Writing Bug-Free Code Using Theorem Provers

Aaron Stump
Computer Science
The University of Iowa

Coding Bliss

The world is running on code

HealthCare.gov

We are benevolent heroes!

Coding Bloodbath

The world is crashing on code

HealthCare.gov

We are arrogant villains!

What Makes Software Lousy?

- Unclear requirements
  - Bad interfaces, missing features, unnecessary features
- Bad coding style
  - Hard to maintain, lots of cutting and pasting, convoluted
- Bugs
  - Invariants broken, function contracts not met, resources misused

What Makes Software Lousy?

- Unclear requirements
  - Bad interfaces, missing features, unnecessary features
- Bad coding style
  - Hard to maintain, lots of cutting and pasting, convoluted
- Bugs
  - Invariants broken, function contracts not met, resources misused

"Building error-free trading software is impossible, and that makes today's stock markets even more fragile."

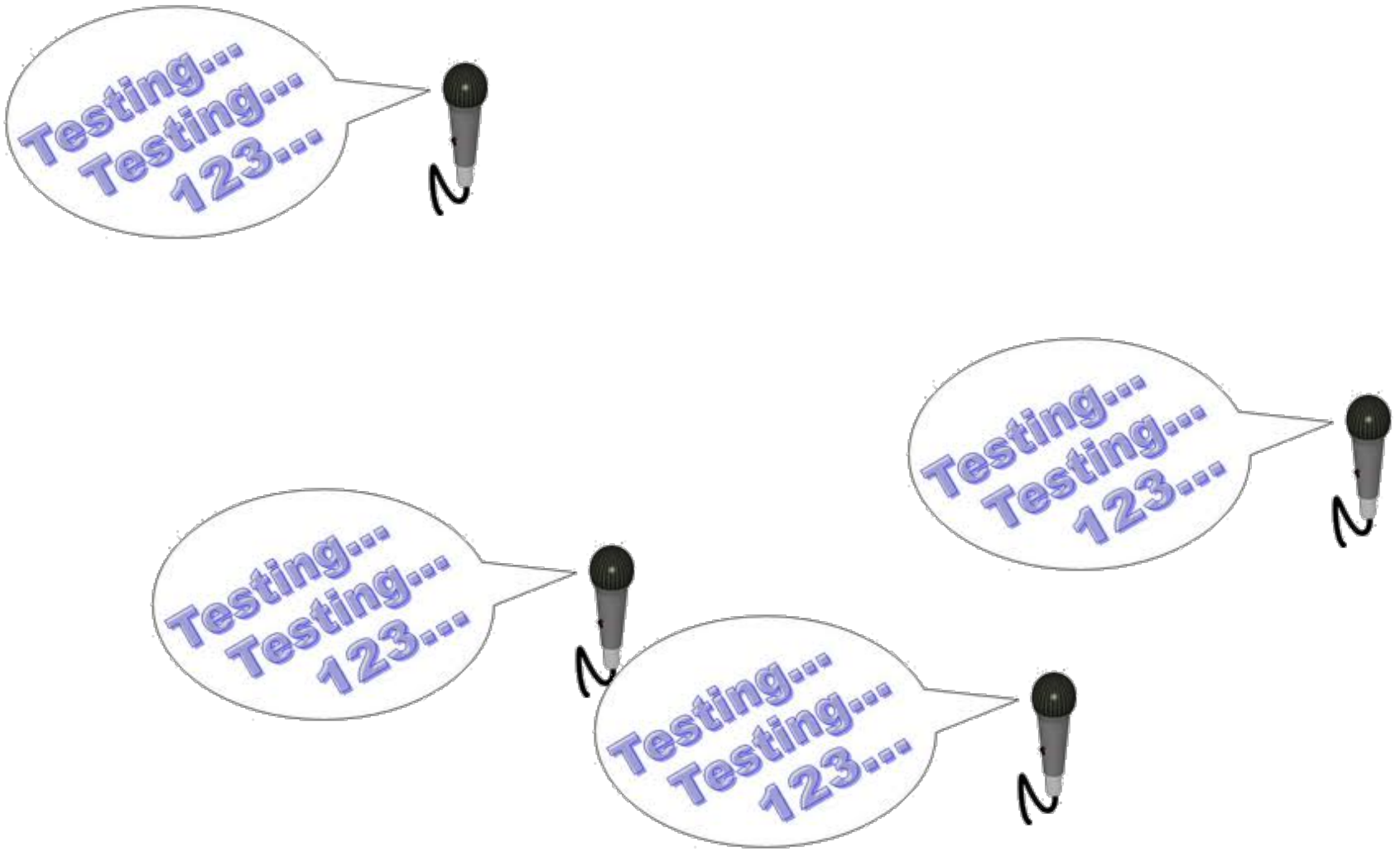
Is error-free code impossible?

