Verifysoft TECHNOLOGY

Testwell CTC++

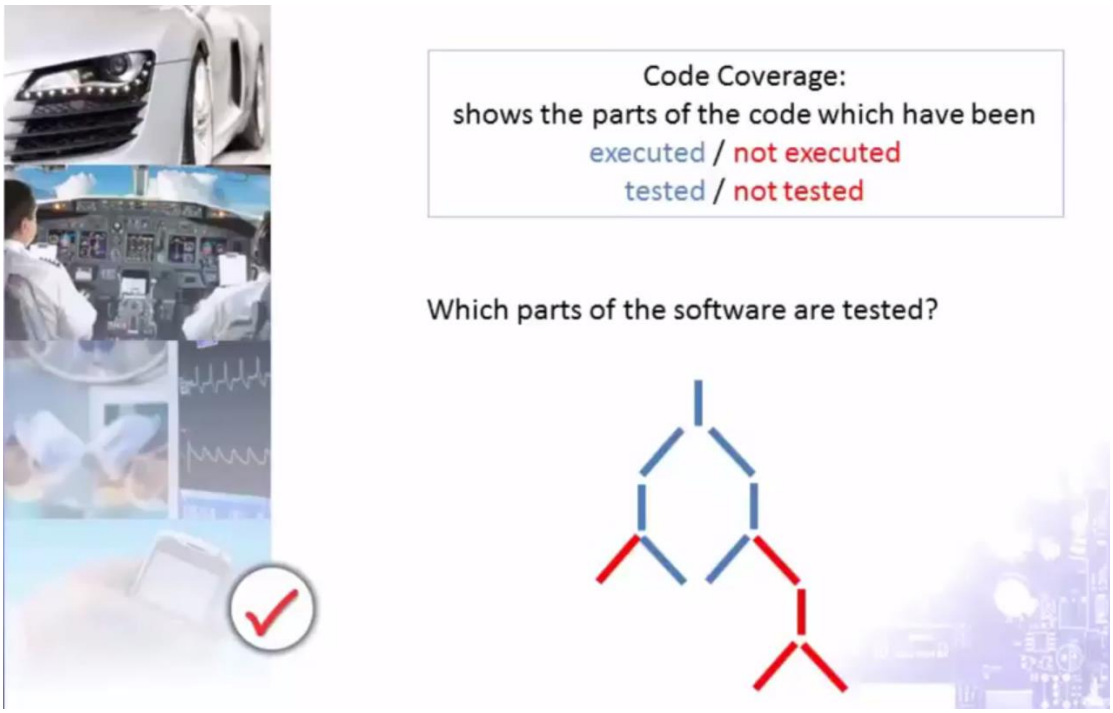
Code Coverage and ISO 26262

Hello and welcome to our presentation “How to fulfil the requirements of ISO 26262” regarding code coverage.

In addition to this standard of the automotive industry, we talk also about other standards in aerospace, healthcare, and other industries.

During this presentation we will show you what code coverage is and why it should be measured. We will explain different coverage levels and the requirements of the standard ISO 26262 and other standards. You will see how coverage is measured with Testwell CTC++, the leading test coverage analyser for embedded software developement.

Code coverage shows you for all parts of your code whether they have been tested or not.



In this picture the blue branches have already been tested, and the red ones not yet. To achieve 100% test coverage we need to test the “red” parts of the code as well.

Why measure code coverage?

With code coverage you can write more adapted test cases in order to test the parts of the code which are not yet covered.

Code coverage helps to avoid redundant test cases and save time and money.

You know when you can stop testing: the goal is achieved when the required test coverage has been reached.

