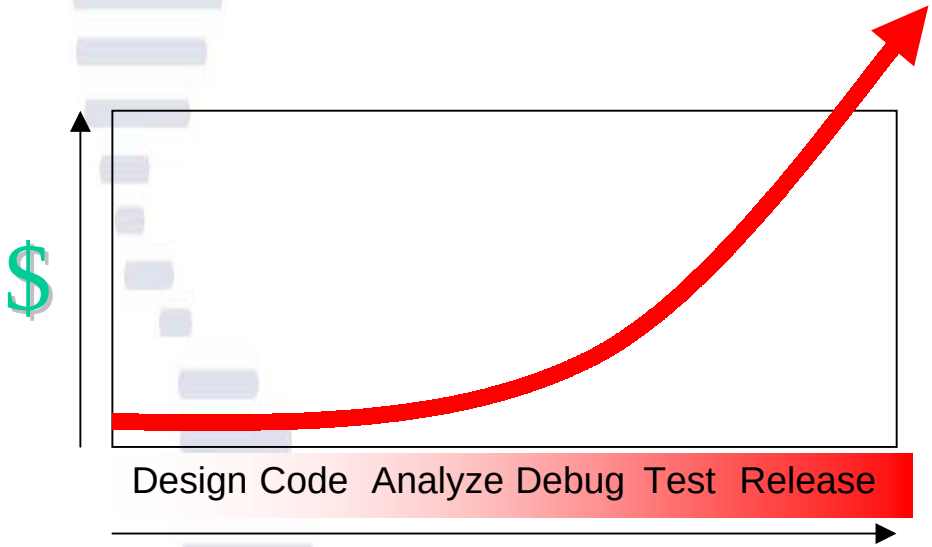
Cleanscape Testwise

Software visualization and analysis tool box

The Cost of Bugs



The cost of fixing a bug goes up dramatically the later a bug is found in the development cycle

- **Cost of Discovery**
  - Very high if the customer finds the bug
  - Lowest if the developer finds it him/herself
- **Cost of Correction**
  - Very high when discovered while running the application
  - Lowest when spotted in the source code

Code

CLEANSCAPE SourceMill

Automatically generate source code by synthesizing object models with templates

Analyze

CLEANSCAPE FortranLint

CLEANSCAPE LintPlus

Stop software problems at their source

Test

CLEANSCAPE Testwise

Integrate redundant test processes into an automated test process

Build & Maintain

CLEANSCAPE qef

Automate the software development process

Manage software construction

CLEANSCAPE
software international

Cleanscape Lint Tools Value Proposition

- **Cleanscape provides software development teams with powerfully simple tools that reduce organizational exposure to risks from latent software problems by automatically identifying problems at their source--in the code prior to compiling or executing programs.**
- **These tools can save software developers hundreds of hours in problem eradication efforts, more than returning their investment on the first use.**

- **Key Values**
  - Shorten the software development cycle
  - Prevent project delays that result from post-compile testing
  - Reduce costs by eliminating problems with cheaper resources earlier in the development process
  - Increase the competitive viability of the software development organization by helping teams produce higher quality products cheaper

Cleanscape Lint Tools

CLEANSCAPE FortranLint

Unix
Windows
Linux
Xlint
Online

CLEANSCAPE LintPlus

Unix
Windows
Linux
Online

- **Classification**
  - Enterprise-class static source code analysis for C and Fortran
- **Function**
  - Identify problems in source code that compilers can't detect
  - Document code
  - Generate call trees & cross-references
  - Web-enabled version available
- **Benefits**
  - Faster
    - Find problems in seconds, not days
    - Prevent project delays that result from post-compile testing
  - Better
    - Produce higher quality products cheaper
  - Smarter
    - Improved processes, standardized reports increase competitive viability of the development organization
  - Cheaper
    - Eliminate coding problems with cheaper resources earlier in the development process
  - Cleaner code

Cleanscape FortranLint

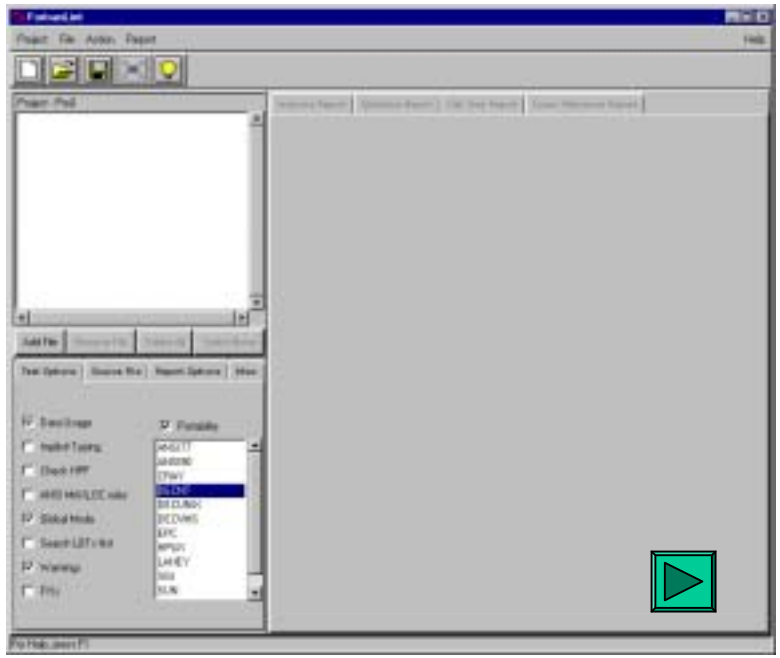
CLEANSCAPE FortranLint

Unix
Windows
Linux
Xlint
Online

CLEANSCAPE LintPlus

Unix
Windows
Linux
Online

- **Classification**
  - Fortran source code analysis tool
- **Function**
  - Processes entire programs written in Fortran for inconsistencies that would prevent applications from running properly
  - Analyzes F77, F90, F95, and many Fortran dialects
  - Conducts local dataflow analysis
  - Performs advanced portability checking
  - Extends static analysis to OpenMP
  - Detects
    - Inconsistencies in variables and argument lists between modules
    - Inconsistencies in common block definitions
    - Non-portable or unused code
    - Unassigned variables or variable type conflicts



Cleanscape Testwise Value Proposition

CLEANSCAPE Testwise

- **Adequate testing can devour about 70% of resources in a typical software development project.**
- **Quality often becomes the first casualty when software developers are faced with tight budget constraints and approaching deadlines.**
- **Software developers need a way to achieve dramatic cost reductions and quality improvements by automatically analyzing the dynamic behavior of software.**

- **Key Value**
  - Testwise allows developers to integrate redundant test processes into an automated test process that helps them identify and eliminate problems early in the development cycle — dramatically reducing resource allocation while increasing software quality.
  - A powerful, yet easy-to-use, software visualization and analysis toolset, Testwise allows software test engineers to automate software testing for faster, better, smarter, cheaper and — of course — *cleaner* software development

Cleanscape Testwise Overview

CLEANSCAPE Testwise

- *x*Atac
- *x*Retress
- *x*Prof
- *x*Slice
- *x*Find
- *x*Vue
- *x*Diff

- **Classification**
  - Software test automation tools for C, C++ programs on Unix, Linux, or Windows platforms.
- **Function**
  - Coverage analysis
  - Regression testing
  - Advanced software maintenance
  - Dynamic debugging
  - Performance analysis
  - Dependency tracing
  - File difference display

