Foundations of Software Engineering

Part 1: Overview

Michael Hilton, Christian Kästner

My Story

- B.S.
- Soft
- M.S.
- PhD
- Inte 2017
- Assi 2017

Learning Goals

- Broad scope of software engineering
- Importance of nontechnical issues
- Overview key challenges

- Syllabus, introduction and team forming

"…participants who multitasked on a laptop during a lecture scored lower on a test compared to those who did not multitask, and participants who were in direct view of a multitasking peer scored lower on a test compared to those who were not. The results demonstrate that *multitasking on a laptop poses a significant distraction to both users and* ***fellow students*** *and can be detrimental to comprehension of lecture content*."

Faria Sana, Tina Weston, and Nicholas J. Cepeda. 2013. Laptop multitasking hinders classroom learning for both users and nearby peers. Comput. Educ. 62 (March 2013), 24-31.

isr institute for SOFTWARE RESEARCH

“…students who took notes on **laptops performed worse** on conceptual questions than students who took notes longhand. We show that whereas taking more notes can be beneficial, laptop note takers’ tendency to transcribe lectures verbatim rather than processing information and reframing it in their own words is detrimental to learning.”

Psychol Sci. 2014 Jun;25(6):Epub 2014 Apr 23.
The pen is mightier than the keyboard: advantages of longhand over laptop note taking. Mueller PA1, Oppenheimer DM2.

