# Toward software engineering in practice

**Michael Hilton**

School of
Computer Science

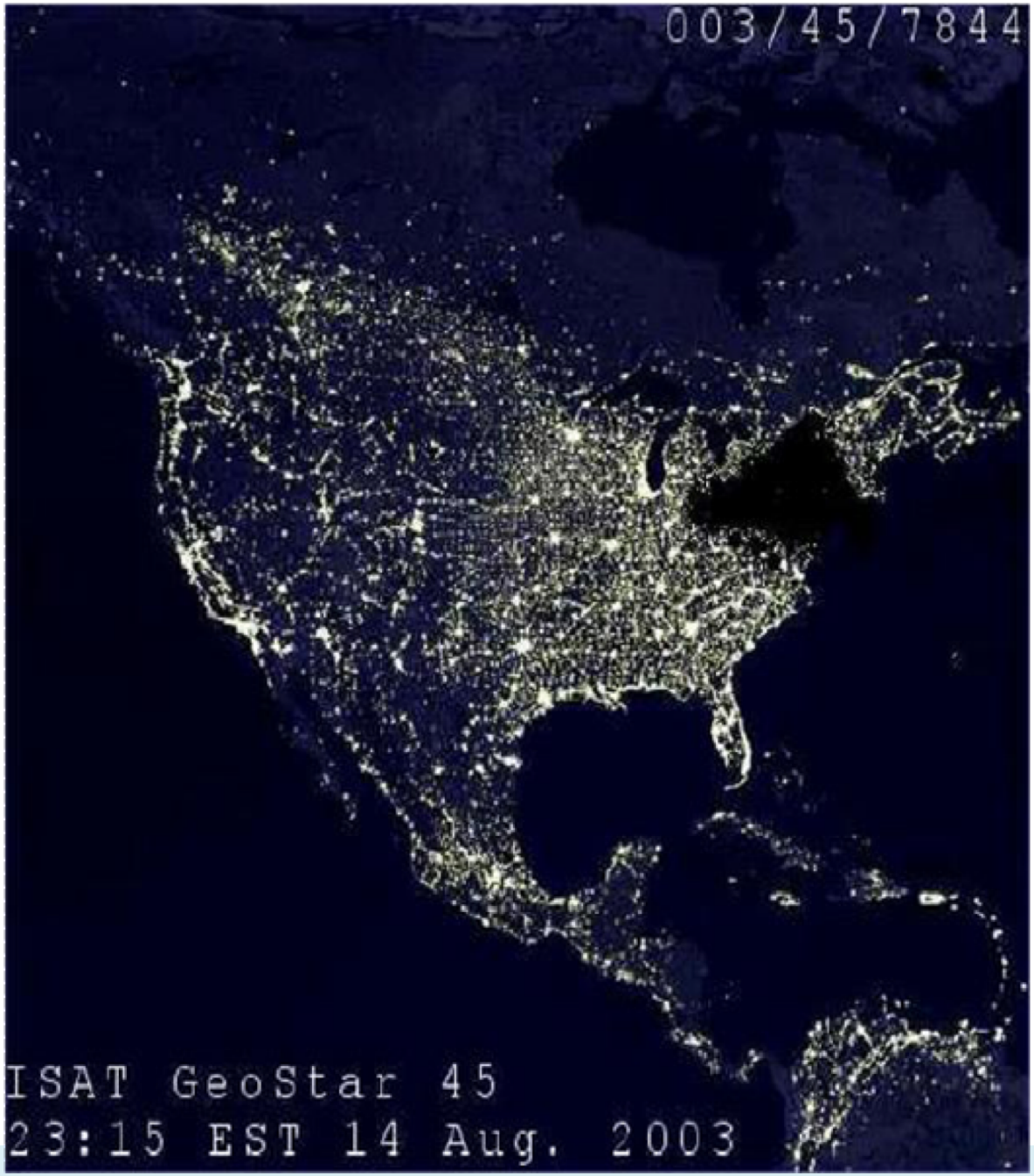
institute for
SOFTWARE
RESEARCH

# Learning Goals

- Introduction to Software Engineering
- Discussion of Test Driven Development

# Introduction

# SOFTWARE IS EVERYWHERE
# SOFTWARE IS IMPORTANT


003/45/7844
ISAT GeoStar 45
23:15 EST 14 Aug. 2003

# Toyota Case: Single Bit Flip That Killed

Junko Yoshida

10/25/2013 03:35 PM EDT

During the trial, embedded systems experts who reviewed Toyota's electronic throttle source code testified that they found Toyota's source code defective, and that it contains bugs -- including bugs that can cause unintended acceleration.

"We did a few things that NASA apparently did not have time to do," Barr said. For one thing, by looking within the real-time operating system, the experts identified "unprotected critical variables." They obtained and reviewed the source code for the "sub-CPU," and they "uncovered gaps and defects in the throttle fail safes."