- **Benefits**
  - Shortens test and maintenance phases of software while providing
    - Faster time to market
    - Better quality product
    - Stronger product
  - Facilitates cross-platform development by providing common test environment for Unix, Linux, and Windows
  - Aids software development, selection, and improvement of test sets
  - Maximizes productivity
  - Easily standardizes and automates established test processes
  - Eliminates need to build test software (Stubs and Drivers)

Cleanscape Testwise Tool Box

CLEANSCAPE Testwise

*x*Atac
*x*Regress
*x*Prof
*x*Slice
*x*Find
*x*Vue
*x*Diff

Testwise

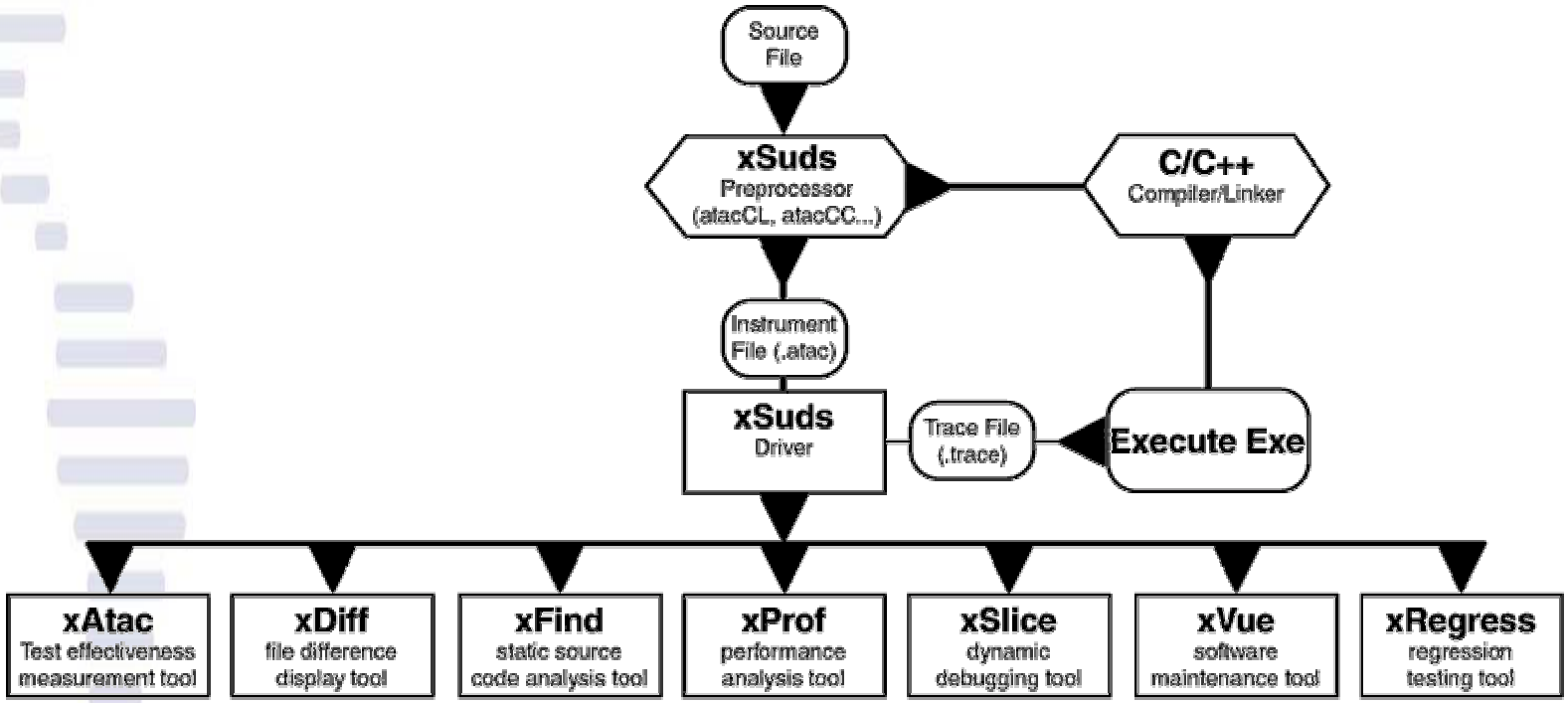
***x*Atac** **Test effectiveness measurement tool**	
***x*Regress** **Intelligent regression testing**	***x*Prof** **Performance analysis**
***x*Slice** **Dynamic debugging**	***x*Find** **Static program dependency tracing**
***x*Vue** **Software maintenance**	***x*Diff** **File difference display**

Software visualization and Analysis Tool Box

Cleanscape Testwise process

CLEANSCAPE Testwise

- ***x*Atac**
- *x*Regress
- *x*Prof
- *x*Slice
- *x*Find
- *x*Vue
- *x*Diff



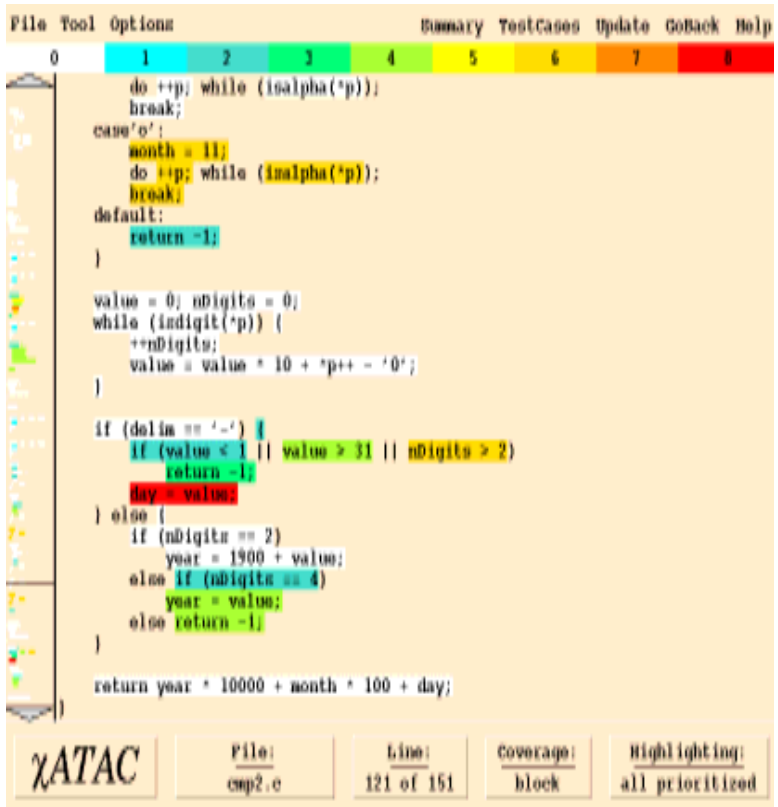
xSuds Software Visualization and Analysis Process

