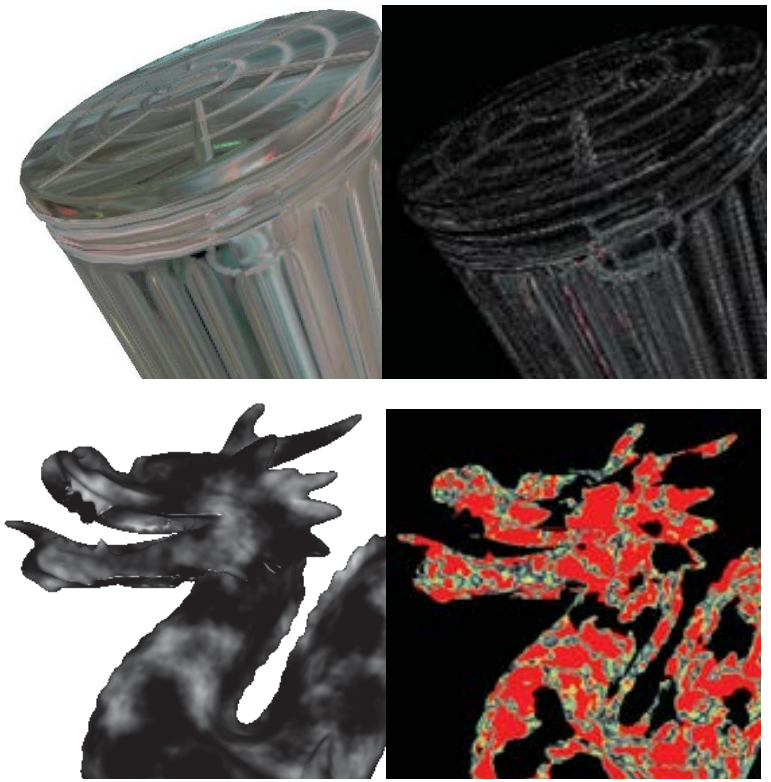

Our research group focuses on advancing software quality by using both static and dynamic programming language approaches. I am particularly concerned with automatic or minimally-guided techniques that can scale and be applied easily to large, existing programs. Finding bugs is insufficient; we work to help programmers address defects, understand error reports, and program correctly. We are also working to design languages and language features to help prevent errors.

Weimer Research Group

Westley Weimer

Associate Professor
weimer@cs.virginia.edu
www.cs.virginia.edu/~weimer/

Department of Computer Science
University of Virginia
Charlottesville, VA
434.924.1021

"Using concepts from other areas of computer science to help address software quality problems."

UNIVERSITY of VIRGINIA
ENGINEERING

Evolutionary Program Repair

A fundamental challenge for computer scientists over the next decade is to produce and maintain systems that have fewer defects and are more resilient to attacks. Software maintenance accounts for over $70 billion/yr and is focused on repairing defects. We are working to reduce the time and effort gaps between finding and fixing software defects by producing a scalable and trustworthy technique to automatically repair bug programs.

Analysis of Non-Executable Artifacts

Even if static analysis eliminate large classes of defects and ensure that software is locally correct, software component integration, evolution and maintenance will still be key security concerns. Security vulnerabilities based on misunderstanding software will soon become low-hanging fruit for attackers. We are developing formal models for, and generating human-readable instances of, software artifacts relating to program documentation, readability, and run-time behavior. These non-executable artifacts are critical for composing, evolving, and maintain critical software. This work will develop techniques for characterizing the trustworthiness of non-executable artifacts in real-world software, as wells as for generating such artifacts in an accurate and complete way.

Analysis of String Variables

An important subclass of software defects is caused by the improper handling of structured text in the form of string variables. Two compelling examples of this type of defect are SQL injection and cross-site scripting, which are now the two most commonly exploited security vulnerabilities. In a future where remotely exploitable vulnerabilities become increasingly costly, it will be crucial for developers to have access to tools that help them analyze string-using code, repair string handling bugs, or prove the absence of string handling bugs in mission-critical code. We are working to develop scalable, expressive, and provably correct constraint solving algorithms for strings, and study the integration of those algorithms into modern satisfiability modulo theories solvers.

Helix Project

The Helix project is a self-regenerative software security architecture for defending computer systems against well-funded and determined attackers. Helix will proactively monitor, adapt and reconfigure software components to present attackers with an ever changing system, thus dramatically raising the expertise and resources required for an attack.

RECENT RESEARCH DEVELOPMENTS

- Developed a computer program that fixes bugs in other computer programs. http://dijkstra.cs.virginia.edu/genprog/
- Decreased the cost of program debugging by 50%, on average.
- Adapted debugging program to other fields, such as the production of optimized and simplified graphics shaders.

RECENT GRANTS

- NSF/CAREER-Scalable and Trustworthy Automatic Program Repair
- NSF-Synthesizing Human-Readable Documentation
- DOD/USAF-Helix--A Self-Regenerative Architecture

SEAS Research Information
Pamela M. Norris, Associate Dean
University of Virginia
Box 400242
Charlottesville, VA 22903
pamela@virginia.edu
434.243.7683

UNIVERSITY of VIRGINIA ENGINEERING