The experts demonstrated that "the defects we found were linked to unintended acceleration through vehicle testing," Barr said. "We also obtained and reviewed the source code for the black box and found that it can record false information about the driver's actions in the final seconds before a crash."

Stack overflow and software bugs led to memory corruption, he said. And it turns out that the crux of the issue was these memory corruptions, which acted "like ricocheting bullets."

Barr also said more than half the dozens of tasks' deaths studied by the experts in their experiments "were not detected by any fail safe."



## Bookout Trial Reporting

http://www.eetimes.com/document.asp?doc_id=1319903&page_number=1 (excerpts)

"Task X death in combination with other task deaths"

14

FORMULA 1 / AUSTRALIAN GP / BREAKING NEWS

PREV / NEXT

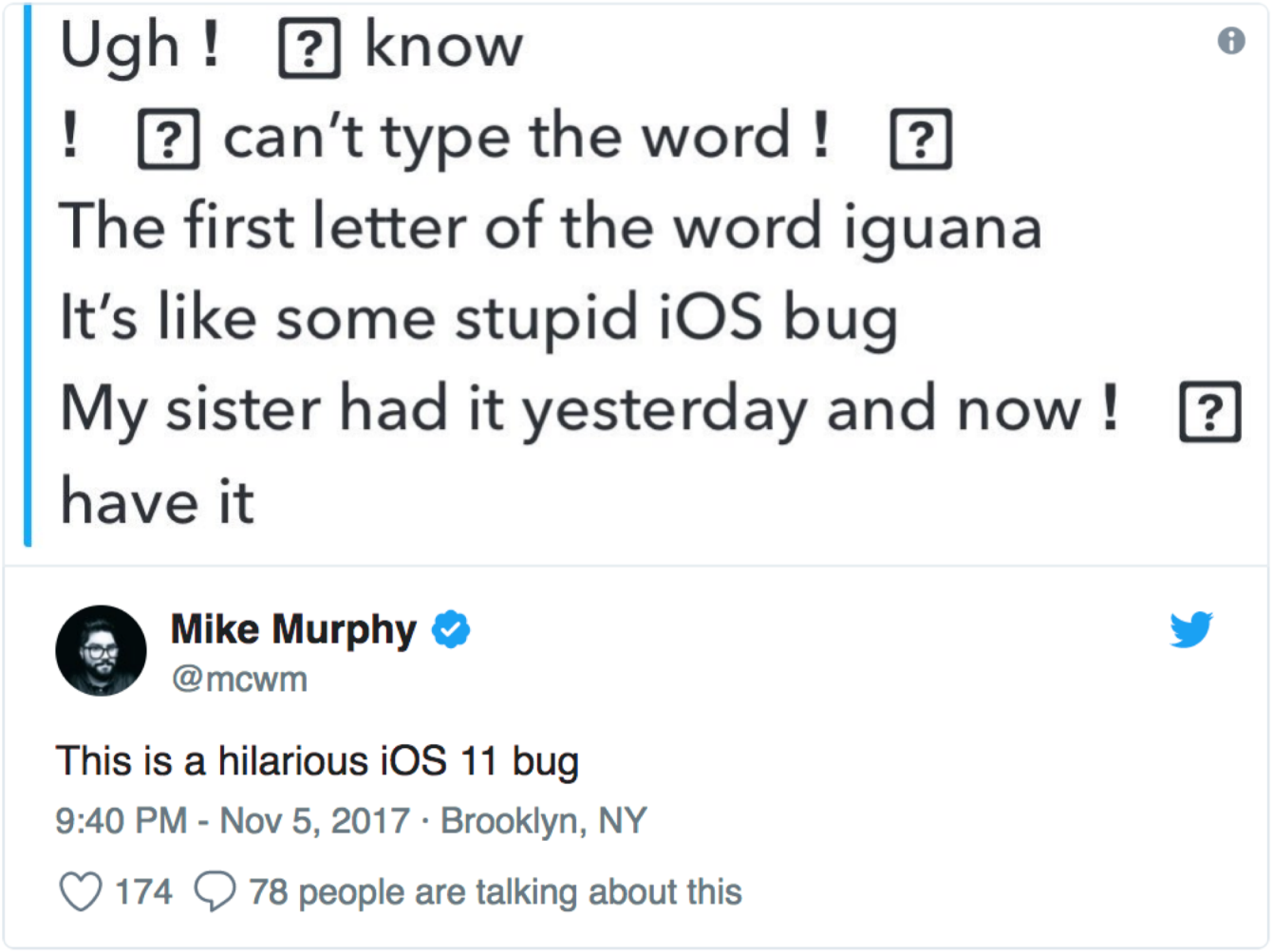
# Mercedes uncovers bug that led to Melbourne defeat