Current Practice: Testing

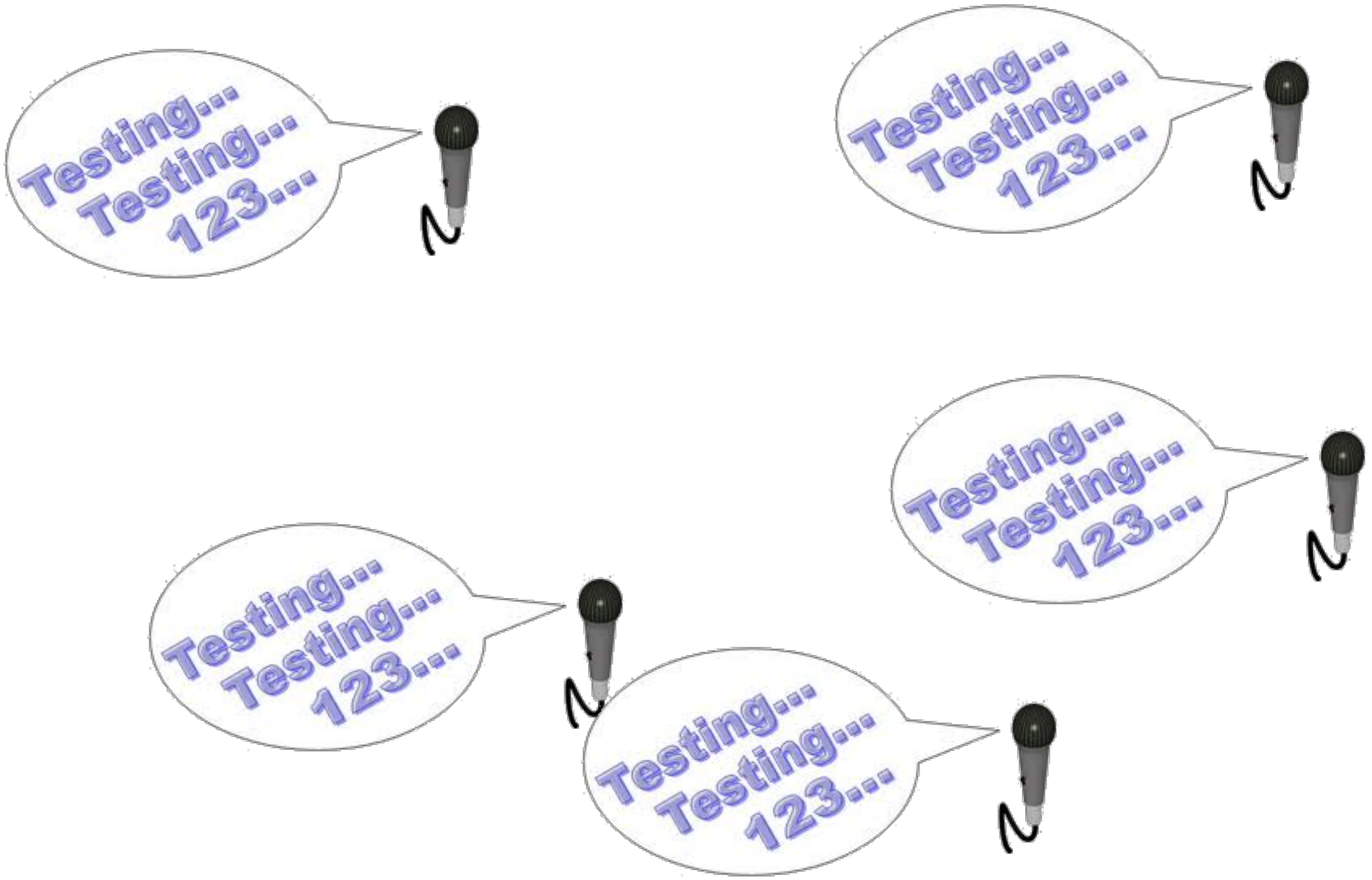
Current Practice: Testing

Current Practice: Testing

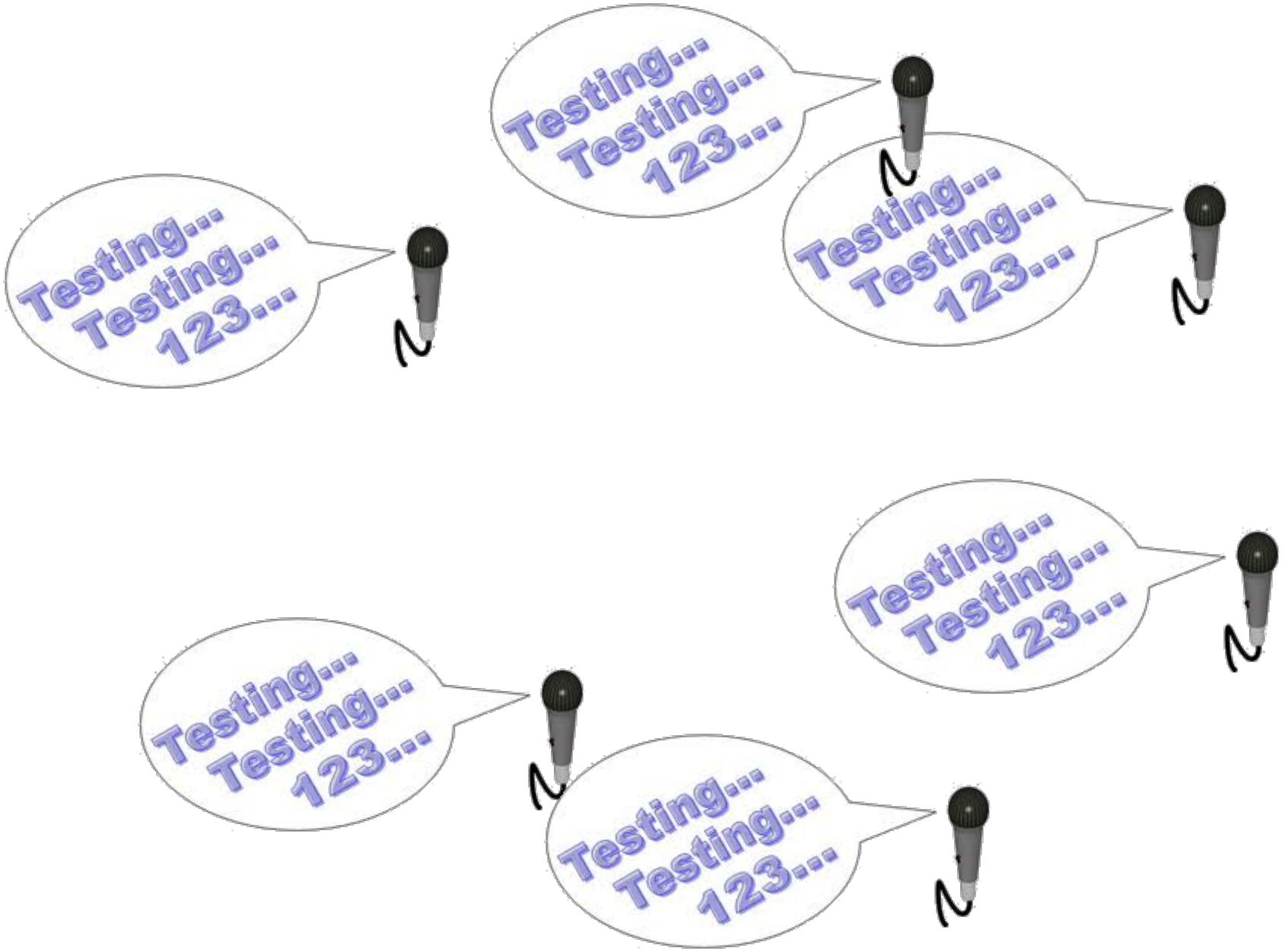
Current Practice: Testing



Current Practice: Testing



Current Practice: Testing



Current Practice: Testing

Current Practice: Testing

Current Practice: Testing

Current Practice: Testing

Current Practice: Testing

Current Practice: Testing

Current Practice: Testing

Current Practice: Testing

Current Practice: Testing

Current Practice: Testing

Testing Pros and Cons

+ Relatively easy to write tests
+ Applies to modules, subsystems, systems
+ Reveals many bugs in practice
+ Relatively easy to adapt as code changes

Testing Pros and Cons

+ Relatively easy to write tests
+ Applies to modules, subsystems, systems
+ Reveals many bugs in practice
+ Relatively easy to adapt as code changes

- Cannot prove absence of bugs

Q. Is there any way to prove the absence of bugs in code?

Q. Is there any way to prove the absence of bugs in code?

A. Yes, with a theorem prover!

Theorem-Proving Tools

- Have been around since the late 1970s
- Increased adoption in academia, industry
  - Quality of tools has improved