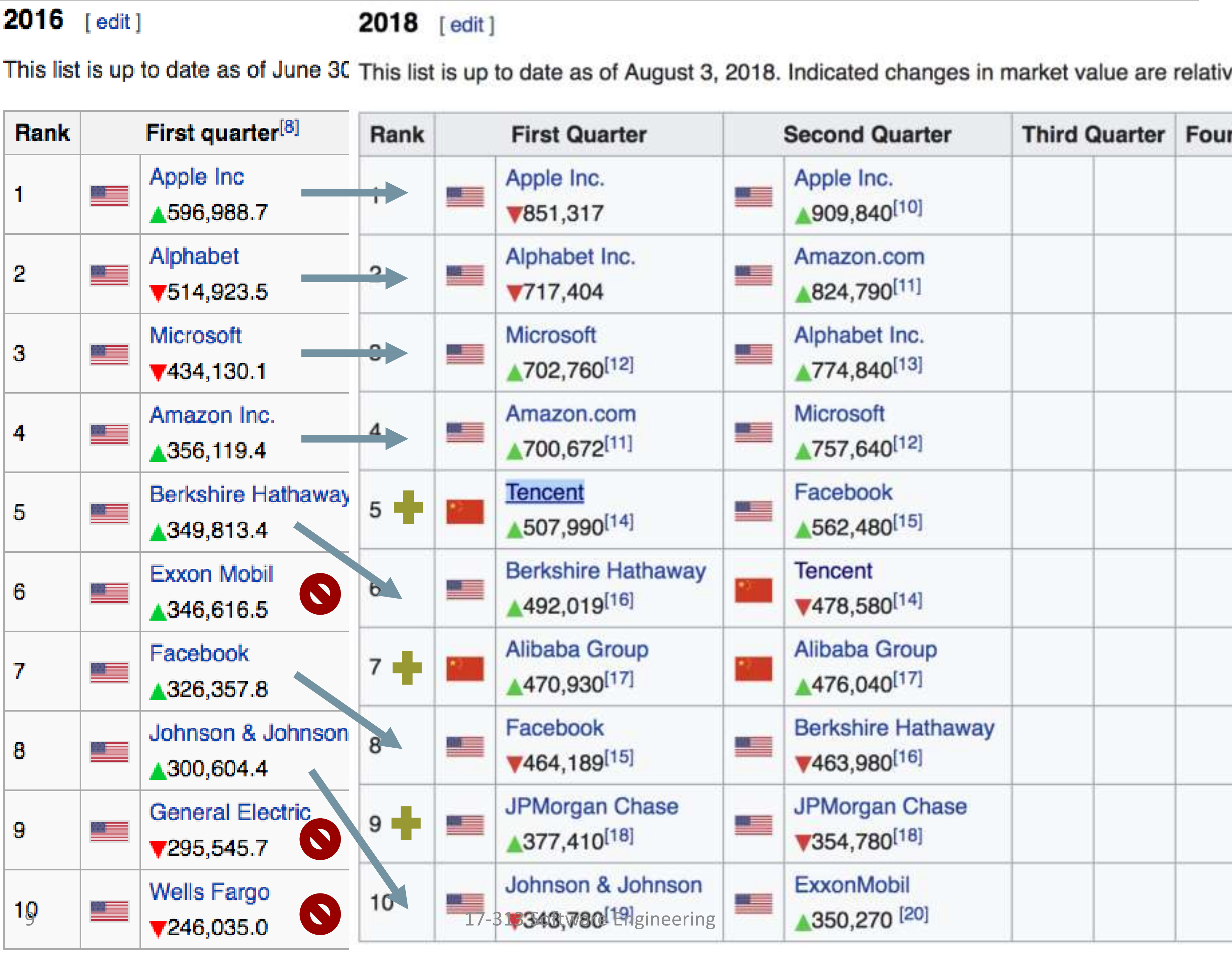
Smoking Section

Software is Everywhere
Software is Important
(duh)

2016 [edit]

This list is up to date as of June 30, 2016. Indicated changes in market value are relative to the previous quarter.

Rank	First quarter[8]		Second quarter		Third quarter		Fourth quarter	
1	🇺🇸	Apple Inc ▲596,988.7	🇺🇸	Apple Inc ▼515,590.0				
2	🇺🇸	Alphabet ▼514,923.5	🇺🇸	Alphabet ▼475,160.0				
3	🇺🇸	Microsoft ▼434,130.1	🇺🇸	Microsoft ▼399,710.0				
4	🇺🇸	Amazon Inc. ▲356,119.4	🇺🇸	Exxon Mobil ▲388,710.0				
5	🇺🇸	Berkshire Hathaway ▲349,813.4	🇺🇸	Berkshire Hathaway ▲356,810.0				
6	🇺🇸	Exxon Mobil ▲346,616.5	🇺🇸	Amazon Inc. ▼337,650.0				
7	🇺🇸	Facebook ▲326,357.8	🇺🇸	Johnson & Johnson ▲333,650.0				
8	🇺🇸	Johnson & Johnson ▲300,604.4	🇺🇸	Facebook ▲326,880.0				
9	🇺🇸	General Electric ▼295,545.7	🇺🇸	General Electric ▼289,480.0				
10	🇺🇸	Wells Fargo ▼246,035.0	🇺🇸	Wells Fargo ▼238,950.0				



2016 [edit]