Code coverage is also a proof of quality for your customers –

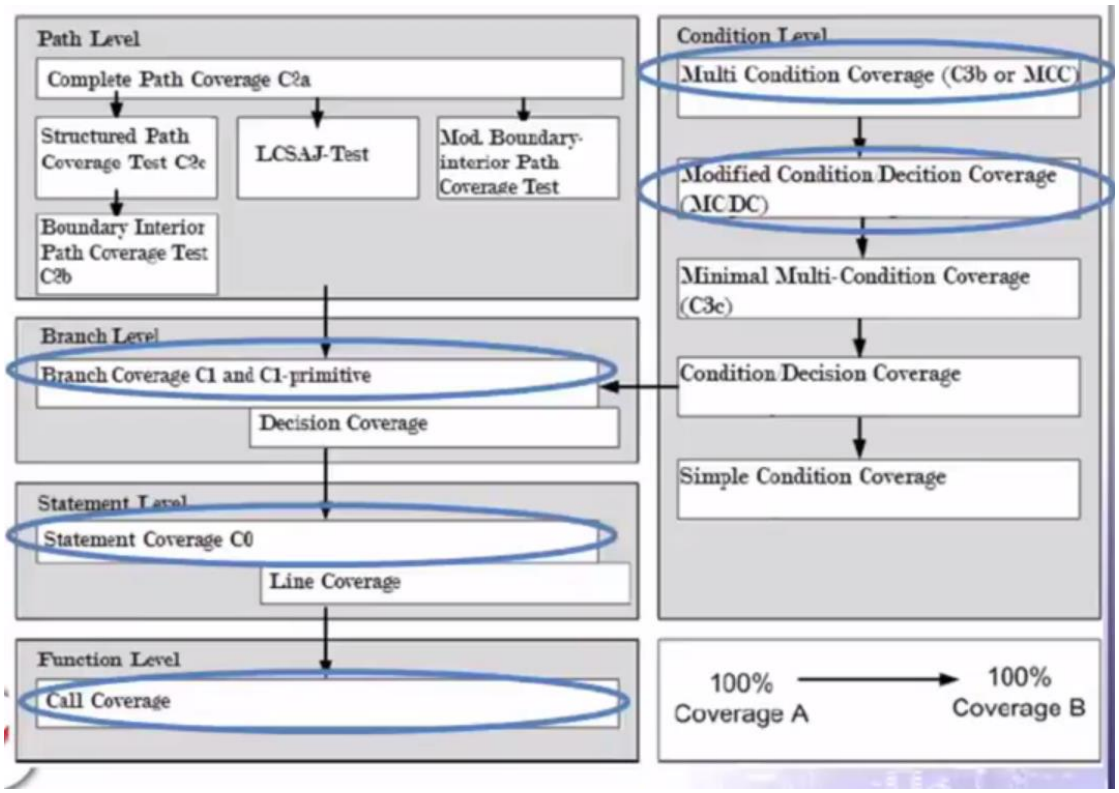
And if you subcontract development or testing, the coverage report shows you the amount of the tests done by your partner.

Code coverage tools help also to find Dead Code.

For all this reasons Code Coverage is required by safety standards.

Now, let's have a look to the different coverage levels.

Left hand on the bottom of our picture we have the function level. 100% Function or Call Coverage is achieved when each function has been executed at least one time. This is a very low coverage level.



On top of this we see the Statement Coverage. To achieve 100% Statement Coverage all different statements within the functions need to be tested.

And more detailed: Branch Coverage shows which branches within the statements have been tested.

The highest coverage levels requested by safety standards are shown on the right side of our picture. Here each condition need to be tested.

A very important coverage level for safety critical development is the MC/DC Coverage.

MC/DC stands for Modified Condition/Decision Coverage.

ISO 26262 is the standard of the automotive industry for the functional safety of Road vehicles.

