755 (13.06.2017 01:00:03)

Umsetzung bei Ottobock mit Jenkins

3. Testergebnis und Reportarchivierung



Testergebnis (106 fehlgeschlagene Tests / +2)

IAR_5_x_msp430X_Sim_C++.TestSuite-IAR.HEADER_OF_SYSTEM_MANAGER_NOTIFICATIONMANAGER.
(cl)OBOS::Notification::NotificationManager<OBOS::Notification::Notification.002

IAR_5_x_msp430X_Sim_C++.TestSuite-IAR.HEADER_OF_SYSTEM_MANAGER_NOTIFICATIONMANAGER.
8U>::SetStateOfNotification.(cl)OBOS::Notification::NotificationManager<OBOS::Notification::Notification.004

Zeige alle fehlgeschlagenen Tests >>>

Artefakte von „OBOS - Unit Tests OBOS - Unit Tests v0.0.88.29709“



OBOS / MicroC / Platform / Tools / VC / SupportFiles-IAR / Reports /



execution

management

xml_data

build.log

164,10 KB anzeigen

OBOS.vcm_full_report.html

36,72 KB anzeigen

OBOS.vcm_full_report.txt

24,15 KB anzeigen

OBOS.vcm_manage_incremental_rebuild_report.html

30,19 KB anzeigen

OBOS.vcm_overall_results.txt

140 B anzeigen

unit_test_fail_count.txt

3 B anzeigen

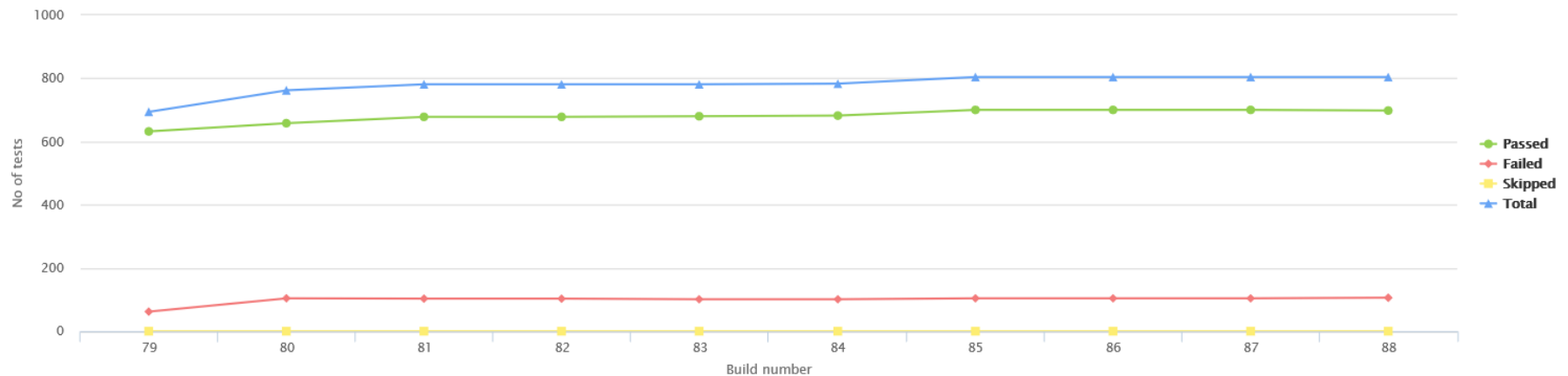
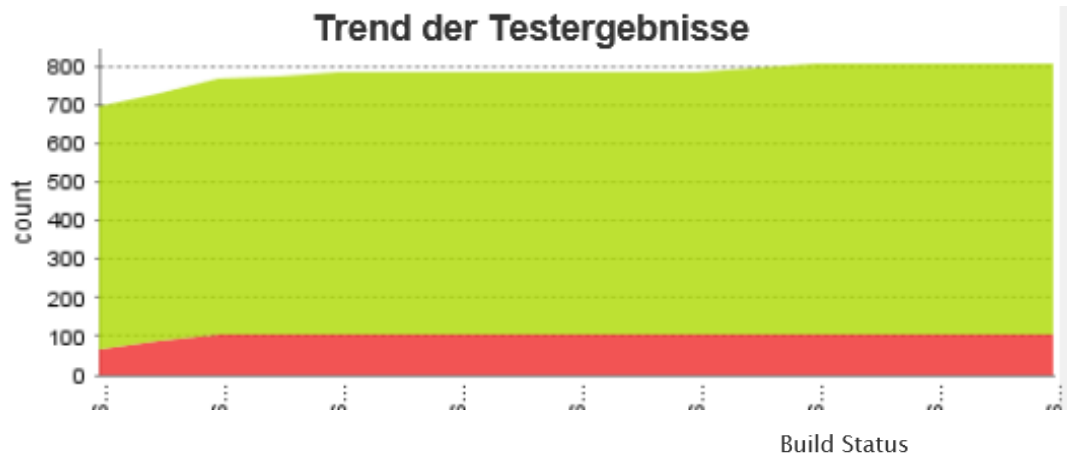


