

# Specification-Based Testing: "cat" Example

Conrad Hughes  
School of Informatics

Slides thanks to Stuart Anderson



## Overview

- Here we consider testing the UNIX "cat" command using the category-partition method we've been looking at.

## Cat man page

- NAME:** cat - concatenate files and print on the standard output
- SYNOPSIS:** cat [OPTION] [FILE]...
- DESCRIPTION:** Concatenate FILE(s), or standard input, to standard output.
  - A, --show-all equivalent to -vET
  - b, --number-nonblank number nonblank output lines
  - e equivalent to -vE
  - E, --show-ends display \$ at end of each line
  - n, --number number all output lines
  - s, --squeeze-blank never more than one single blank line
  - t equivalent to -vT
  - T, --show-tabs display TAB characters as ^I
  - u (ignored)
  - v, --show-nonprinting use ^ and M- notation, except for LFD and TAB
  - help display this help and exit
  - version output version information and exitWith no FILE, or when FILE is -, read standard input.
- EXAMPLES:**
  - cat f - g Output f's contents, then standard input, then g's contents.
  - cat Copy standard input to standard output.

## Identifying Independently Testable Features

- Here we might think that there are three ITFs associated with cat:
  - Error checking for the option string - checking the syntax is OK and for consistent combinations of options.
  - Error checking the syntax of filenames.
  - That cat functions correctly given a legal option string and a syntactically correct sequence of filenames
- Here we will consider generating categories and partitions of those categories for the third ITF we have identified.
- For each ITF we begin by identifying:
  - Parameters** relevant to the feature.
  - Other elements of the execution environment that the ITF is dependent on. Typical elements are: databases, the file system, hardware devices, ...

## Identifying parameters relevant to the ITF

- Parameters:**
  - Option string:
  - Filename sequence:
- Environment:**
  - The file system (this is a mapping from valid filenames to file contents):
- The parameters relevant to an ITF are those that cause a change in behaviour in the Feature when they are changed.
- The next stage is to identify categories - these are elementary characteristics of the parameters which are either found explicitly in the specification or they are implicit - i.e. they arise from the experience of the tester.

## Identifying Categories

- Parameters:**
  - Option string:
    - Option string length
  - Filename sequence:
- Environment:**
  - The file system (this is a mapping from valid filenames to file contents):
  - Standard input: