

Call Coverage mit Testwell CTC++ und Imagix 4D

Was ist Call Coverage?

Während für die Function Coverage eines Programmes während des Tests jede Funktion überhaupt einmal aufgerufen werden muss, werden für die Call Coverage für jede Funktion alle *möglichen* Aufrufe betrachtet.

Wird beim Testen jeder Funktionsaufruf ausgeführt, ist volle Call Coverage erreicht. Für jede einzelne Funktion lässt sich die prozentuale Call Coverage als Anteil der tatsächlich ausgeführten Aufrufe an allen vorhandenen Aufrufen messen.

Im Sicherheitsstandard ISO 26262-6 ist die Call Coverage als eine von zwei alternativen Methoden für Integrationstests gefordert (Abschnitt 10.4.6, die Alternative ist die Function Coverage).

Die Call Coverage wird auch als Call Pair Coverage bezeichnet.

Ableitung mit Testwell CTC++

Während der Code Coverage Analyzer Testwell CTC++ die Function Coverage direkt misst, wird die Call Coverage nicht direkt gemessen, lässt sich aber aus stärkeren Abdeckungsmaßen ableiten¹:

Wird für die Tests der gesamten Software 100% Condition Coverage erreicht, dann wurden auch alle Funktionsaufrufe durchgeführt. Damit besteht volle Call Coverage.

Die Messung der Condition Coverage ist für diese Ableitung nur dann nötig, wenn Funktionsaufrufe auch innerhalb zusammengesetzter Bedingungen stattfinden, wie z.B.

```
if (x == 5 && foo())
```

In diesem Beispiel muss die Bedingung `x==5` als wahr ausgewertet werden, damit der Aufruf von `foo` an dieser Stelle erfolgt.

Werden innerhalb solcher Bedingungen keine Funktionen aufgerufen, dann reicht 100% Statement Coverage aus, um die Call Coverage nachzuweisen.

Detaillierte Analyse mit Imagix 4D

Durch den Einsatz des grafischen Analysewerkzeugs Imagix 4D lässt sich die Auswertung der Call Coverage noch verfeinern. Dazu bietet Imagix 4D ein Plug-In, um Berichte aus Testwell CTC++ zu integrieren und auszuwerten.

Für die Call Coverage steht ein eigener Bericht zur Verfügung, der für jede Funktion die Call Coverage als prozentualen Anteil ermittelt. Zusätzlich wird jeder *nicht* erfolgte Aufruf ausgegeben (Spalte „Caller“ mit „Line“ und „File“ Angabe am Ende).

¹ Diese Ableitung gilt strenggenommen nicht, wenn Funktionszeiger oder vergleichbare Konstrukte in C++ verwendet werden. Im Kontext der ISO 26262-6 Forderung wird aber auch dann ein höheres Coverage-Maß geeignet sein, um eine der vergleichsweise schwachen Maße Function bzw. Call Coverage zu ersetzen.

File	Callee	Caller	Func.Cov.	Call Cov.	(Line)
					Line File
myhome.c	main	Function has no callers	73%	-	(10)
regulators.c	air_condition	temperature_control	0%	0%	(42) 69 regulators.c
	close_windows	main	100%	50%	(20) 28 myhome.c
	heat	main	100%	50%	(37) 31 myhome.c
	lights	main main	50%	34%	(4) 27 myhome.c 60 myhome.c
	open_windows	main temperature_control	100%	34%	(25) 50 myhome.c 65 regulators.c
	open_windows_for		100%	100%	(30)
	temperature_control		50%	100%	(47)
sensors.c	strange_smell		100%	100%	(13)
	temperature_inside	main	100%	67%	(3) 29 myhome.c

In zwei Fällen ist diese erweiterte Analyse von besonderem Interesse:

- Im Rahmen der Integrationstest ist es unter Umständen nur mit hohem Testaufwand möglich, vollständige Statement Coverage oder sogar Condition Coverage zu erreichen. Mit Imagix 4D und dem Testwell CTC++ Plug-In lässt sich dieser zusätzliche Aufwand vermeiden.
- Solange keine vollständige Call Coverage erreicht ist, lassen sich mit dem integrierten Bericht direkt die Ursachen dafür ermitteln – die fehlenden Funktionsaufrufe werden ausgegeben und können in den Quellcode zurückverfolgt werden.
Über die anteilige Messung der Call Coverage entsteht ein quantitativer Eindruck, wie weit die Tests von der vollständige Call Coverage noch entfernt sind.