Share on Facebook Share on Twitter

59,698 VIEWS


Mercedes has discovered there was a computer bug in the software it uses at Formula 1 races, after concluding its investigation into what went wrong at the Australian Grand Prix.

# iOS 11 bug


Ugh ! [?] know
! [?] can't type the word ! [?]
The first letter of the word iguana
It's like some stupid iOS bug
My sister had it yesterday and now ! [?]
have it
Mike Murphy
@mcwm
This is a hilarious iOS 11 bug
9:40 PM - Nov 5, 2017 · Brooklyn, NY
174
78 people are talking about this

12.18.17 | PLATFORM WARS

# The Year That Software Bugs Ate The World

In 2017, bugs banned people from Twitter, secretly recorded them in their homes, and even caused a train crash. Is there anything they *can't* do?

[Photo and illustration: AVIcons/iStock; Flickr user Tinh tế Photo]

## Ottawa County Central Dispatch works through **software bugs**

HollandSentinel.com - Apr 13, 2018

A joint meeting between the dispatch authority's policy board and technical advisory committee was held Thursday morning and some discussion touched on the **software** updates. A couple of months after implementing **software** updates, Ottawa County Central Dispatch officials say they are close to ...

## Ottawa County, Mich., Irons Out **Bugs** in Dispatch System

Government Technology - 19 hours ago

(TNS) — A couple of months after implementing **software** updates, Michigan's Ottawa County Central Dispatch officials say they are close to working out all of the **bugs**. A joint meeting between the authority's policy board and technical advisory committee was Thursday, April 12, and some discussion ...


## **Software bugs** leave Cisco devices vulnerable to exploitation

RCR Wireless News - Apr 5, 2018

Cisco discovered another **bug** in its IOS XE **software**, called CVE-2018-0150. According to the company, an attacker could exploit the vulnerability by using this account to remotely connect to an affected device. It said the glitch impacts devices running a vulnerable release of its IOS XE **Software** Release ...

## Pagani recalls every Huayra BC hypercar for battery **software bugs**

CNET - Apr 2, 2018

The issue stems from the battery's management **software**. A **bug** within the **software** may cause the battery to not provide enough power to the vehicle, which presents a safety hazard. Pagani does not know of any injuries or collisions related to the defect. The **bug** was found during standard vehicle testing, ...

What is engineering? And how is it different from hacking/programming?