Cleanscape Testwise *x*Atac

CLEANSCAPE Testwise

- ***x*Atac**
- *x*Regress
- *x*Prof
- *x*Slice
- *x*Find
- *x*Vue
- *x*Diff

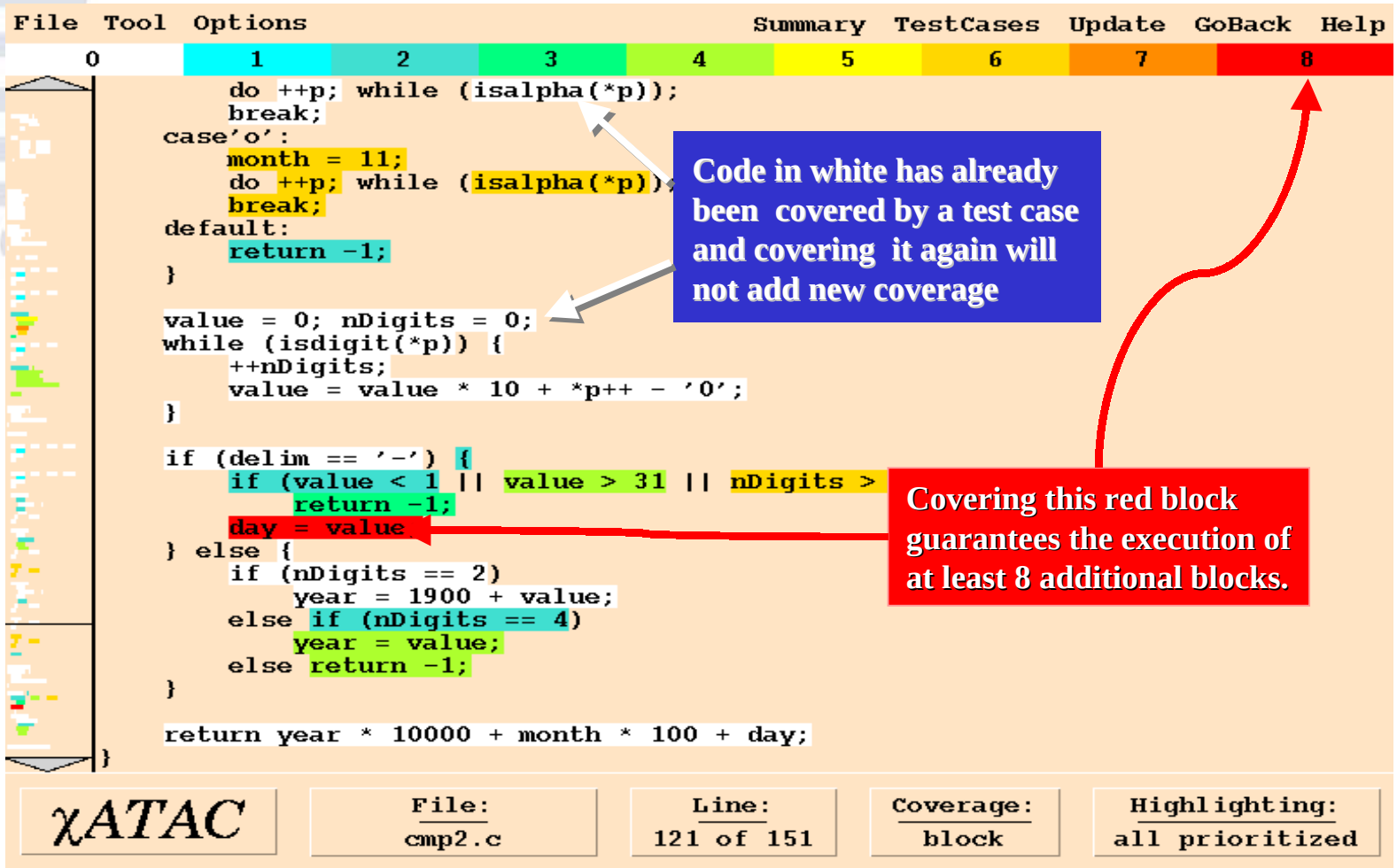
- **Test effectiveness measurement tool**
  - Determines how much of your code is currently tested
  - Facilitates test creation
  - Determines what is missing
  - Identifies redundant test cases
  - Determines whether product testers are finding bugs that have been found and fixed by developers
  - Improves the software testing process



Coverage testing with *x*Atac

CLEANSCAPE Testwise

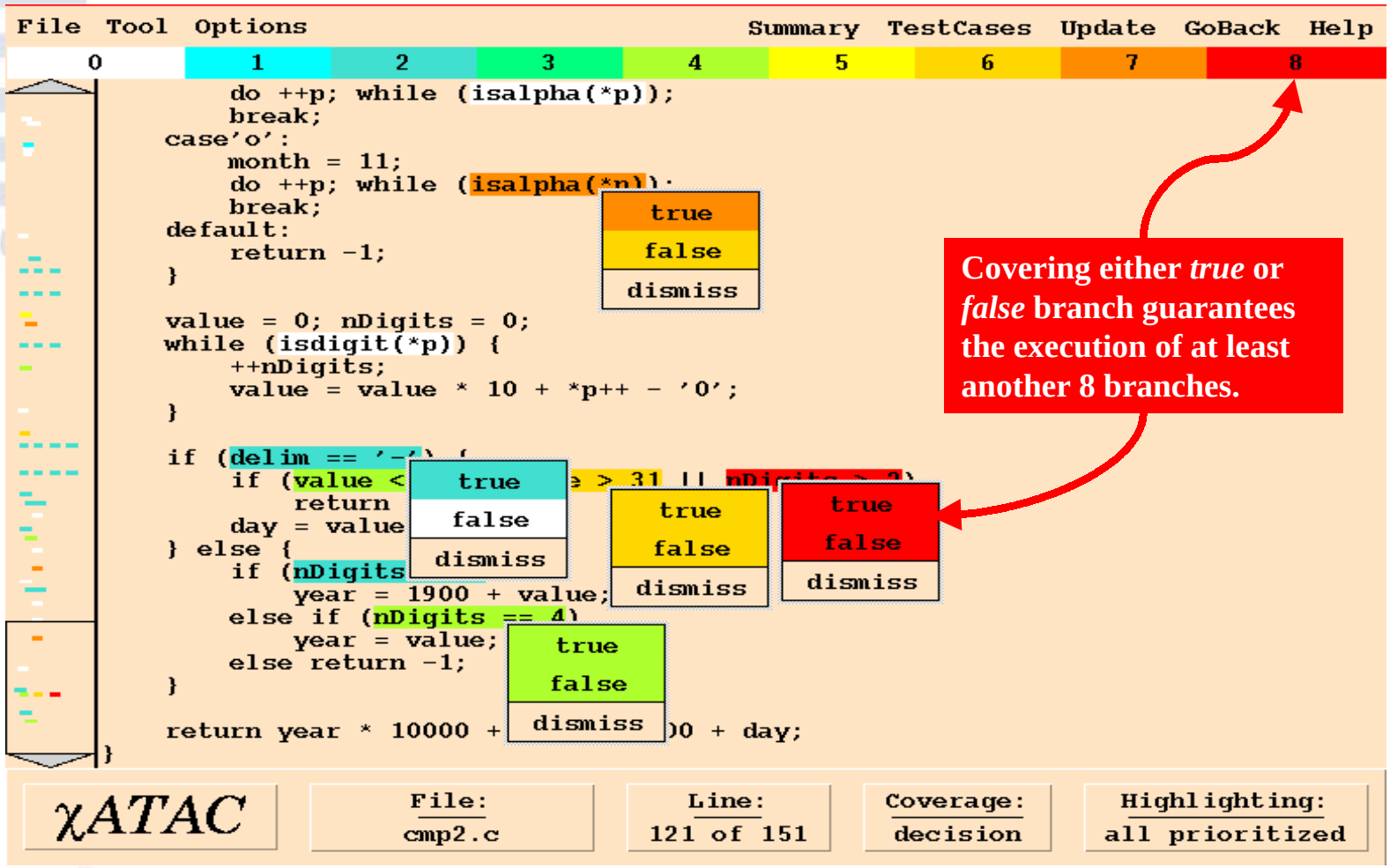
- **xAtac**
- xRegress
- xProf
- xSlice
- xFind
- xVue
- xDiff



