### 2.0.3 (April 2017)

\* [FIX] Fixed issue with stubbing multiple return values and then accessing

them in parallel. (#282, #294). Thanks to Janusz Białobrzewski (jbialobr) for

implementing this, and to Alexandr Nikitin and Alex Povar for reviewing and

contributing to Janusz' PR.

\* [UPDATE] .NET Core build now targets netstandard-1.3 (was 1.5). Thanks to

Ian Johnson (@ipjohnson) for this change. (#303)

\* [UPDATE] .NET Core build is now signed, which removes a warning when

referencing .NET standard library. Thanks to Connie Yau (@conniey) for

this change. (#302)

\* [FIX] Removed redundant .NET standard dependencies when referencing

NSubstitute from as a .NET Framework library. Thanks to Alex Povar (@zvirja). (#295)

### 2.0.2 (February 2017)

\* Dropped Release Candidate tag now that Castle.Core 4.0.0 final has been

released.

\* [UPDATE] Updated all builds to use Castle Core 4.0.0 release.

\* [NEW] Task-compatible ReturnsNull extension method added to ReturnsExtensions (#270). Thanks to Michael Wolfenden for implementing this.

\* [NEW] Support for delegates with out/ref parameters (#271, #273). Thanks to Oleg Sych for raising this, and to Alexandr Nikitin for this fix. (Implemented for NET40/45/Core, not for NET35 at present)

### 2.0.1 RC (December 2016)

\* [NEW] Initial custom handler support (#259). Mainly for integration with libraries like AutoFixture, this allows some basic fallback logic to be injected into a substitute's call handling pipeline. Thanks to Alex Povar (@zvirja) for implementing this and incorporating review comments, and also to Marcio Rinaldi (@mrinaldi) for his initial work on this in #234. Also thanks to @alexandrnikitin for a related spike (#156), and his work with Alex and Marcio in refining this feature and reviewing and merging the final PR.

\* [UPDATE] Improved concurrency support, relating to how NSubstitute tracks the last call and queued call specifications (#264). Thanks a lot to Alex Povar (@Zvirja) for designing and implementing this improvement. Thanks to Alexandr Nikitin for reviewing this PR.

### 2.0.0 RC (August 2016)

\* Build: 2.0.0.0

\* [NEW] Initial .NET Core support. Thanks to @alexandrnikitin for tonnes of work on this. Thanks also to Peter Jas (@jasonwilliams200OK) for both .NET Core code contributions and help with CI setup. And thanks to everyone who contributed to the conversation and testing in https://github.com/nsubstitute/NSubstitute/pull/197.

\* [NEW] Clear substitute configuration using `sub.ClearSubstitute(ClearOptions)`. This extension can be brought in to scope via `using NSubstitute.ClearExtensions;`. Thanks to @trullock for defining and implementing this change, and to @alexandrnikitin for review feedback. (#157, #235, #179)

\* [DOC] Improved CallInfo.ArgAt documentation, thanks to asbjornu. (#231)

\* [FIX] Fix an issue with Received.InOrder when an exception is thrown. Thanks to @m3zercat for finding and fixing this. (#237)

### 1.10.0 (March 2016)

\* [NEW] Callbacks builder for more control over When..Do callbacks. Thanks to bartoszgolek. (#202, #200)

\* [NEW] Auto-substitute for IQueryable<T>. Thanks to emragins. (#67)

\* [FIX] Fix bug when showing params arguments for value types (#214)

\* [FIX] Fix bug when showing params arguments for Received.InOrder calls (#211)

### 1.9.2 (October 2015)

\* [UPDATE] Mark Exceptions as [Serializable]. Thanks to David Mann. (#201)

\* [FIX] Fix bug with concurrently creating delegate substitutes. Thanks to Alexandr Nikitin. (#205)

### 1.9.1 (October 2015)

\* [FIX] Fix bug introduced in 1.9.0 that made checking a call was Received() clear previously stubbed values for that call.

### 1.9.0 (October 2015)

\* [NEW] Allow awaiting of async methods with Received()/DidNotReceive(). Thanks to Marcio Rinaldi for this contribution. (#190, #191)

\* [NEW] Task-specific Returns methods to make it easier to stub async methods. Thanks to Antony Koch for implementing this, and thanks to Marius Gundersen and Alexandr Nikitin for the suggestion and discussion regarding the change. (#189) (Also thanks to Jake Ginnivan who tried adding this back in #91, but I didn't merge that part of the PR in. Sorry!)