(Alle Dateien als ZIP-Archiv herunterladen)

Umsetzung bei Ottobock mit Jenkins

3. Test-Trends

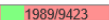
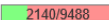
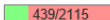
Trend der Testergebnisse



















































































Umsetzung bei Ottobock mit Jenkins

4. Coverage-Ergebnisse

Overall Coverage Summary

	Statement	Branch	MC/DC Pairs
All Units	21,1% 	22,6% 	20,8% 

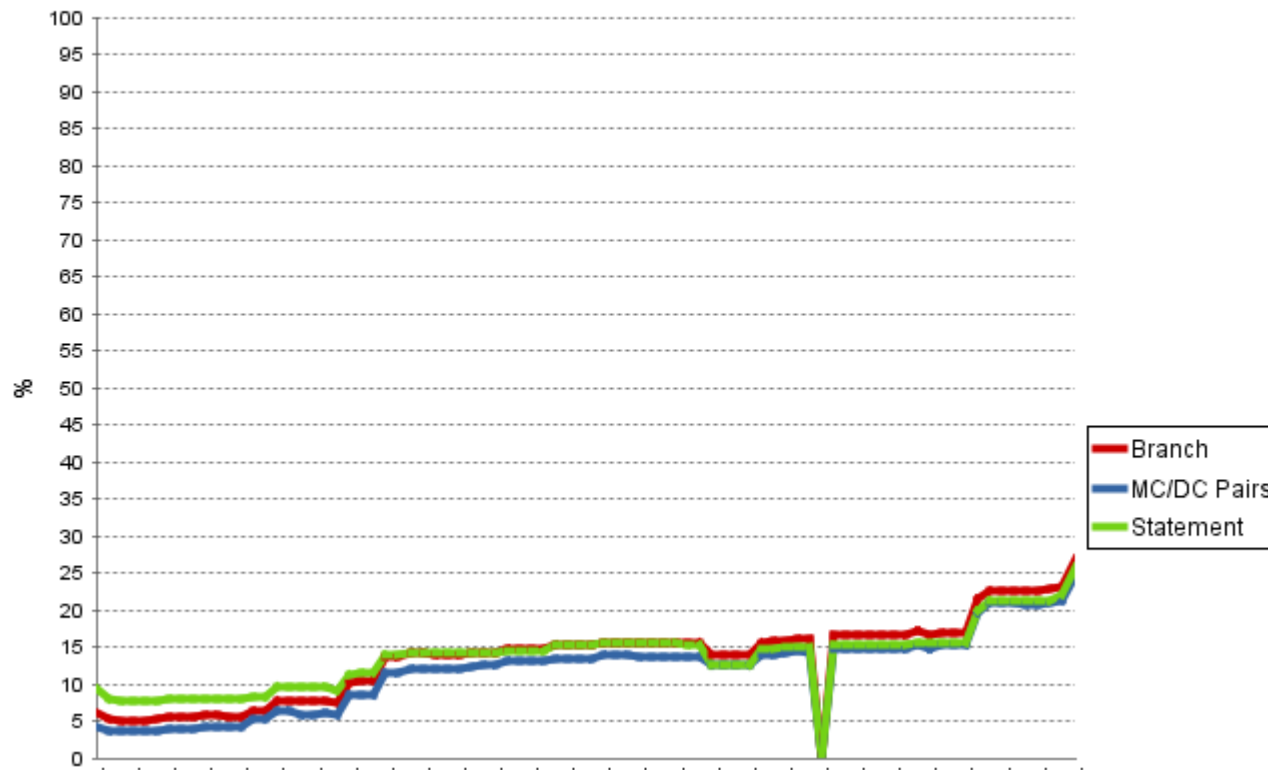
Coverage Breakdown by Environment

IAR_5x_msp430X_Sim_C++_TestSuite-IAR_HEADER_OF_HIL_ZACWIRE_ZACWIREINTERRUPT	0,0% 	0,0% 	0,0% 
IAR_5x_msp430X_Sim_C++_TestSuite-IAR_HEADER_OF_KERNEL_ABSTRACTAPPMODULE	100,0% 	100,0% 	100,0% 
IAR_5x_msp430X_Sim_C++_TestSuite-IAR_HEADER_OF_KERNEL_ABSTRACTNATIVEAPPMODULE	100,0% 	100,0% 	
IAR_5x_msp430X_Sim_C++_TestSuite-IAR_HEADER_OF_KERNEL_APPMODULEINPUT	100,0% 	100,0% 	
IAR_5x_msp430X_Sim_C++_TestSuite-IAR_HEADER_OF_KERNEL_APPMODULEOUTPUT	100,0% 	100,0% 	
IAR_5x_msp430X_Sim_C++_TestSuite-IAR_HEADER_OF_SYSTEM_COLLECTIONS_STACK	98,5% 	95,2% 	89,5% 
IAR_5x_msp430X_Sim_C++_TestSuite-IAR_HEADER_OF_SYSTEM_COLLECTIONS_STACKITEM	100,0% 	100,0% 	
IAR_5x_msp430X_Sim_C++_TestSuite-IAR_HEADER_OF_SYSTEM_CONFIG_CONFIGDATAHANDLER	98,6% 	81,4% 	54,2% 
IAR_5x_msp430X_Sim_C++_TestSuite-IAR_HEADER_OF_SYSTEM_CRC_GENERICCRC	92,9% 	87,2% 	82,4% 
IAR_5x_msp430X_Sim_C++_TestSuite-IAR_HEADER_OF_SYSTEM_INTERFACE_ABSTRACTCALIBRATE	0,0% 	0,0% 	0,0% 