Coverage testing with *x*Atac

CLEANSCAPE Testwise

- **xAtac**
- xRegress
- xProf
- xSlice
- xFind
- xVue
- xDiff

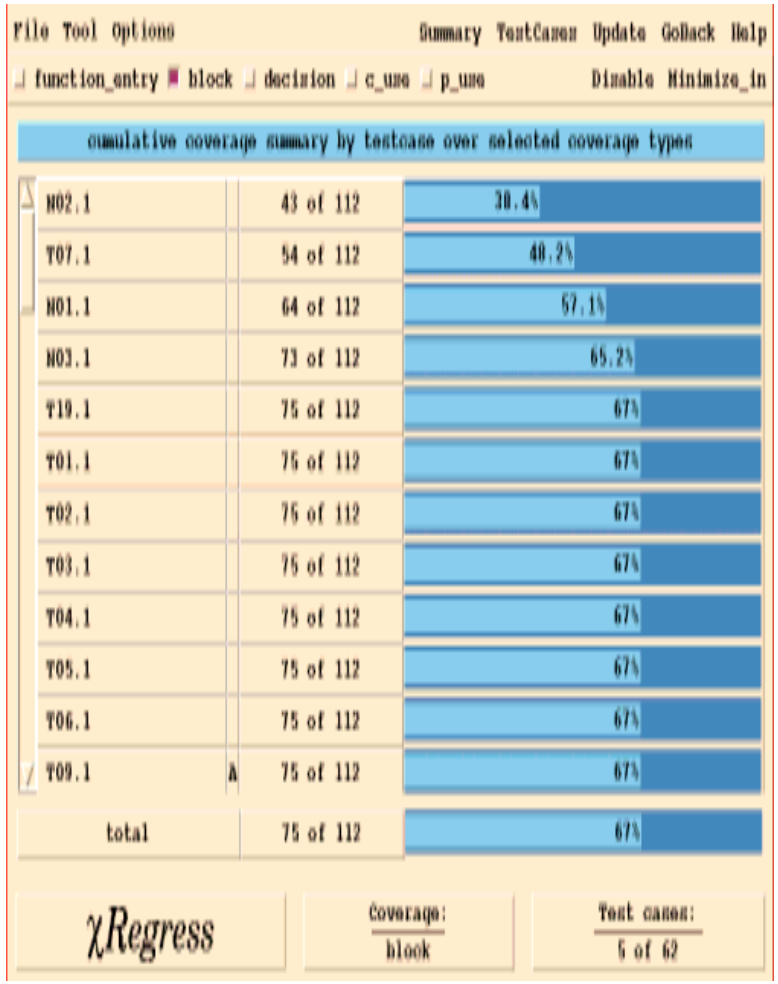


Cleanscape Testwise *x*Regress

CLEANSCAPE Testwise

- *x*Atac
- ***x*Regress**
- *x*Prof
- *x*Slice
- *x*Find
- *x*Vue
- *x*Diff

- **Intelligent regression testing tool**
  - Significantly reduce regression test costs
  - Determine whether you are spending excessive resources in regression testing
  - Understand how to select effective regression tests



Test set minimization with *x*Regress

CLEANSCAPE Testwise

- *x*Atac
- ***x*Regress**
- *x*Prof
- *x*Slice
- *x*Find
- *x*Vue
- *x*Diff

File Tool Options Summary TestCases Update GoBack Help

function_entry block decision c_use p_use Disable Minimize_in

cumulative coverage summary by testcase over selected coverage types

N02.1		43 of 112	38.4%
T07.1		54 of 112	48.2%
N01.1		64 of 112	57.1%
N03.1		73 of 112	65.2%
T19.1		75 of 112	67%
T01.1		75 of 112	67%
T02.1		75 of 112	67%
T03.1		75 of 112	67%
T04.1		75 of 112	67%
T05.1		75 of 112	67%
T06.1		75 of 112	67%
T09.1	A	75 of 112	67%
total		75 of 112	67%

χRegress | Coverage: block | Test cases: 5 of 62

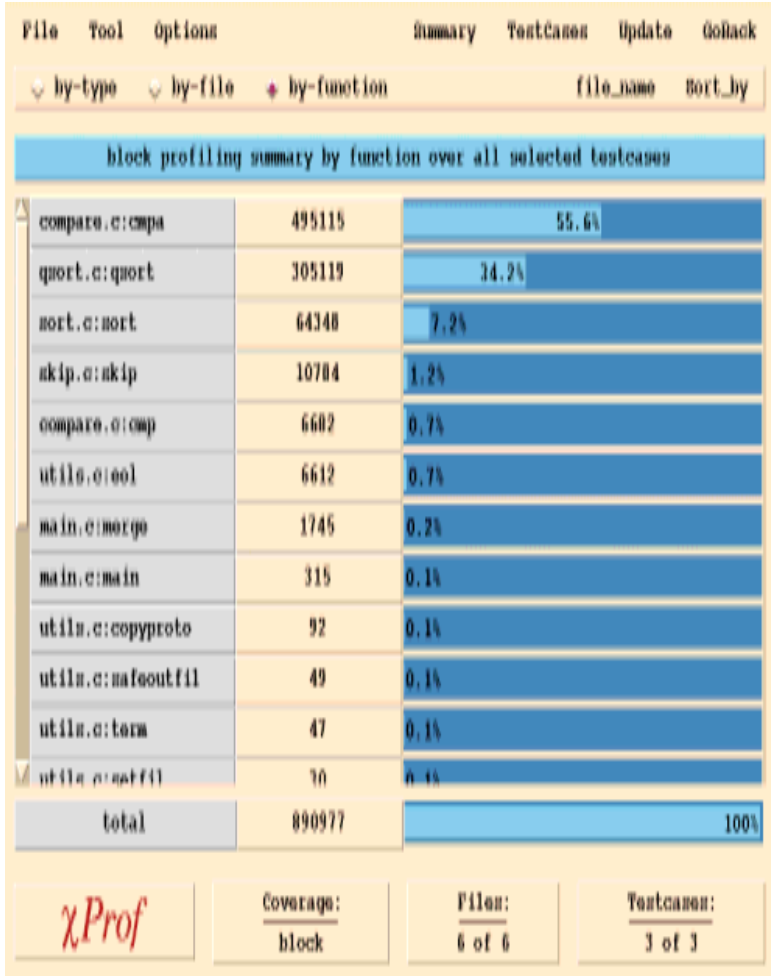
Cleanscape Testwise *x*Prof

CLEANSCAPE Testwise

- *x*Atac
- *x*Regress
- ***x*Prof**
- *x*Slice
- *x*Find
- *x*Vue
- *x*Diff

Performance analysis tool

- Identify performance bottlenecks visually
- Improve the performance of your program
- Identify which part of the program slows execution
- Visualize the most frequently executed pieces in code
- Develop repeatable performance measurements



Identifying frequently executed code with *x*Prof

CLEANSCAPE Testwise

- *x*Atac
- *x*Regress
- ***x*Prof**
- *x*Slice
- *x*Find
- *x*Vue
- *x*Diff

File Tool Options Summary TestCases Update GoBack

by-type by-file by-function file_name Sort_by

block profiling summary by function over all selected testcases

compare.c:cmpa	495115	55.6%
qsort.c:qsort	305119	34.2%
sort.c:sort	64348	7.2%
skip.c:skip	10784	1.2%
compare.c:cmp	6682	0.7%
utils.c:eol	6612	0.7%
main.c:merge	1745	0.2%
main.c:main	315	0.1%
utils.c:copyproto	92	0.1%
utils.c:safeoutfil	49	0.1%
utils.c:term	47	0.1%
utils.c:setfil	30	0.1%
total	890977	100%