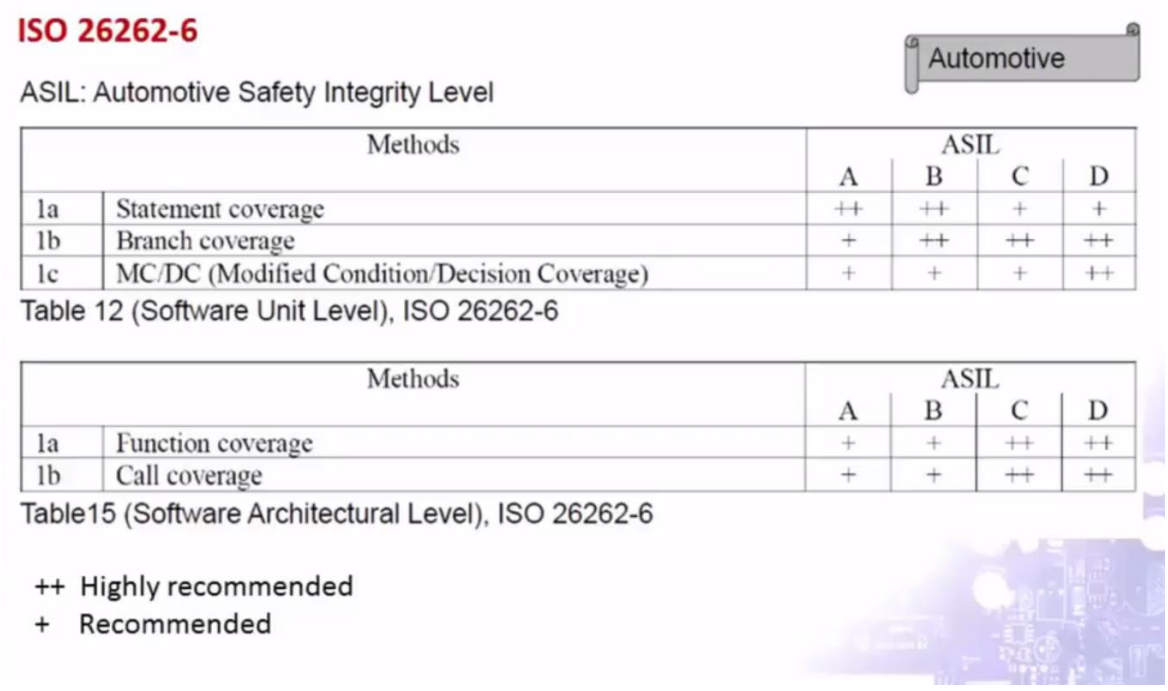
ISO 26262 是适用于道路行驶车辆的汽车工业标准

The required code coverage depends on the ASIL level. ASIL stands Automotive Safety Integrity Level.

For ASIL A only Statement Coverage is “highly recommanded”.

For ASIL B the higher level Branch Coverage is required.

For the highest level ASIL D Modified Condition/Decision Coverage need to be achieved.



To show and prove that you have achieved the requested levels, a good code coverage tool like Testwell CTC++ is needed.

Now let's also have a look to other standards and their requirements in terms of code coverage:

There is the standard IEC 61508 titled ***Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems***

IEC 61508 标准是电气/电子/可编程系统的功能安全标准。

DIN EN 61508-3

General Industry

SIL: Safety Integrity Level

Method		SIL 1	SIL 2	SIL 3	SIL 4
...	...	...	...	...	...
7a	Function Coverage	++	++	++	++
7b	Statement Coverage	+	++	++	++
7c	Branch Coverage	+	+	++	++
7d	MC/DC	+	+	+	++

Table B.2 from DIN EN 61508-3

++ Highly recommended
+ Recommended

IEC 61508 is a basic functional safety standard applicable to all kinds of industries.

According to the different Safety Integrity Levels (also referred as SIL levels) different coverage levels need to be achieved.

For SIL 1 (which is a relatively low level), only Function Coverage is "highly recommended".

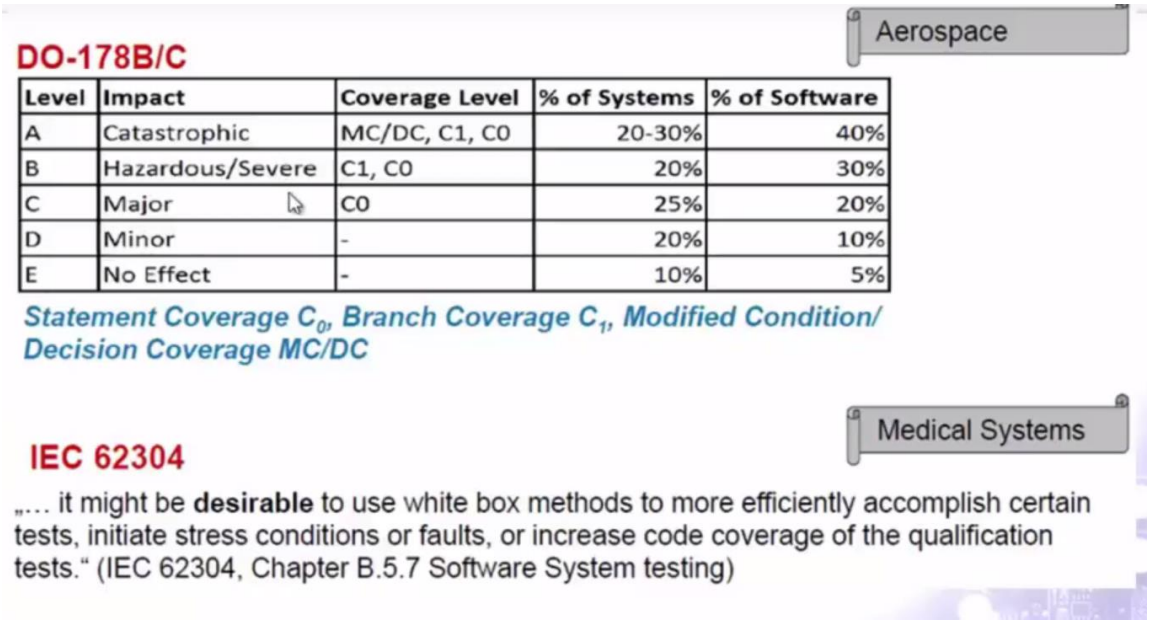
For SIL 2 Statement Coverage needs to be achieved (which includes Function Coverage) -> as we remember 100% Statement Coverage is 100% Function Coverage).

For the highest Safety Integrity Level (SIL 4) MC/DC coverage is "highly recommended".

And highly recommended stands for mandatory in this standards...

The Modified Condition/Decision Coverage or short MC/DC coverage is also very important in the aerospace standard DO-178C.

MC/DC 逻辑覆盖或者短的 MC/DC 逻辑覆盖对宇航标准 DO-178C 同样非常重要。



As the other standards, DO-178C requires high coverage levels for high safety levels.

The different DO-178C-levels are defined according to the possible consequences of a software error:

For Level A a software error would have a catastrophic impact

for Level B it would be Hazardous/Severe

for level C the impact is still Major

a software error in Level D software has only Minor effect

and for level E, there is no Effect

And there is no surprise, for the highest level A, 100% of MC/DC coverage is mandatory.

For level B, C1 coverage which is branch coverage needs to be achieved.

And for level C (major impact), all statements have to be tested.

The standard IEC/EN 62304 for medical devices is a bit faint about code coverage:

“... it might be desirable to use white box methods to more efficiently accomplish certain tests, .. or increase code coverage of the qualification tests”.

But also here, you can be sure, that you will run into problems when software errors remain in the software because it has not been tested properly and this causes injuries or deaths.

Table A.21 – Test Coverage for Code

Technique / measure	Reference	SIL 0	SIL 1	SIL 2	SIL 3	SIL 4
Statement		R	HR	HR	HR	HR
Use the Coverage module to report Statement Coverage for the executed Unit Tests and/or monitored application runs – on the host, simulator and/or target platform.						
Branch		-	R	R	HR	HR
Use the Coverage module to report Decision/Branch Coverage for the executed Unit Tests and/or monitored application runs – on the host, simulator and/or target platform.						
Compound Condition		-	R	R	HR	HR
Use the Coverage module to report Condition Coverage for the executed Unit Tests and/or monitored application runs – on the host, simulator and/or target platform.						
Path		-	R	R	HR	HR
Use the Coverage module to report Path Coverage for the executed Unit Tests and/or monitored application runs – on the host, simulator and/or target platform.						

And the last standard for today, EN 50128 / IEC 62279 for railway applications has also similar requirements in terms of code coverage.

Now, how can Testwell CTC++ Test Coverage Analyser help you with all this standards?

Well, we analyse for all requested code coverage levels,

Function coverage,

Decision coverage,

Statement Coverage,

Condition Coverage,

Modified Condition/Decision Coverage,

and with Multicondition Coveerage we do even more...

Testwell CTC++ is an excellent choice when it comes to code coverage.

I will show you why:

