QuickCheck Design Specification	
Author	Nick Battle
Date	16/12/24
Issue	0.1

0. Document Control

0.1. Table of Contents

0.2. References

[1] Wikipedia entry for The Vienna Development Method, http://en.wikipedia.org/wiki/Vienna_Development_Method

[2] Wikipedia entry for Specification Languages, http://en.wikipedia.org/wiki/Specification_language

[3] VDMJ, https://github.com/nickbattle/vdmj

[4] Overture, https://github.com/overturetool/overture

0.3. Document History

Issue 0.1	16/12/24	First draft.

0.4. Copyright

1. Overview

This design describes the QuickCheck plugin for the VDMJ tool.

Section 1 gives an overview of the architecture and the Java package structure. Section 2 gives detailed information about each package. Section 3 walks through various common scenarios to describe the interaction of the internals.

1.1. VDMJ

VDMJ provides basic tool support for the VDM-SL, VDM++ and VDM-RT specification languages, written in Java [1][2][3]. It includes a parser, a type checker, an interpreter (with arbitrary precision arithmetic), a debugger, a proof obligation generator and a combinatorial test generator with coverage recording, as well as *JUnit* support for automatic testing and user definable annotations.

1.2. VDMJ Plugins

1.3. Proof Obligations

1.4. QuickCheck Package Overview

The implementation is divided into the following Java packages.

Packages	

1.4.1. Comments

2. Package Detail

This section gives more detail about the Java classes in the packages within QuickCheck.

3. Scenarios

This section describes the sequence of actions that occur during common tasks with QuickCheck. The intention is to provide a more tangible description of how the classes described in section 2 work together in practice.

3.1. Server Creation and Lifecycle

3.1.1. Comments