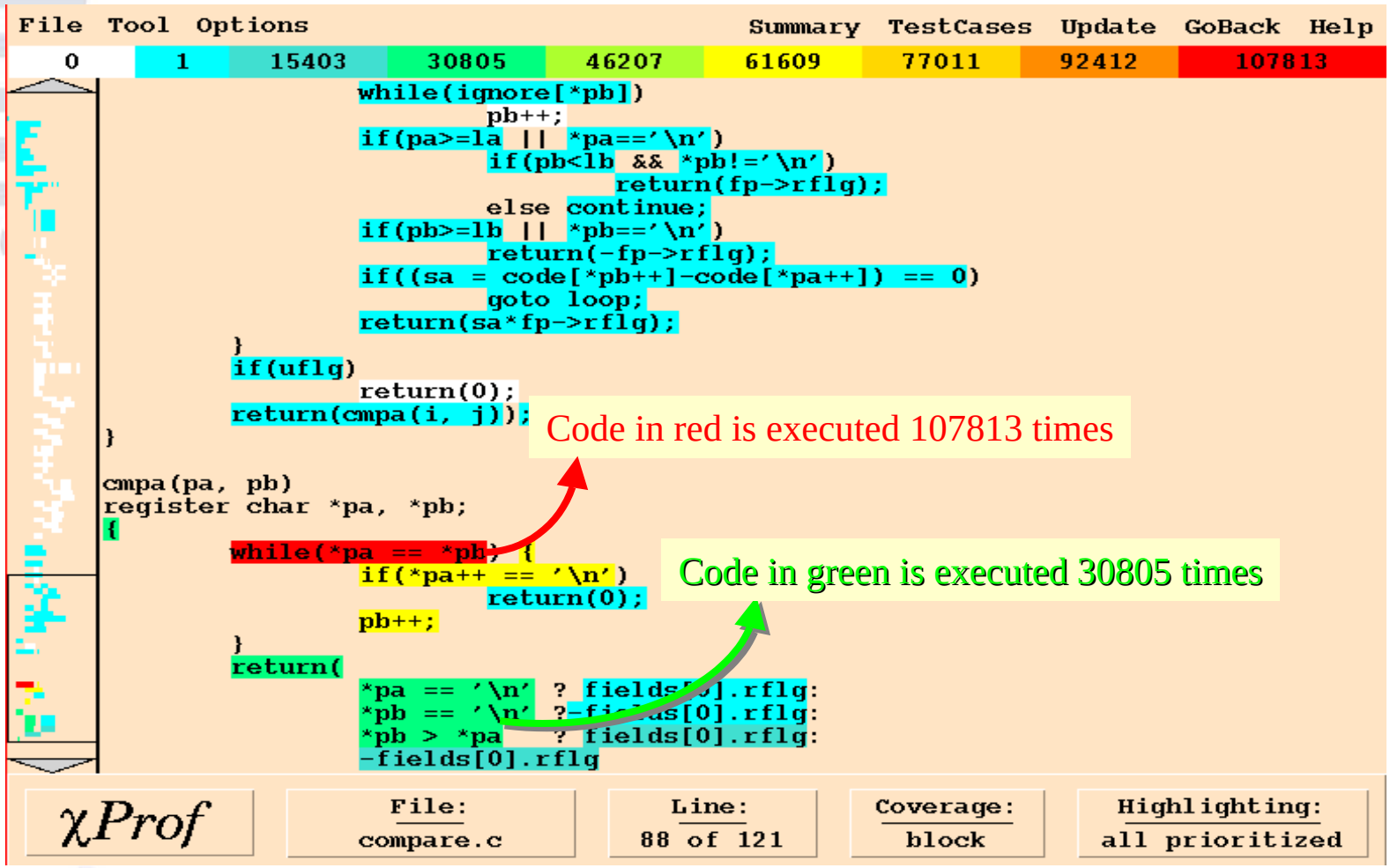
Function *cmpa* is responsible for 55.6% of the block executions.