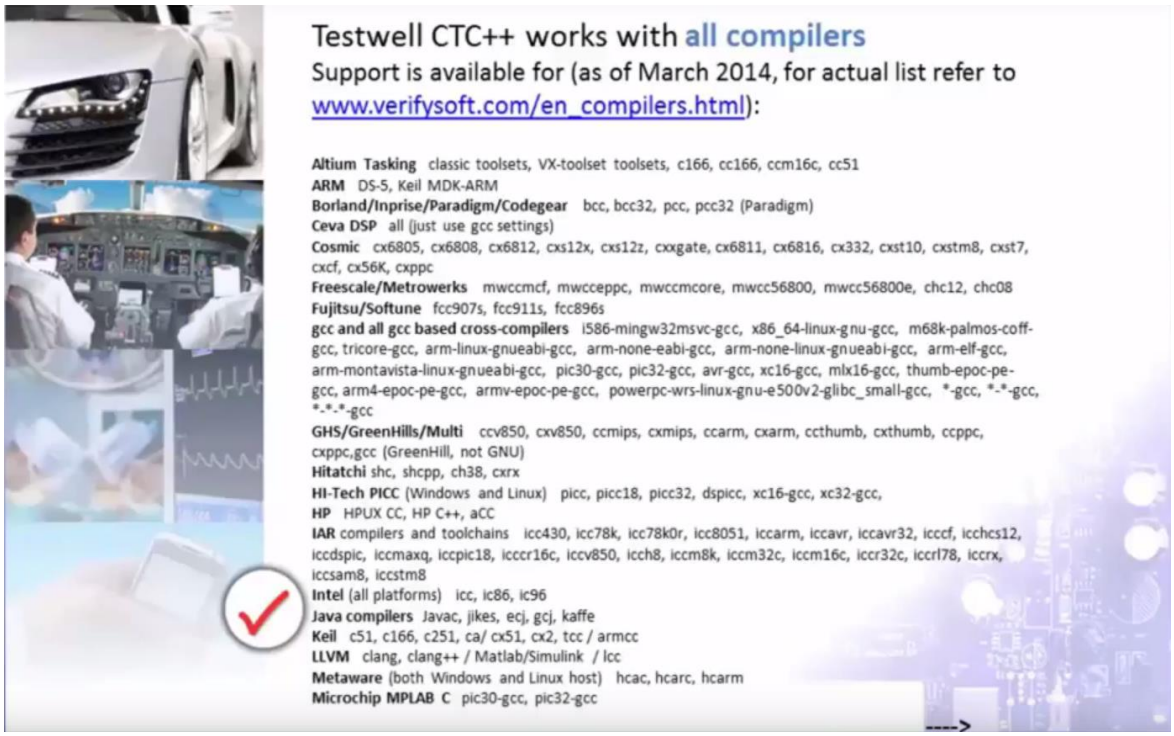
First Testwell CTC++ works with all compilers.

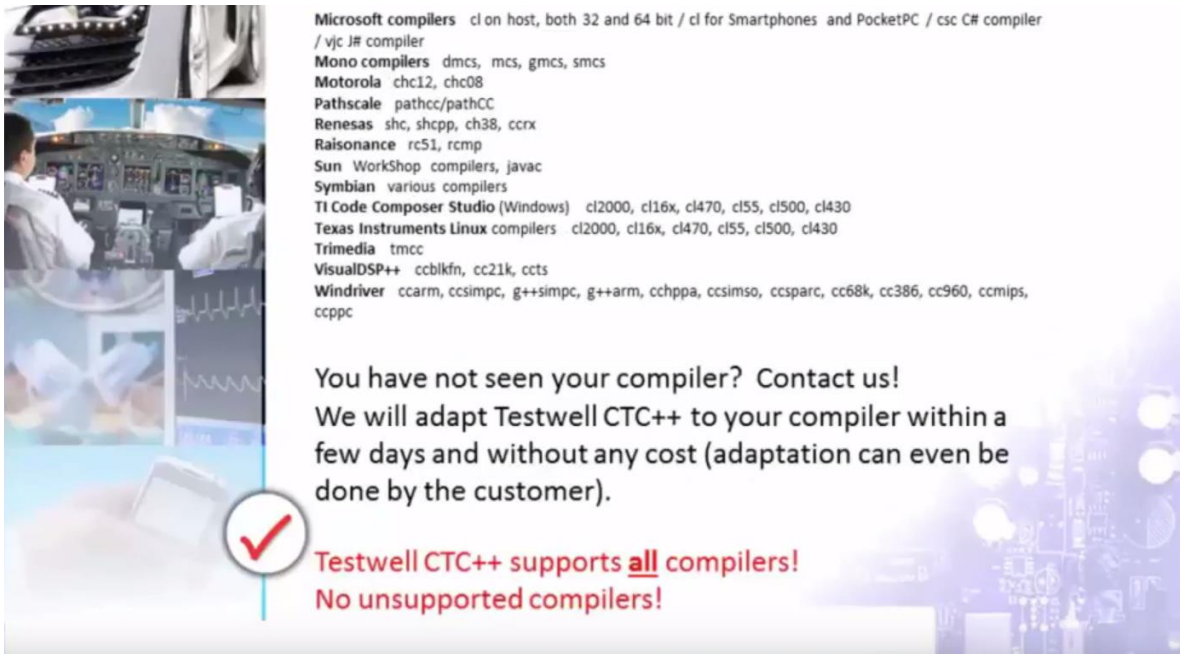
You see here a list of compilers where we have ready settings for

the list is continued here, and every month we add compilers to this list.

But Testwell CTC++ can be adapted easily to any other compiler.

You can do this yourself or we do this for you.