\* [NEW] ReturnsForAll<T> extension method. Thanks to Mike Hanson for this contribution. (#198, #196)

### 1.8.2 (May 2015)

\* [NEW] Convenience .ReturnsNull() extensions in NSubstitute.ReturnsExtensions. Thanks to Michal Wereda for this contribution. (#181)

\* [NEW] CallInfo.ArgAt<T>(int) added to match argument at a given position and cast to the required type. Thanks to @jotabe-net for this contribution. (#175)

\* [NEW] Convenience Throws() extensions in NSubstitute.ExceptionExtensions (sub.MyCall().Throws(ex)). Thanks to Michal Wereda for this contribution. Thanks also to Geir Sagberg for helpful suggestions relating to this feature. (#172)

### 1.8.1 (December 2014)

\* [FIX] Fix for methods returning multidimensional arrays. Thanks to Alexandr Nikitin. (#170)

### 1.8.0 (November 2014)

\* [NEW] Convenience methods for throwing exceptions with When-Do. Thanks to Geir Sagberg for this contribution.

\* [FIX] Throw exception when arg matcher used within Returns. (#149)

### 1.7.2 (March 2014)

\* [FIX] Basic support for types that return dynamic. Thanks to Alexandr Nikitin. (#75)

\* [NEW] Auto-subbing for observables. Thanks to Paul Betts.

### 1.7.1 (January 2014)

\* [FIX] Ambiguous arg exception with out/ref parameters. Thanks to Alexandr Nikitin. (#129)

### 1.7.0 (January 2014)

\* [NEW] Partial subs (Substitute.ForPartsOf<T>()). Thanks to Alexandr Nikitin for tonnes of hard work on this feature (and for putting up with a vacillating project owner :)).

\* [UPDATE] Received.InOrder moved out of Experimental namespace.

\* [FIX] Argument matching with optional parameters. Thanks to Peter Wiles. (#111)

\* [FIX] Argument matching with out/ref. Thanks to Kevin Bosman. (#111)

\* [FIX] The default return value for any call that returns a concrete type that is purely virtual, but also has at least one public static method in it will be a substitute rather than null. Thanks to Robert Moore (@robdmoore) for this contribution. (#118)

### 1.6.1 (June 2013)

\* [FIX] Detect and throw on type mismatches in Returns() caused by Returns(ConfigureOtherSub()).

\* [FIX] Support raising exceptions that do not implement a serialisation constructor (#110). Thanks to Alexandr Nikitin for this contribution.

### 1.6.0 (April 2013)

\* [NEW] .AndDoes() method for chaining a callback after a Returns(). (#98)

\* [FIX] Handling calls with params argument of value types, thanks to Eric Winkler.

\* [FIX] Can now substitute for interfaces implementing System.Windows.IDataObject, thanks to Johan Appelgren.

\* [UPDATE] Improved XML doc comments, thanks to David Gardiner.

### 1.5.0 (January 2013)

\* [EXPERIMENTAL] Asserting ordered call sequences

\* [FIX] Arg.Invoke with four arguments now passes fourth arg correctly (#88). Thanks to Ville Salonen (@VilleSalonen) for finding and patching this.

\* [FIX] Substitute objects now use actual implementation for base object methods (Equals, GetHashCode, ToString). Thanks to Robert Moore (@robdmoore) for this contribution. (#77)

\* [NEW] Auto-substitute for Task/Task<T>. Task<T> will use substitute rules that T would use. Thanks to Jake Ginnivan (@JakeGinnivan) for this contribution.

\* [NEW] Match derived types for generic calls (#97). Thanks to Iain Ballard for this contribution.

\* [NEW] Returns now supports passing multiple callbacks, which makes it easier to combine stubbing multiple return values followed by throwing an exception (#99). Thanks to Alexandr Nikitin for this contribution.

### 1.4.3 (August 2012)

\* [FIX] Updated to Castle.Core 3.1.0 to fix an issue proxying generic methods with a struct constraint (#83).

### 1.4.2 (July 2012)

\* [FIX] Issue using NET40 build on Mono (due to NET45 build tools incompatibility)

### 1.4.1 (June 2012)

\* [FIX] Fix matching Nullable<T> arguments when arg value is null. Thanks to Magnus Olstad Hansen (@maggedotno) for this contribution. (#78)

\* [UPDATE] Updated to Castle.Core 3.0.0.

### 1.4.0 (May 2012)

\* [NEW] [BREAKING] Auto-substitute for types returned from substitutes of delegates/Funcs (follows same auto-substitute rules as for methods and properties). Thanks to Sauli Tähkäpää for implementing this feature. (#52)

\* [NEW] Show details of params arguments when displaying received calls. Thanks to Sauli Tähkäpää for implementing this feature. (#65)

\* [FIX] Race condition between checking received calls and building the exception could cause nonsensical exception messages like "Expected 5, actually received 5" when called concurrently. (#64)

### 1.3.0 (Nov 2011)

\* [NEW] Support for Received(times) to assert a call was received a certain number of times. Thanks to Abi Bellamkonda for this contribution. (#63)

\* [FIX] Improved support for calling substitutes from multiple threads. (#62)

### 1.2.1 (Oct 2011)

\* [FIX] Some combinations of Arg.Do and Returns() caused incorrect values to be returned. (#59)

\* [UPDATE] WCF ServiceContractAttribute no longer applied to proxies. (#60)

\* [FIX] Passing null could cause argument actions to fail. (#61)

\* [FIX] Calls to virtual methods from constructors of classes being substituted for are no longer recorded, to prevent non-obvious behaviour when calling Returns() on an auto-substituted value. (#57)

### 1.2.0 (Sep 2011)

\* [NEW] Arg.Do() syntax for capturing arguments passed to a method and performing some action with them whenever the method is called.