*x*Prof | Coverage: block | Files: 6 of 6 | Testcases: 3 of 3

Identifying frequently executed code with *x*Prof

CLEANSCAPE Testwise

- *x*Atac
- *x*Regress
- ***x*Prof**
- *x*Slice
- *x*Find
- *x*Vue
- *x*Diff

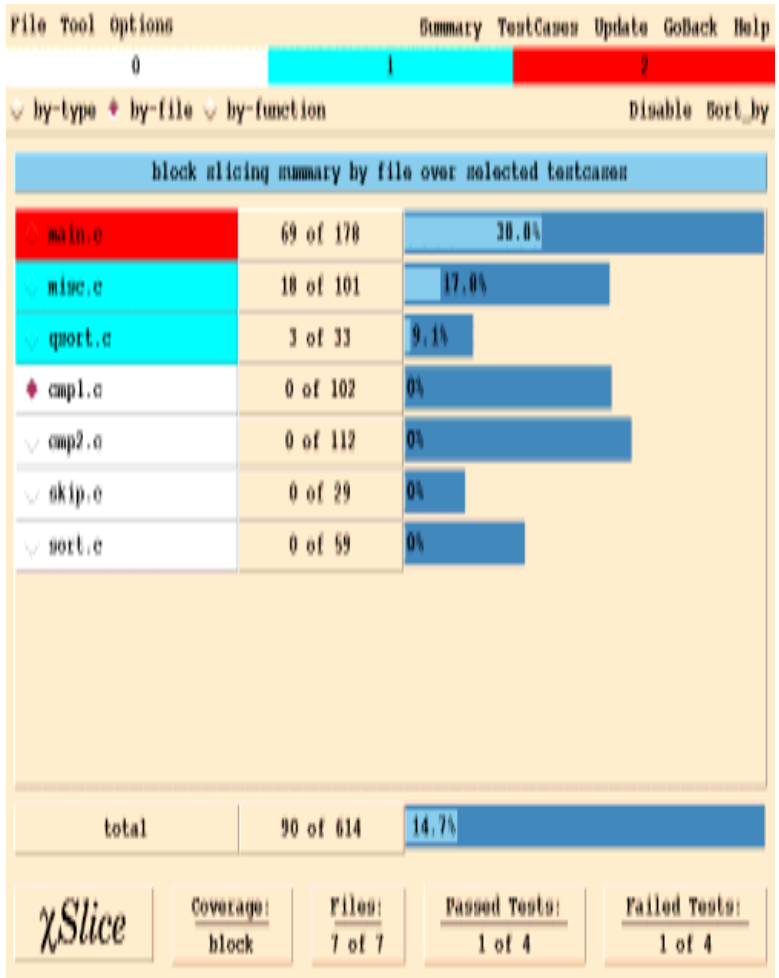


Cleanscape Testwise *x*Slice

CLEANSCAPE Testwise

- *x*Atac
- *x*Regress
- *x*Prof
- ***x*Slice**
- *x*Find
- *x*Vue
- *x*Diff

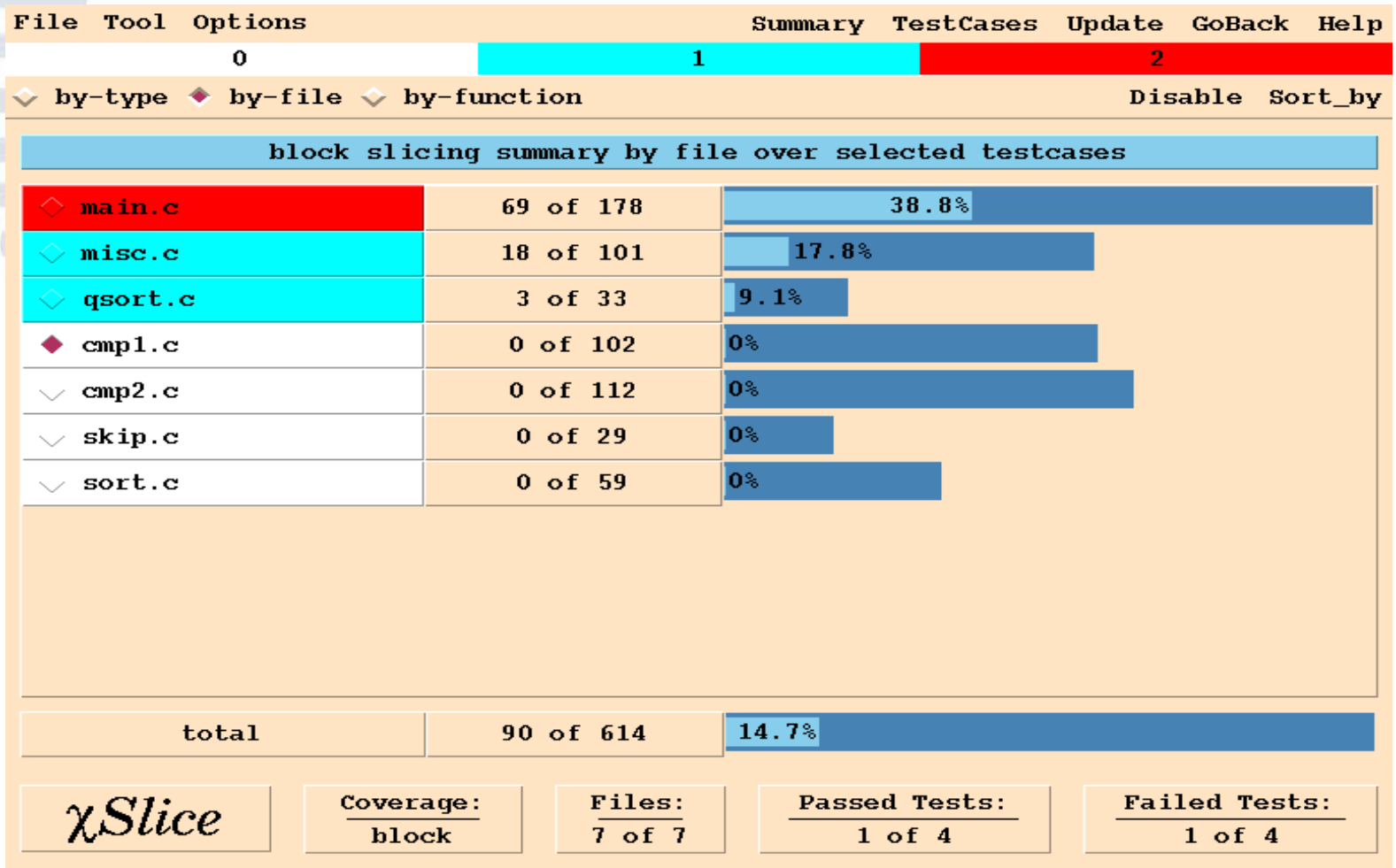
- **Dynamic debugging tool**
  - Pinpoint the location of faults from failures
  - Locate bugs quickly
  - Narrow down bugs to files, then functions, then lines of code



Localizing programming bugs with xSlice

CLEANSCAPE Testwise

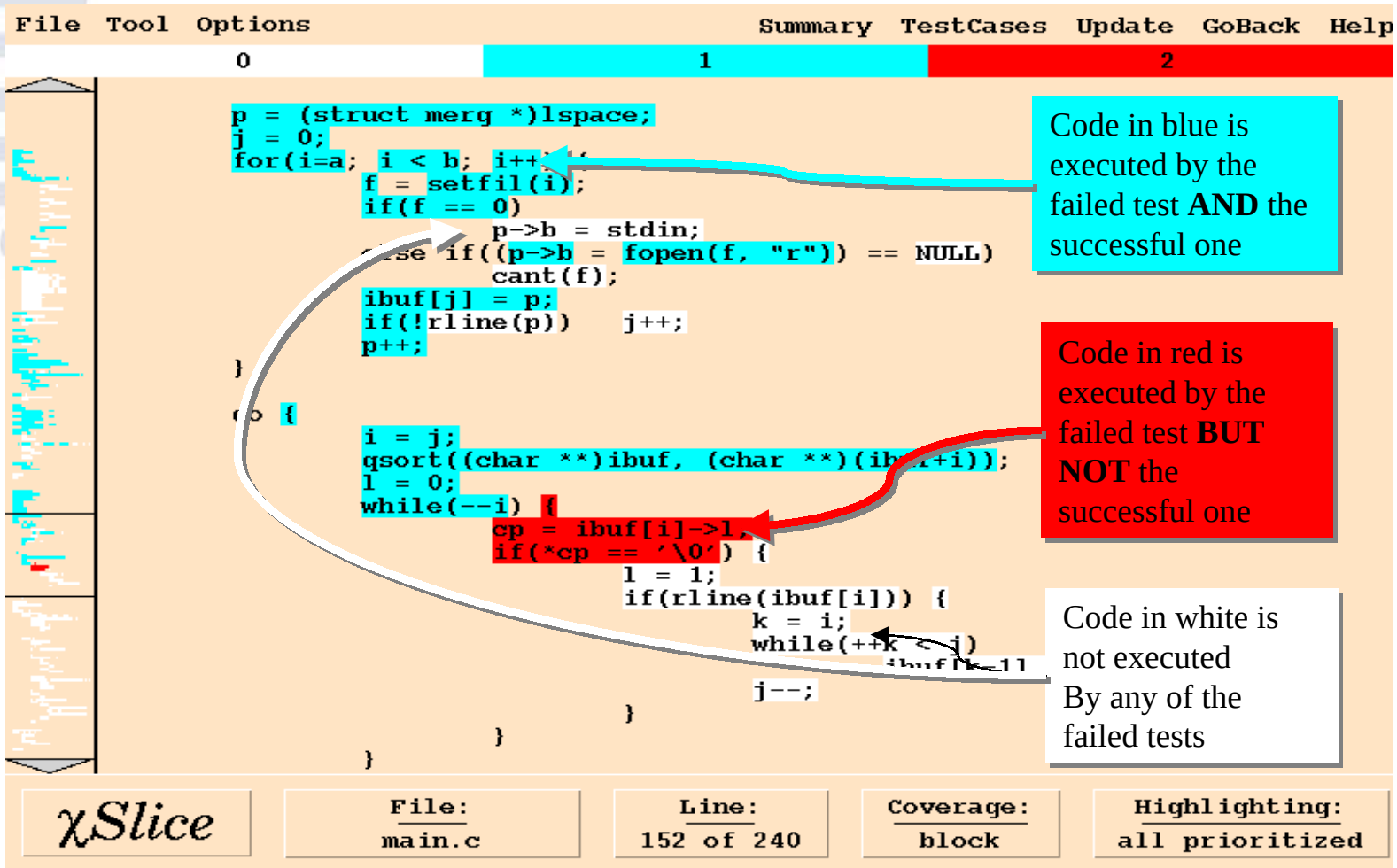
- *x*Atac
- *x*Regress
- *x*Prof
- **xSlice**
- *x*Find
- *x*Vue
- *x*Diff