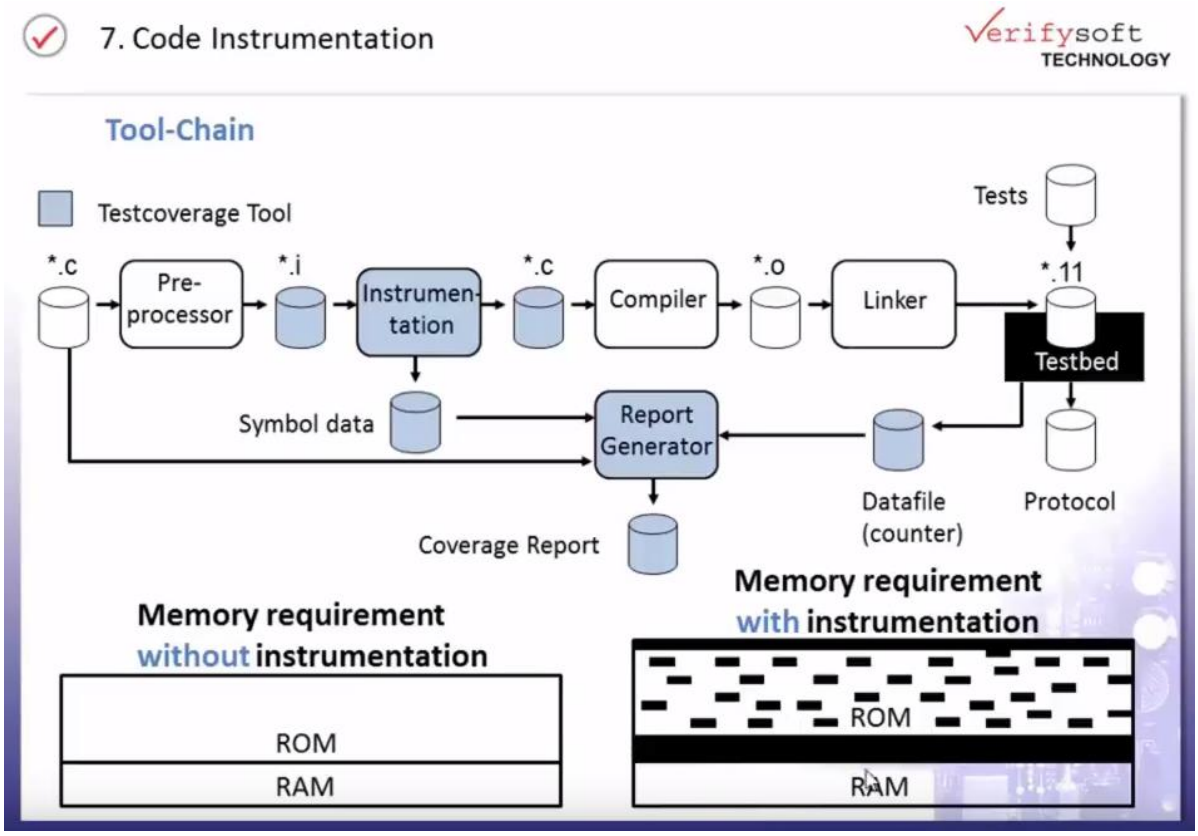


Of course the adaptation is Free of charge for our customers.

Good to know: A CTC++ license covers all compilers – no need to buy a further license if you have a new compiler.

And again: Testwell CTC++ supports all compilers – there is no unsupported compiler.

Now let's have a look to how it works.



Testwell CTC++ does an instrumentation of the code which needs to be tested.

It adds counters to the source code, Stores information about it, Increments the counters with each run, and gives you the analysis of the counter values at the end or on demand.

The code instrumentation is done between pre-processor and compiler.

And we get the coverage report at the end of the process.

The problem – or let's say challenge – is here, that the code instrumentation needs additional memory.

Additional RAM and ROM is needed (as we see at the bottom of this slide).

From our experience, the reason for lack of memory is in most cases the RAM.

But I have good news concerning the memory problem - we are very lucky with Testwell CTC++ because compared to other test coverage tools, the need of supplementary memory is very, very low....

And if in spite of the low memory need of Testwell CTC++ your available memory space is to low, we can use the Bit coverage add-on for CTC++.

With the bit coverage add-on we don't count how many times a piece of code is executed – we count only if it is executed or not.

The usage of the Bit Coverage Add-on reduces the needed RAM by a factor of 32.

For example for function coverage we need only 1 bit supplementary RAM per function instead of 32 bit RAM.

Testwell CTC++ is a tool of Verifysoft Technology, a test tool vendor and distributor located in Offenburg, Germany.