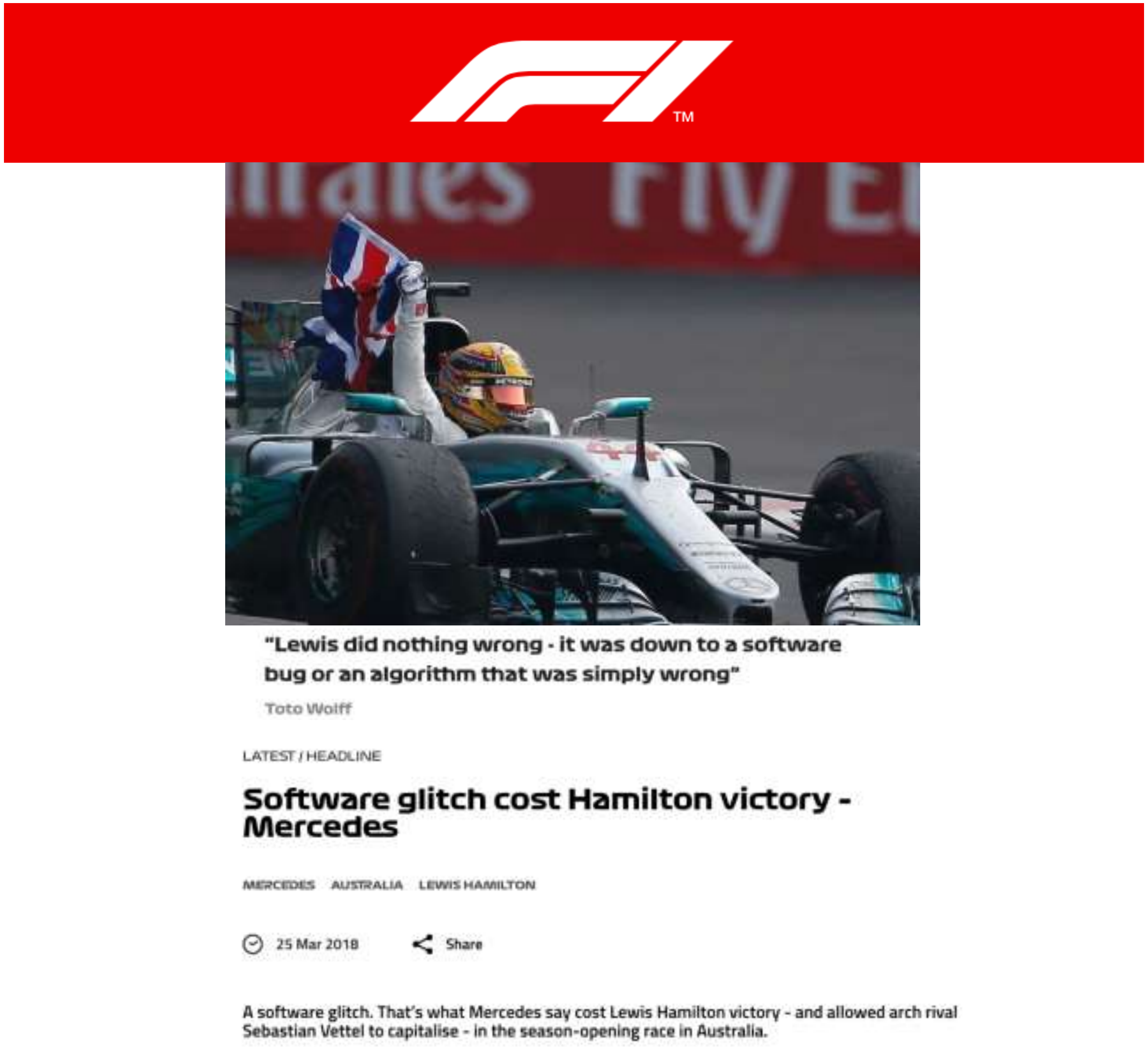
This list is up to date as of June 3C

Rank	First quarter[8]	
1	🇺🇸	Apple Inc ▲596,988.7
2	🇺🇸	Alphabet ▼514,923.5
3	🇺🇸	Microsoft ▼434,130.1
4	🇺🇸	Amazon Inc. ▲356,119.4
5	🇺🇸	Berkshire Hathaway ▲349,813.4
6	🇺🇸	Exxon Mobil ▲346,616.5 🚫
7	🇺🇸	Facebook ▲326,357.8
8	🇺🇸	Johnson & Johnson ▲300,604.4
9	🇺🇸	General Electric ▼295,545.7 🚫
10	🇺🇸	Wells Fargo ▼246,035.0 🚫

2018 [edit]

This list is up to date as of August 3, 2018. Indicated changes in market value are relativ

Rank	First Quarter		Second Quarter		Third Quarter		Four
1	🇺🇸	Apple Inc. ▼851,317	🇺🇸	Apple Inc. ▲909,840[10]			
2	🇺🇸	Alphabet Inc. ▼717,404	🇺🇸	Amazon.com ▲824,790[11]			
3	🇺🇸	Microsoft ▲702,760[12]	🇺🇸	Alphabet Inc. ▲774,840[13]			
4	🇺🇸	Amazon.com ▲700,672[11]	🇺🇸	Microsoft ▲757,640[12]			
5 ✚	🇨🇳	Tencent ▲507,990[14]	🇺🇸	Facebook ▲562,480[15]			
6	🇺🇸	Berkshire Hathaway ▲492,019[16]	🇨🇳	Tencent ▼478,580[14]			
7 ✚	🇨🇳	Alibaba Group ▲470,930[17]	🇨🇳	Alibaba Group ▲476,040[17]			
8	🇺🇸	Facebook ▼464,189[15]	🇺🇸	Berkshire Hathaway ▼463,980[16]			
9 ✚	🇺🇸	JPMorgan Chase ▲377,410[18]	🇺🇸	JPMorgan Chase ▼354,780[18]			
10	🇺🇸	Johnson & Johnson ▼343,780[19]	🇺🇸	ExxonMobil ▲350,270 [20]			



"Lewis did nothing wrong - it was down to a software bug or an algorithm that was simply wrong"