Localizing programming bugs with *x*Slice

CLEANSCAPE Testwise

- *x*Atac
- *x*Regress
- *x*Prof
- ***x*Slice**
- *x*Find
- *x*Vue
- *x*Diff

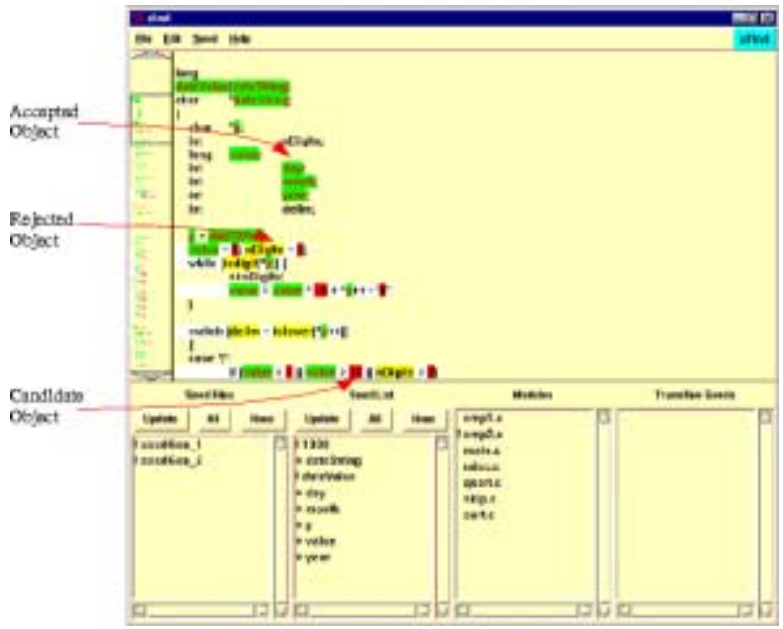


Cleanscape Testwise *x*Find

CLEANSCAPE Testwise

- *x*Atac
- *x*Regress
- *x*Prof
- *x*Slice
- ***x*Find**
- *x*Vue
- *x*Diff

- **Static source code analysis tool**
  - Trace static program dependencies
  - Identify date sensitivities in your applications
  - Analyze difficult languages like C and C++
  - Identify redundant test cases

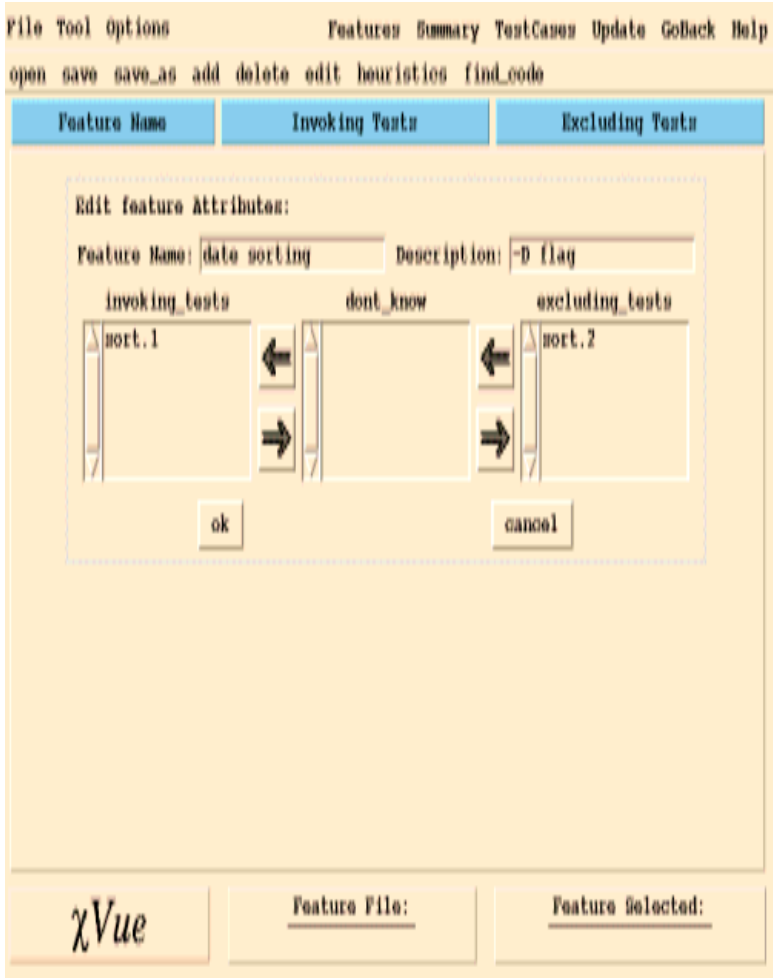


Cleanscape Testwise *x*Vue

CLEANSCAPE Testwise

- *x*Atac
- *x*Regress
- *x*Prof
- *x*Slice
- *x*Find
- ***x*Vue**
- *x*Diff

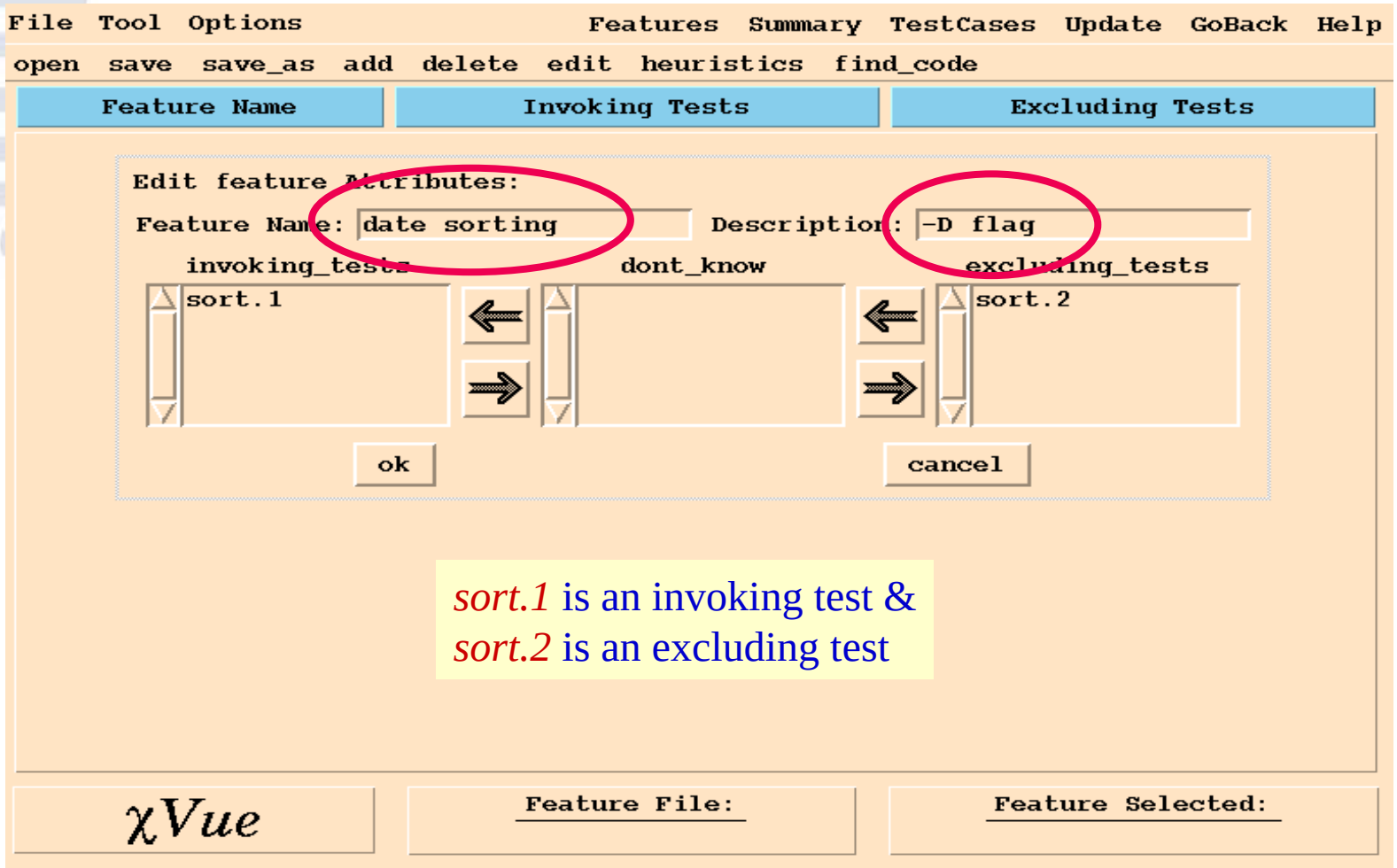
- **Software maintenance tool**
  - See where user functions are implemented in the software
  - Know where features are implemented
  - Visualize features in code
  - Reduce time involved with resolving reports