Support and sales is done by our own offices in Germany and France as well as by distributors all over the world.

The development already started in 1989 within the Nokia group.

Three years later the company Testwell was founded in Finland with the mission of the further development of the tool.

Verifysoft Technology was founded 2003 as distributor for Testwell tools in Germany.

After ten years of successful distribution of Testwell tool in Europe, Verifysoft Technology acquired the Testwell tools in 2013 from the Finish company Testwell.

Today hundreds of companies all over the world rely on Testwell CTC++.

For safety critical projects we propose a qualification kit covering the most important safety standards.

Although Testwell CTC++ is mainly used for white box testing to show code coverage of your unit tests, it can be used during the whole software development cycle, which means that it can also cover black box testing.

Let me summarise the advantages of Testwell CTC++ Test Coverage Analyser:

Testwell CTC++ has a very low instrumentation overhead,

It works with all targets and microcontrollers, even with very small ones.

And CTC++ works with all compilers and all cross-compilers.

Testwell CTC++ has different reports in text, XML, JSON and HTML formats.

We see here the summary for a directory

for a file

and here more detailled for a function.

CTC++ Coverage Report - Functions Summary #1/1

Directory Summary | Files Summary | Functions Summary | Untested Code | Execution Profile
To directories: First | Previous | Next | Last | Index | No Index

Directory: C:\Projects\hcontrol
TER: 71 % (44/62) structural, 78 % (49/63) statement

Source file: C:\Projects\hcontrol\regulators.c
Instrumentation mode: multicondition **Reduced to:** MC/DC coverage
TER: 71 % (20/28) structural, 71 % (17/24) statement
To files: Previous | Next

TER % - MC/DC		TER % - statement		Calls	Line	Function
75 % -	(6/8)	83 % -	(5/6)	3	4	lights()
100 %	(2/2)	100 %	(1/1)	1	20	close_windows()
100 %	(2/2)	100 %	(1/1)	2	25	open_windows()
100 %	(2/2)	100 %	(3/3)	1	30	open_windows_for()
100 %	(2/2)	100 %	(1/1)	1	37	heat()
0 % -	(0/2)	0 % -	(0/1)	0	42	air_condition()
60 % -	(6/10)	55 % -	(6/11)	2	47	temperature_control()
71 % -	(20/28)	**71 %** -	(17/24)			**regulators.c**

Source file: C:\Projects\hcontrol\sensors.c
Instrumentation mode: multicondition **Reduced to:** MC/DC coverage
TER: 100 % (8/8) structural, 100 % (4/4) statement
To files: Previous | Next

Within the execution profile you see exactly in your source code which tests are missing.

Let's have a deeper look into it:

Hits/True	False	Line	Source
		1	/* File io.c ----------
		2	#include <stdio.h>
		3	#include "io.h"
		4	/* Prompt for an unsigend int value and return it */
Top			
10		5	unsigned io_ask()
		6	{
		7	unsigned val;
		8	int amount;
		9	
		10	printf("Enter a number (0 for stop program): ");
0	10	11	if ((amount = scanf("%u", &val)) <= 0) {
		12	val = 0; /* on 'non sense' input force 0 */
		13	}
10		14	return val;
		15	}

The white part is your source code, the grey part is the information Testwell CTC++ has added.

In the table we see that the red test is missing.

There is also a report with the untested code.

Once all tests are done, this report will be empty.

You can browse all CTC++ output on our CTC++ page at verifysoft.com

We propose a qualification kit for Testwell CTC++ in order to support you with the qualification process of Testwell CTC++ for your safety critical projects.

The tool qualification demonstrates that Testwell CTC++ works properly and fulfills it task safely. The Qualification Kit covers also the other standards like ISO 26262, DO 178-C, and IEC 61508.

The Tool Qualification Kit for Testwell CTC++ shows you that the tool works properly with your environment.