Toto Wolff

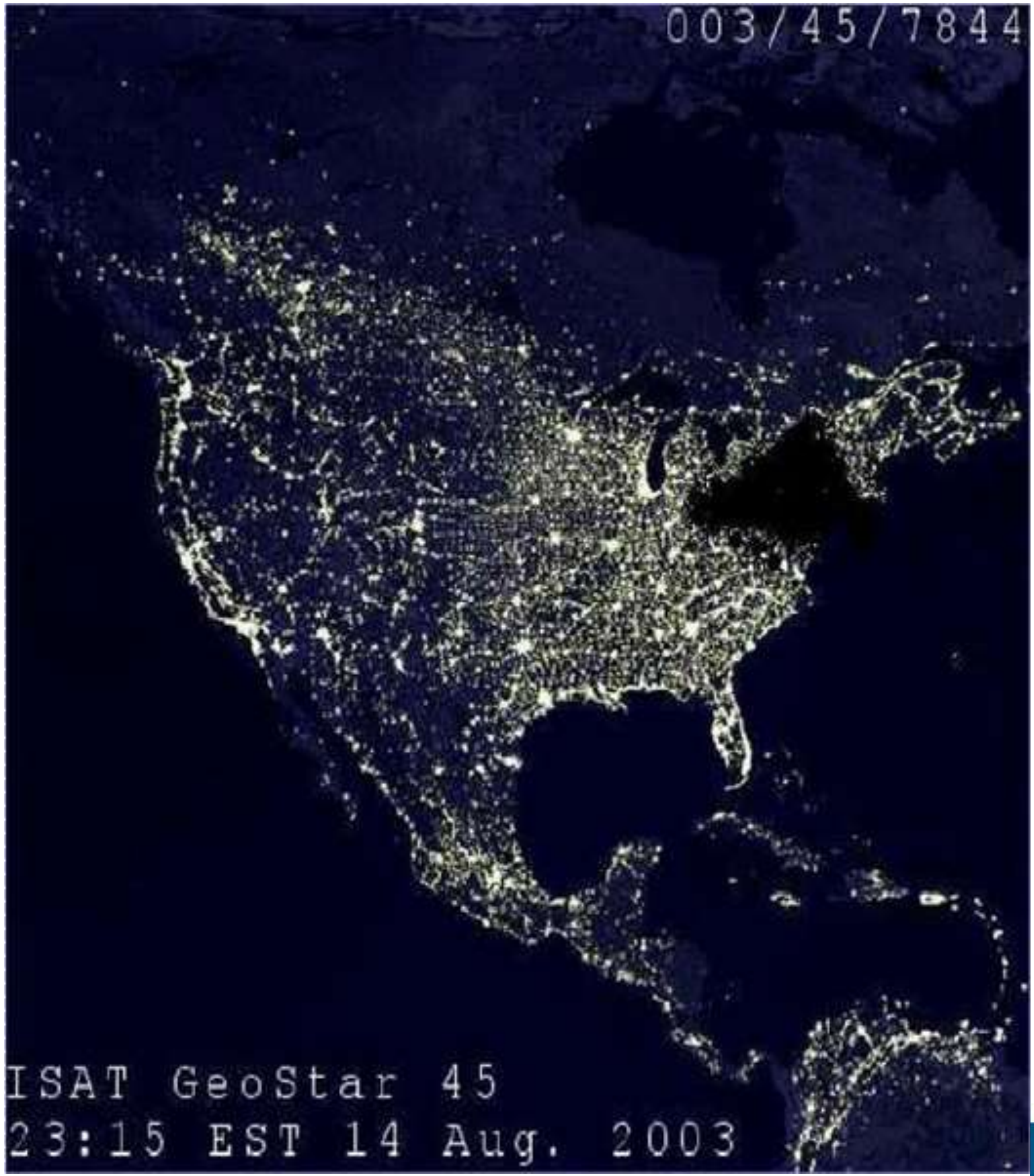
LATEST / HEADLINE

Software glitch cost Hamilton victory - Mercedes

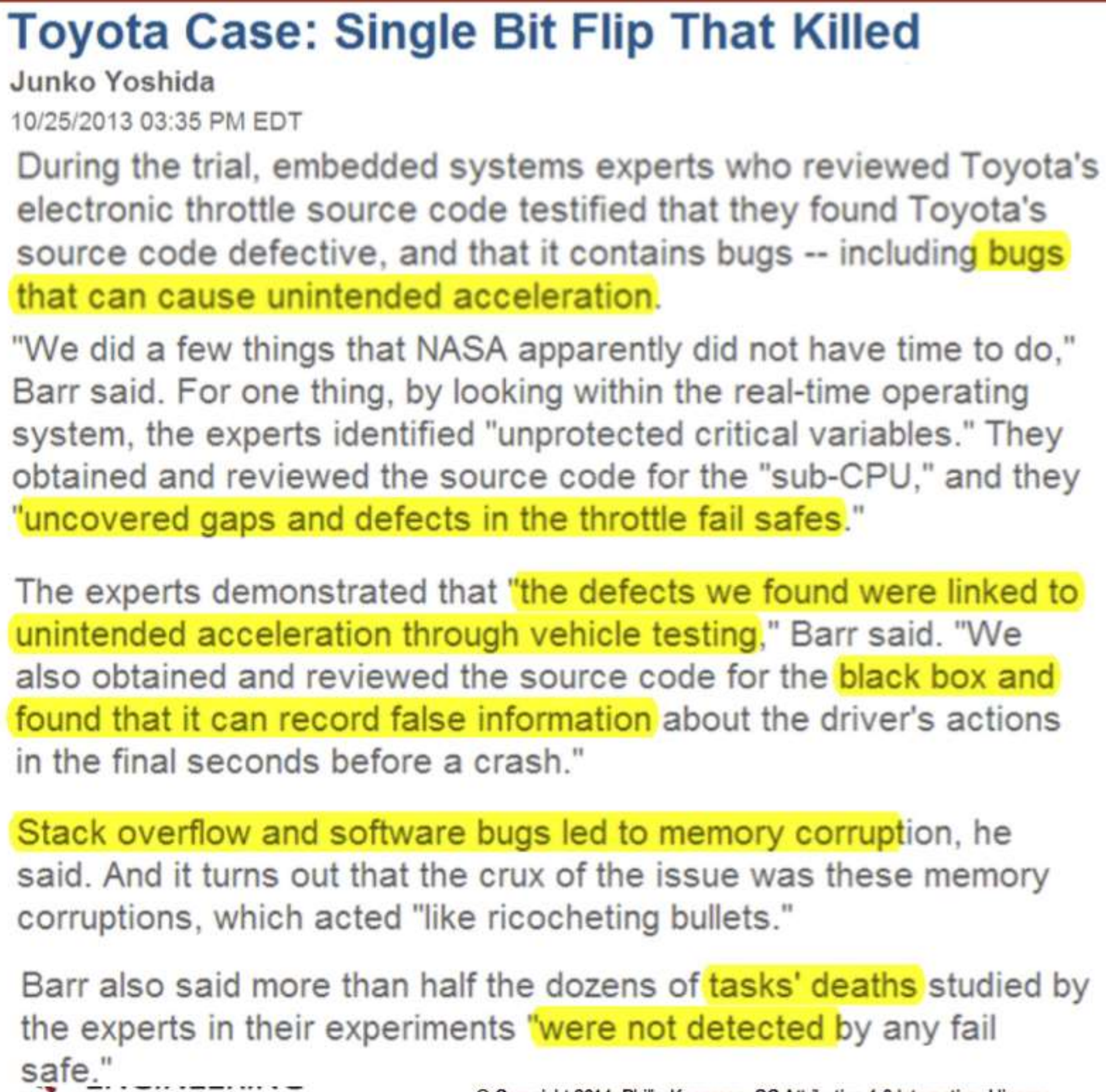
MERCEDES AUSTRALIA LEWIS HAMILTON

25 Mar 2018 Share

A software glitch. That's what Mercedes say cost Lewis Hamilton victory - and allowed arch rival Sebastian Vettel to capitalise - in the season-opening race in Australia.


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

Bookout Trial Reporting



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."