Visualizing features in code with *x*Vue

CLEANSCAPE Testwise

- *x*Atac
- *x*Regress
- *x*Prof
- *x*Slice
- *x*Find
- **xVue**
- *x*Diff



Visualizing features in code with *x*Vue

CLEANSCAPE Testwise

- *x*Atac
- *x*Regress
- *x*Prof
- *x*Slice
- *x*Find
- ***x*Vue**
- *x*Diff

File Tool Options Features Summary TestCases Update GoBack Help

```
        );
}

long
cmpd(dString)
char    *dString;
{
    char        *p;
    int         nDigits;
    long        value;
    int         day;
    int         month;
    int         year;
    int         delim;

    p = dString;
    value = 0; nDigits = 0;
    while (isdigit(*p)) {
        ++nDigits;
        value = value * 10 + *p++ - '0';
    }

    switch (delim = tolower(*p++))
    {
    case '/':
        if (value < 1 || value > 12 || nDigits > 2)
            return -1;
        month = value;
        break;
    case '.':
```

Code in red is *uniquely* related to the *date sorting* feature.

χVue | File: cmp2.c | Line: 16 of 151 | Coverage: block | Highlighting: highest weight

Cleanscape Testwise *x*Diff

CLEANSCAPE Testwise

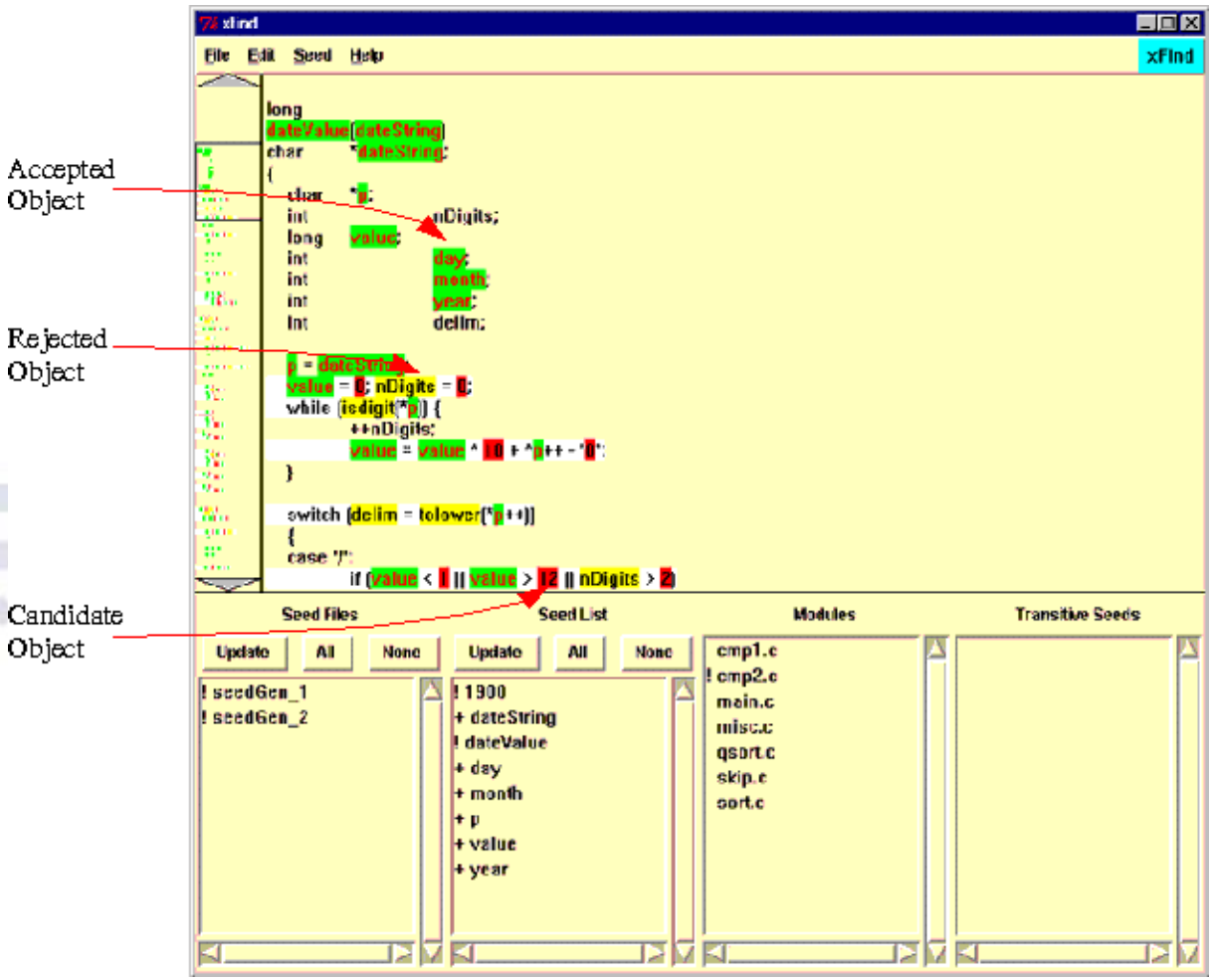
- *x*Atac
- *x*Regress
- *x*Prof
- *x*Slice
- *x*Find
- *x*Vue
- ***x*Diff**

- **File difference display tool**
  - Display program differences
  - Visualize the difference between two files
  - Compare versions of code, documents, data, computer output

Identifying problems in source code with *x*Diff

CLEANSCAPE Testwise

- *x*Atac
- *x*Regress
- *x*Prof
- *x*Slice
- *x*Find
- *x*Vue
- ***x*Diff**



Displaying program differences with *x*Diff

CLEANSCAPE Testwise

- *x*Atac
- *x*Regress
- *x*Prof
- *x*Slice
- *x*Find
- *x*Vue
- ***x*Diff**


CLEANSCAPE
software international