  - Need for tools has skyrocketed
- Applied for verification
  - seL4 mobile phone microkernel [Isabelle]
  - CompCert optimizing C compiler [Coq]
  - Quark web browser [Coq]
- Also for mathematics
  - Four-Color Theorem [Coq]
  - Feit-Thompson Theorem [Coq]
  - Kepler Conjecture [Isabelle]

The Agda Theorem Prover

- From Chalmers U. of Technology, Sweden
- Supports verified pure functional programming
  - Inductive datatypes (immutable)
  - Anonymous functions, higher-order functions
  - Pattern-matching, recursion
- One language for programs, proofs
  - Proofs correspond to terminating functional programs
  - Curry-Howard isomorphism
  - Mathematical induction = terminating recursion
  - Minimalistic language, very elegant
  - Supports only *constructive* logic
- Unicode support, user-defined mixfix notation

demo

The Future of Programming?

- Revolutionize open-source
  - Anyone can be a committer
  - (Just require proofs on commit)
- Quantum leap in software quality
  - Prove critical properties
  - Issue: still need to prove the right properties
- Software engineering gets *harder*
  - How to use development budget?
  - More complicated tradeoffs
- Happier programmers
  - Proving software correct is awesome experience
  - Watching it actually run correctly even better!

Conclusion

- Software is increasingly critical for society
- We programmers have an important role
- Great code is a key ingredient
- Testing only goes so far
- Theorem provers: to infinity and beyond!

Slides written in `slideshow` (www.racket-lang.org)

Thank you!

The University of Iowa Computing Conference (UICC)

- Run by U. Iowa ACM student chapter
- Intended for computing students in the region
- Outside speakers
- Programming/puzzle contest
- Will be Feb. 28/March 1 for 2014

http://acm.cs.uiowa.edu/uicc/schedule.html

CS Graduate Program at U. Iowa

- Master of Computer Science (MCS)
  - Improve credentials, knowledge, for industry
  - Typically 2 years, 10 grad classes
  - New for 2014: Regional Scholars Fellowship
    - Teaching assistantship for one academic year
    - Tuition plus stipend (approx $2000/month)
    - Renewable for second year with GPA of 3.25
- Doctor of Philosophy (PhD)
  - Learn to do original research
  - Develop advanced expertise
  - Preparation for career in academia, industrial research
  - Typically 5-6 years
  - Supported through teaching assistantships, fellowships