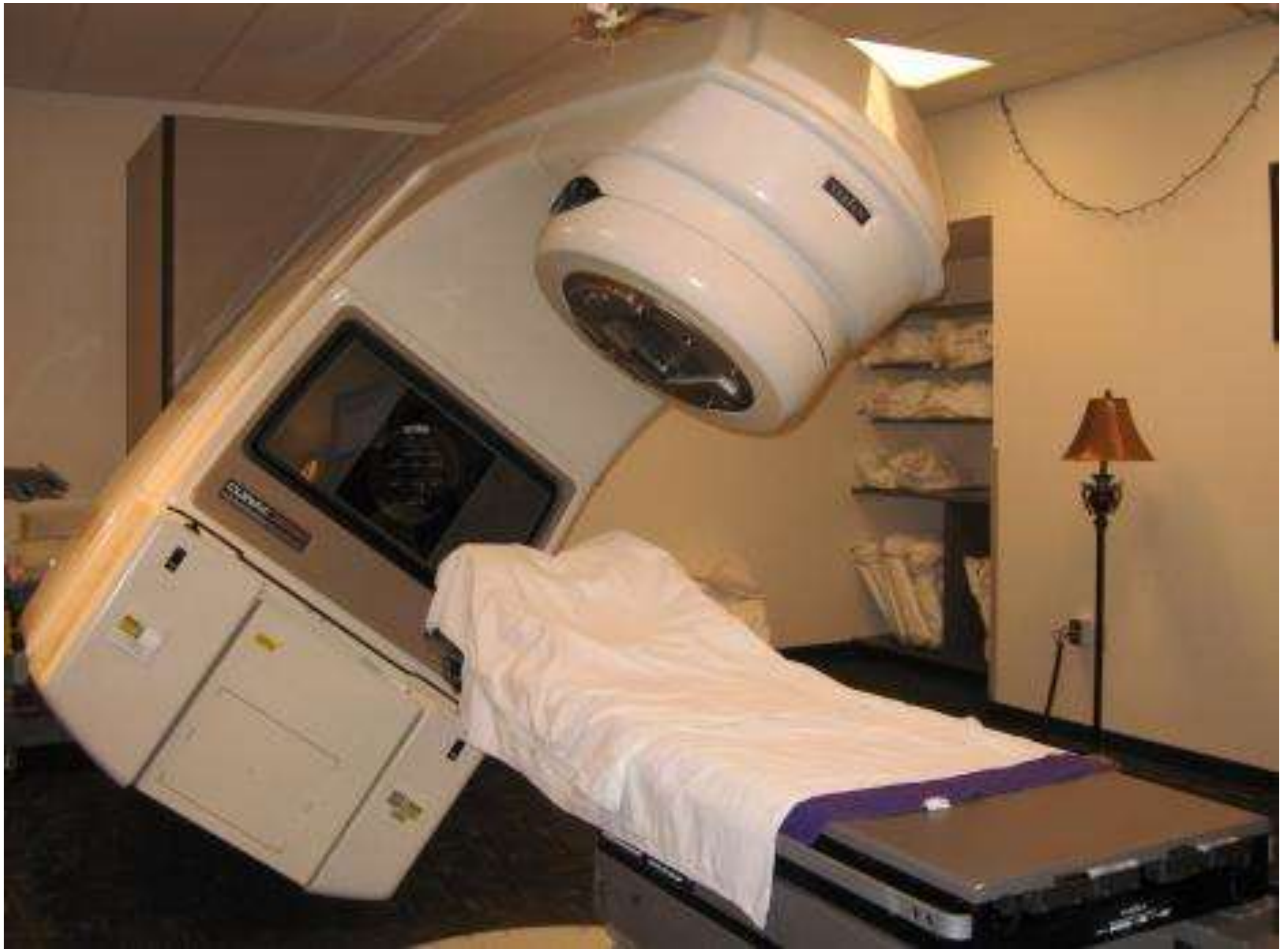
Barr also said more than half the dozens of tasks' deaths studied by the experts in their experiments "were not detected by any fail safe."