\* [NEW] Arg.Invoke() syntax for invoking callbacks passed as arguments to a method whenever the method is called.

\* [NEW] Basic support for setting out/ref parameters.

\* [FIX] Property behaviour for indexed properties (Issue #53)

\* [UPDATE] [BREAKING] Auto-substitute for pure virtual classes, including common ASP.NET web abstractions. Use .Returns(x=>null) to explicitly return null from these members if required. Thanks to Tatham Oddie for original idea and patch, and Krzysztof Kozmic for suggesting and defining which classes would be safe to automatically proxy.

\* [UPDATE] Changed layout of package for eventual support for multiple framework targets.

\* [FIX] Failure to match calls with ref arguments using ReceivedWithAnyArgs().

\* [FIX] Incorrect ambiguous args exception when supplying value type arg specs for object arguments.

### 1.1.0 (May 2011)

\* [UPDATE] Updated to Castle.Core 2.5.3.

\* [FIX] Fixed bug when raising a delegate event with a null argument.

\* [FIX] CallInfo.Arg<T>() now works more reliably, including for null arguments.

\* [FIX] Better exception when accidentally calling substitute extension method with a null reference (e.g. foo.Received().Call() when foo is null)

\* [UPDATE] Exceptions thrown in custom argument matchers (Arg.Is<T>(x => ...)) will now silently fail to match the argument, rather than allowing exceptions to bubble up.

\* [NEW] Support for test fixtures run in parallel.

### 1.0.0 (Dec 2010)

\* [FIX] Using Returns(null) for value types throws, rather than returning default(T).

0.9.5 Release Candidate

\* [FIX] Fixed bug when trying to return null from a call to a substitute.

\* [FIX] Equals() for class substitutes fixed by not intercepting Object methods Equals(), ToString() and GetHashCode().

\* [NEW] Raise.Event<THandler>() methods to raise any type of event, including delegates.

\* [BREAKING] Raise.Action() methods removed. Use Raise.Event<THandler>() (e.g. Raise.Event<Action>>()).

\* [BREAKING] Renamed Raise.Event<TEventArgs>() methods to Raise.EventWith<TEventArgs>().

\* [UPDATE] Arg matchers can be specified using more specific, compatible type (Arg.Is<string>(..) for arg of type object).

\* [NEW] NSubstitute website and documentation http://nsubstitute.github.com

\* [FIX] Formating for argument matchers that take predicate functions.

\* [FIX] Match single argument matcher passed to params arg (#34)

\* [FIX] Detect ambiguous arg matchers in additional case (#31)

\* [FIX] Can modify event handler subscriptions from within event callback

\* [UPDATE] Update to Castle.Core 2.5.2

\* [FIX] Can substitute for SynchronizationContext in .NET4 (fixed in Castle.Core)

\* [NEW] NSubstitute available as NuPack package

### 0.9.0 Beta 1

\* [FIX] Now handles argument specifiers used for params arguments correctly

\* [UPDATE] Updated to Castle.Core 2.5 final.

### 0.1.3 alpha 4

\* [NEW] Support auto/recursive substituting for members that return interfaces or delegates.

\* [NEW] Support auto substituting for members that return arrays and strings (return empty values rather than null).

\* [NEW] Raise.Event<TEventArgs>() will now attempt to construct arguments with default ctors, so in most cases they will not need to be explictly provided.

\* [UPDATE] Added support for raising events with custom delegate types.

\* [UPDATE] Formatting for event subscription and unsubscription calls in call received/not received exceptions.

\* [UPDATE] Updated to pre-release build of Castle.Core 2.5 to get dynamic proxy to support modopts.

\* [FIX] Throw correct exception when raising an event and event handler throws. (Fix by Rodrigo Perera)

\* [FIX] Record call as received when it throws an exception from the When..Do callback.

### 0.1.2 alpha 3

\* [NEW] Marked non-matching parameters in the actual calls listed for CallNotReceivedException messages.

\* [NEW] Added WhenForAnyArgs..Do syntax for callbacks.

\* [UPDATE] Updated arg matching to be smarter when matchers are not used for all args.

\* [FIX] Fixed bug when substituting for delegates with multiple parameters.

\* [FIX] Removed redundant cast operator which sometimes caused the compiler trouble in resolving Raise.Event().

### 0.1.1 alpha 2

\* [NEW] Added ReturnsForAnyArgs() extension methods

\* [FIX] Compiled for Any CPU to run on x64 platforms

### 0.1.0 alpha

\* Initial release