**Name of System**: Bourne-Again Shell

**Reviewer**: Kelvin Chung

**Date**: 11-14-2011

**Author of Software**: Brian Fox

**Author of Book Chapter**: Chet Ramey

**Five-Star Rating**: ★½

**Purpose of System**

The Bourne-Again Shell, or bash, is a reimplementation of the Bourne shell using concepts of other shells. The name was chosen as a tribute to Stephen Bourne (creator of the original shell), as well as the concept of the reimplementation being a “spiritual rebirth”. It is currently being used as the default shell in Linux and Mac OS X, and has been ported to virtually every environment that has a UNIX shell.

Bash is itself backwards-compatible with Bourne shell scripts, to the extent that the only difference is in behavior that is newly-introduced to bash. However, it is POSIX-compliant and built with that in mind, so this is another source of some incompatibility. Bash has been criticized by some as been “not open enough” due to its long release times, as well as its glacial changes in architecture due to the needs of its users.

**Basic Metrics**

**KLOC**: 250K lines

**Project Start-Up**: 1989

**Number of Major Releases**: Currently at version 4.2.

**Number of Developers**: Appears to be solo, data appears to be unreliable

**Size of User Community/Number of Installations**: Too many to count. It is the default on many computers.

**Major Stakeholders**: GNU Project, as bash is used as the default shell. To a lesser extent, anyone who uses Linux or Mac OS

**Use of concurrency**: [INC – Not known]

**Implementation Language**: C++

**Supporting Software**: GNU readline

**High Level Architecture**

**Diagram of Software Architecture**:



**High Level Scenarios**: Interactive and non-interactive input is largely the same – only the first box differs in whether GNU readline or its own internal tools are used to lex and parse the input.

**Data Structures or Algorithms**: All data structures used by bash are integers, shallow arrays, and shallow associative arrays.

**Control Flow / Key Data**: Nothing beyond what is described.

**Architectural Style**: Pipe and filter is heavily used in bash.

**Major Evolutionary Changes**: The architecture has changed relatively little since Ramey’s takeover in 1994, as it was described as “feature-complete” at that point.

**Performance Bottlenecks**: Keyword expansion phases may have a performance bottleneck from poor interpretation.

**Real-time**: No parts require critical response times, unless bash as a whole is itself used for real-time applications.

**Notation for Architecture**: Data flow diagrams are exclusively used in the chapter.

**Methodology**: It would appear to be a waterfall or large spiral methodology, as there has been no significant changes for some time.

**Kruchten’s Attributes**

**Size**: Medium-Large (250K lines of code)

**Criticality**: High due to its presence in what’s considered to be operating system code. Actual usage may determine its criticality

**Age of System**: Extra Large (20+ years)

**Rate of Change**: Low. Data is unreliable due to inaccurate statistics tracking.

**Business Model**: Open Source, though at times it appears to be in-house.

**Stable Architecture**: Low. Considered to be “feature-complete”.

**Team Distribution**: Low. Appears to be a solo project at this point.

**Governance**: Low. Appears to be a solo project at this point.