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

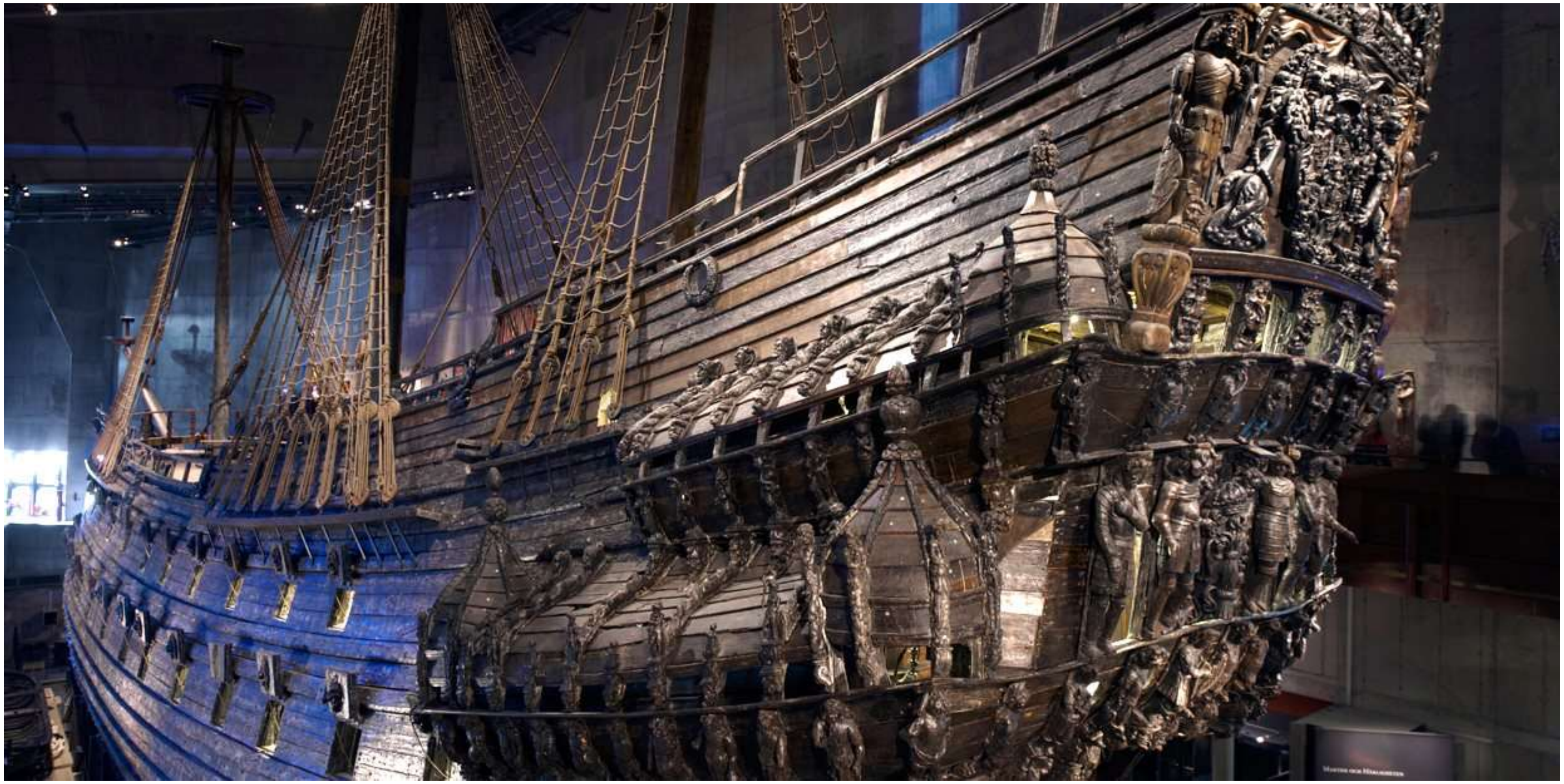
"Task X death in combination with other task deaths"

Healthcare.gov: Government IT Project Failure at its Finest

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



Vasa



Vasa

Skeppsgården
1627
1
1626
2
1628
Djurgården
Skeppsholmen
Gamla stan
Kastellholmen
Beck-
holmen
3
500 m

What happened?

- Changing shipbuilding orders
- No specifications for modified keel
- Shifting armaments requirements

Requirements

- Shipwright's death

teams

- No way to calculate stability, stiffness, or sailing characteristics

measurement

- Failed pre-launch stability tests

QA

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”

Margaret Hamilton

- Name
- Interesting software development experience?
- Specific topic of interest?