# SOFTWARE *ENGINEERING*?

# 1968 NATO Conference on Software Engineering

- Provocative Title
- Call for Action
- “Software crisis”

# “Software Engineering”

# Envy of Engineers

- Producing a car/bridge
  - Estimable costs and risks
  - Expected results
  - High quality
- Separation between plan and production
- Simulation before construction
- Quality assurance through measurement
- Potential for automation

# Software Engineering?

*"The Establishment and use of sound **engineering principles** in order to obtain **economically** software that is **reliable** and works **efficiently** on **real** machines."*

*[Bauer 1975, S. 524]*

# Healthcare.gov: Government IT Project Failure at its Finest

Posted: 10/18/2013 6:33 pm

Read more > Project Management, Government, Healthcare, It Projects, Open Source, Business News



The *BusinessWeek* article on the Healthcare.gov failure is nothing if not instructive. From the piece:

> Healthcare.gov isn't just a website; it's more like a platform for building health-care marketplaces. Visiting the site is like visiting a restaurant. You sit in the dining room, read the menu, and tell the waiter what you want, and off he goes to the kitchen with your order. The dining room is the front end, with all the buttons to click and forms to fill out. The kitchen is the back end, with all the databases and services. The contractor most responsible for the back end is CGI Federal. Apparently it's this company's part of the system that's burning up under the load of thousands of simultaneous users.

The restaurant analogy is a good one. Projects with scopes like these fail for all sorts of reasons. *Why New Systems Fail* details a bunch of culprits, most of which are people-related.

As I read the article, a few other things jumped out at me, as they virtually guarantee failure:

- The sheer number of vendors involved
- The unwillingness of key parties involved with the back-end to embrace transparency

# What happened with HealthCare.gov?

- Poor team and process coordination.
- Changing requirements.
- Inadequate quality assurance infrastructure.
- Architecture unsuited to the ultimate system load.

# PROCESS

# How to develop software?

1. Discuss the software that needs to be written
2. Write some code
3. Test the code to identify the defects
4. Debug to find causes of defects
5. Fix the defects
6. If not done, return to step 1

# Example process issues

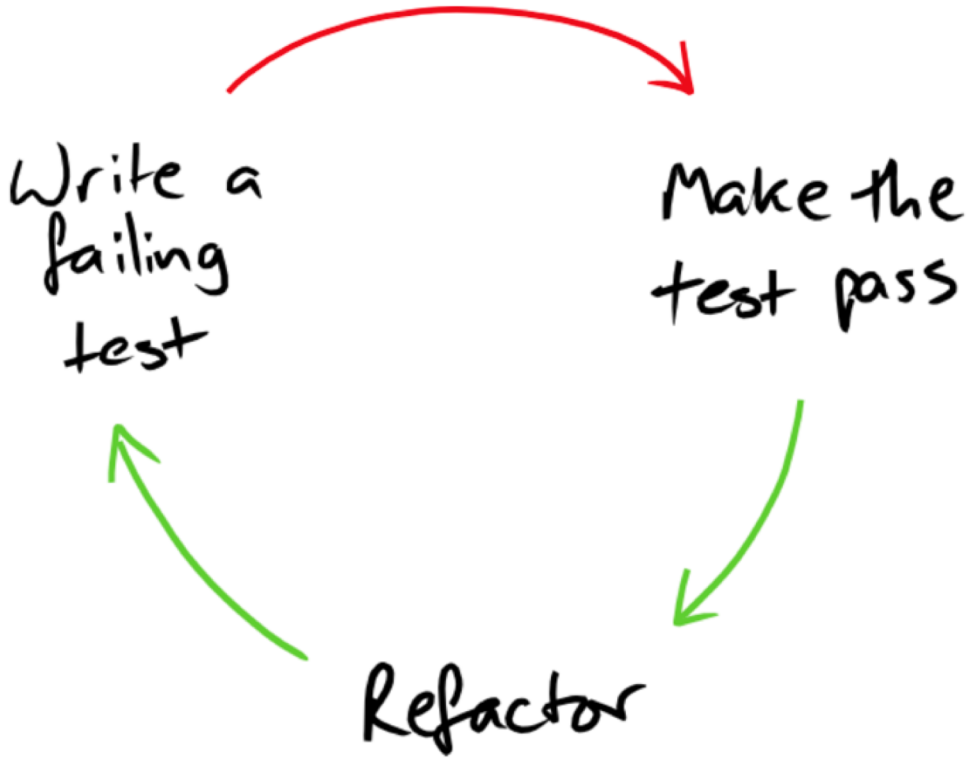
- **Change Control:** Mid-project informal agreement to changes suggested by customer or manager. Project scope expands 25-50%
- **Quality Assurance:** Late detection of requirements and design issues. Test-debug-reimplement cycle limits development of new features. Release with known defects.
- **Defect Tracking:** Bug reports collected informally, forgotten
- **System Integration:** Integration of independently developed components at the very end of the project. Interfaces out of sync.
- **Source Code Control:** Accidentally overwritten changes, lost work.
- **Scheduling:** When project is behind, developers are asked weekly for new estimates.