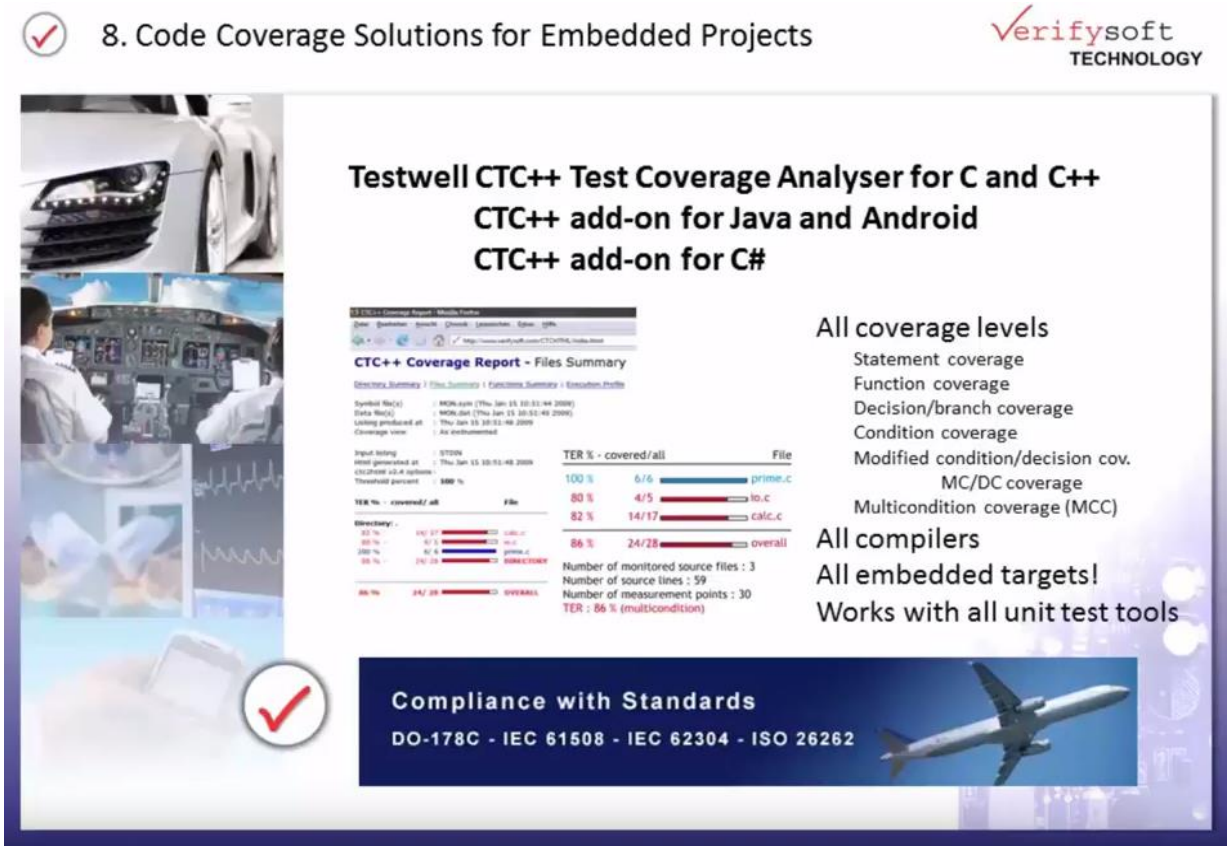
Back to Testwell CTC++ Test Coverage Analyser:

The tool supports all important platforms like Windows, Linux, Solaris and others ...

Testwell CTC++ is also integrated in many IDEs like Visual Studio, Eclipse and others

We have many integrations in tool chains and testing environments CATIA Autosar Builder,dSpace System Desk and TargetLink, dSpace System Desk, TargetLink, QTronic TestWeaver, and many more...

To conclude our presentation, here once more the most important features of Testwell CTC++:



Testwell CTC++ is a code coverage tool for C, C++, Java and C#

It analyses for all coverage levels.

It supports all compilers and all embedded targets.

And it works together with all unit test tools.

Testwell CTC++ complies to safety standards.

Qualification kits are available.

Brief: Testwell CTC++ is your best choice when it comes to code coverage analysis.

Several hundreds of customers in almost 40 countries all over the world use Testwell CTC++ successfully.

We have customers in automotive, aeronautics, healthcare, automation and transportation industries.

We are proud that today all German car-manufactures and nearly all of their software suppliers rely on Testwell CTC++.

Join this companies and secure your software development with Testwell CTC++ Test Coverage Analyser.

We propose a free tool evaluation including access to our support team.

Although Testwell CTC++ is easy to use, we propose trainings to our customers and evaluators.

Further information about our offering is available from our homepage www.verifysoft.com

I hope you enjoyed our presentation.

Thank you very much for your time.

And if you have questions, please do not hesitate to contact Verifysoft or one of our local distributors.

Thanks once more and have a nice day.

For further information please contact:

Verifysoft Technology GmbH, In der Spöck 10-12, 77656 Offenburg, Germany,
www.verifysoft.com

File: Code_Coverage_and_ISO_26262_EN_pictures.pdf