IAR_5x_msp430X_Sim_C++_TestSuite-IAR_HEADER_OF_SYSTEM_MANAGER_NOTIFICATIONMANAGER	95,8% 	92,2% 	83,3% 
IAR_5x_msp430X_Sim_C++_TestSuite-IAR_HEADER_OF_SYSTEM_MATH_MATRIX_MATRIXX	100,0% 	100,0% 	100,0% 
IAR_5x_msp430X_Sim_C++_TestSuite-IAR_HIL_AS5055_AS5055	0,0% 	0,0% 	0,0% 
IAR_5x_msp430X_Sim_C++_TestSuite-IAR_HIL_BMP280_BMP280	0,0% 	0,0% 	0,0% 
IAR_5x_msp430X_Sim_C++_TestSuite-IAR_HIL_BNO055_BNO055	19,6% 	19,4% 	19,0% 
IAR_5x_msp430X_Sim_C++_TestSuite-IAR_HIL_BNO055_BNO055ACC	0,0% 	0,0% 	0,0% 
IAR_5x_msp430X_Sim_C++_TestSuite-IAR_HIL_BNO055_BNO055COMDISPATCHER	100,0% 	100,0% 	100,0% 
IAR_5x_msp430X_Sim_C++_TestSuite-IAR_HIL_BNO055_BNO055COMDISPATCHERI2C	100,0% 	100,0% 	100,0% 
IAR_5x_msp430X_Sim_C++_TestSuite-IAR_HIL_BNO055_BNO055COMDISPATCHERUART	0,0% 	0,0% 	0,0% 
IAR_5x_msp430X_Sim_C++_TestSuite-IAR_HIL_BNO055_BNO055GYRO	0,0% 	0,0% 	0,0% 
IAR_5x_msp430X_Sim_C++_TestSuite-IAR_HIL_BNO055_BNO055MAG	0,0% 	0,0% 	0,0% 
IAR_5x_msp430X_Sim_C++_TestSuite-IAR_HIL_BUTTON_BUTTON	100,0% 	100,0% 	100,0% 
IAR_5x_msp430X_Sim_C++_TestSuite-IAR_HIL_FEEDBACK_BEEPER	100,0% 	100,0% 	100,0% 
IAR_5x_msp430X_Sim_C++_TestSuite-IAR_HIL_FEEDBACK_EXTENDEDDEEPIER	83,8% 	88,5% 	72,2% 
IAR_5x_msp430X_Sim_C++_TestSuite-IAR_HIL_FEEDBACK_EXTENDEDVIBRATION	3,5% 	1,2% 	0,0% 
IAR_5x_msp430X_Sim_C++_TestSuite-IAR_HIL_FEEDBACK_VIBRATION	98,5% 	96,6% 	91,7% 
IAR_5x_msp430X_Sim_C++_TestSuite-IAR_HIL_FRAM_FRAMMB85RS2MT	2,3% 	0,4% 	0,0% 
IAR_5x_msp430X_Sim_C++_TestSuite-IAR_HIL_INTERFACE_ABSTRACTFEEDBACK	100,0% 	100,0% 	100,0% 