# TEST DRIVEN DEVELOPMENT (TDD)

# Three simple rules

1. You are not allowed to write any production code unless it is to make a **failing unit test pass**.
2. You are not allowed to write any more of a unit test than is **sufficient to fail**; and compilation failures are failures.
3. You are not allowed to write any more production code than is **sufficient to pass** the one failing unit test.

# TDD Cycle




From Growing Object-Oriented Software by Nat Pryce and Steve Freeman
http://www.growing-object-oriented-software.com/figures.html

@sebrose

http://cucumber.io

# Why TDD?

"The act of writing a unit test is more an act of **design** than of verification.

It is also more an act of **documentation** than of verification.

The act of writing a unit test closes a remarkable number of feedback loops, the least of which is the one pertaining to **verification** of function".

## Advantages of TDD

- Clear place to start
- Much less code thrown away, less wasted effort
- Less Fear
- Side Effect: Robust test suite

A programming exercise that you repeat many many times, looking to make small, incremental improvements.

# CODE KATA

# Diamond Kata

- Given a letter, print a diamond starting with ‘A’ with the supplied letter at the widest point.

- For example: ‘C’ prints

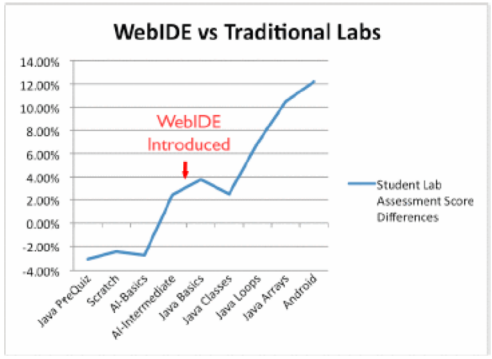
```
  A
 B B
C   C
 B B
  A
```


**TDD Demo**

# IMPRESSIONS?

# TDD Research

- Hilton et al.: Students learn better when forced to write tests first
- Bhat et al.: At Microsoft, projects using TDD had greater than two times code quality, but 15% more upfront setup time




- George et al.: TDD passed 18% more test cases, but took 16% more time
- Scanniello et al.: Perceptions of TDD include: novices believe TDD improves productivity at the expense of internal quality

# More TDD Research

- Fucci et al.:  Results: The Kruskal-Wallis tests did not show any significant difference between TDD and TLD in terms of testing effort (p-value = .27), external code quality (p-value = .82), and developers' productivity (p-value = .83).

- Fucci et al.: Conclusion: The claimed benefits of TDD may not be due to its distinctive test-first dynamic, but rather due to the fact that TDD-like processes encourage fine-grained, steady steps that improve focus and flow.

# WHY IS THIS HARD?

# Summary: take 17-313 this fall!

- Software Engineering in practice requires consideration of numerous issues---technical and social---above the level of individual class design/implementation.
- Do you think this is interesting? 17-313, Foundations of Software Engineering is offered in the Fall.
- And consider the undergraduate SE minor!