Umsetzung bei Ottobock mit Jenkins

4. Coverage-Trend

VectorCAST Coverage Trend






Umsetzung bei Ottobock mit Jenkins

5. Entwickler-Feedback


BUILD UNSTABLE

Build URL <http://dedufevs001:8080/job/OBOS%20-%20Unit%20Tests/88/>
 Project: OBOS - Unit Tests
 Date of build: Thu, 08 Jun 2017 01:00:01 +0200
 Build duration: 2 hr 1 min
 Build cause: Started by timer
 Built on: Build-VM-1-ES-Win7x86-DEDUPET029

Health Report

W	Description	Score
	Coverage: Statement 1989/9423 (21%). Branch 2140/9488 (23%). MC/DC Pairs 439/2115 (21%).	21
	Test Result: 106 tests failing out of a total of 804 tests.	86
	Build stability: No recent builds failed.	100

TFS Changes Report

 Revision **29701** by **Faulhaber, Sven**

Comment: Dummy byte at SPI is now configurable

Action	File
edit	\$/OBOS/MicroC/Platform/HAL/Implementation/MSP430/MSP430F54xx56xx/BusCommunication/SpiDrr
edit	\$/OBOS/MicroC/Platform/HAL/Implementation/MSP430/MSP430F54xx56xx/BusCommunication/SpiDrr
edit	\$/OBOS/MicroC/Platform/HAL/Implementation/MSP430/MSP430F54xx56xx/BusCommunication/SpiDrr

Fazit (1)

Positiv

- Die genannten Anforderungen können von der Werkzeugkette weitestgehend erfüllt werden
- Robustheit und Performanz des Prozesses sind zufriedenstellend
- Wichtige Arbeitsschritte sind quellcodezugänglich (z.B. Python Skripte, Jenkins-Plugins) und können somit adaptiert/optimiert werden
- Frühes Feedback und somit frühzeitiges Erkennen von Bugs konnte erreicht werden

Fazit (2)

Negativ

- Hohe Komplexität bezüglich Compiler-, Linker-, Debugger-Einstellungen
 - Support war notwendig
- Hoher (aber einmaliger) Konfigurationsaufwand des Testintegrations-Prozesses
 - Jenkins Job Erstellskript war vorhanden
 - Kleinere Anpassungen waren nötig

Vielen Dank für Ihre
Aufmerksamkeit!

www.ottobock.com



Quality for life