Syllabus and course mechanics

Course Themes

- Software engineering as a human process
- Process
- Requirements
- Measurement
- Quality, incl. Security
- Time and team management
- Economics
- Strategic thinking about software

Prerequisites

- Assumes working knowledge of popular programming language
- Assumes experience with team-based software development in medium-sized projects (e.g., Scrabble)

- vs 17-214
  - 313 largely focused on human issues and quality beyond functional correctness
  - 313 focused on larger scale

Active Lecture

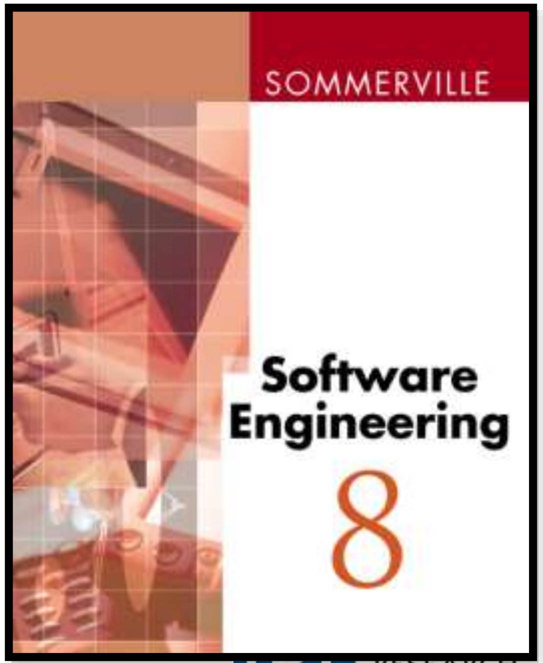
- Case study driven
- Discussion highly encouraged
- Contribute own experience
- Regular active in-class exercises
- In-class presentation
- Discussions over definitions

Readings and Quizzes

- Reading assignments for most lectures
  - Preparing in-class discussions
  - Background material, case descriptions, possibly also podcast, video, wikipedia
  - Complement with own research
- Short and easy online quizzes on readings, due by start of lecture

Textbook

- No single textbook
- Assigned readings from different sources
  - Book chapters (library)
  - News articles
  - Lecture notes
- Recommended supplementary reading: Sommerville, Software Engineering, edition 7 or 8
  - Aim for a used edition for <10$



Gaining Experience

- Case study analyses
- Team assignments
- Open source engagement

- No “survivor”-style projects – wait till 17-413

Evaluation

- Assignments (50 %)
  - Regular homework,
    mostly in teams with individual component
  - Open source engagement
- Midterm (15 %)
- Final (20 %)
- Participation in lecture and recitation (10 %)
- Quizzes on reading assignments (5%)
- Read the learning goals!

Participation

- Participation is important
  - Participation in in-class discussions
  - Active participation in recitations
  - Both quality and quantity are important, quality more than quantity
- Participation != Attendance

Recitations

- Practical tasks, preparation for homework, extra material, discussions
- Please bring laptop, have github account

- This week: Collaborating with Git and other tools

Assignments

- Planning and developing a nontrivial software project as a team
- Develop and execute a test plan
- Solicit requirements
- Implement an own static and dynamic analysis, extending FindBugs
- Contributing to an open source project of your choice

Team Assignments

- Mirror realistic setting
- Assigned teams throughout the semester
  - Fill in team building survey before next lecture
- Peer evaluation and conflict resolution process as needed
- Most team assignments have individual components

Late day policy

- No late days
  - (simply doesn't work with team assignments)
- Accommodations in case of health issues, travel for interviews, … on case by case base
  - Inform us at least 2 days before deadline

Academic Honesty

- 214-like Collaboration Policy
- University Policy on Academic Integrity

+

- In group work, be honest about contribution of group members; do not cover for others

Course Infrastructure

- Course website
  - schedule, slides, syllabus, office hours
- Canvas + Piazza
  - homework, grades, discussions
- Git/Github for coding and collaboration

- Office hours on web page, open door policy
  staff-17313@lists.cmu.edu,

Two Surveys

Survey Goals

- Forming balanced groups
- Shaping the courses based on
  - your background knowledge
  - your interests
- Identifying experience in the room

Reading Assignment Sep 1

- Sommerville Software Engineering, ed 7 or 8 – Chapter “Project Management”
- Complete quick quiz on Canvas before class

Case Study 1: PeopleCars

Case Study 1: PeopleCars

- Scenario and question from prior final
- Read scenario and question
- Discuss answers with your neighbors

- Keep answers until last lecture

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

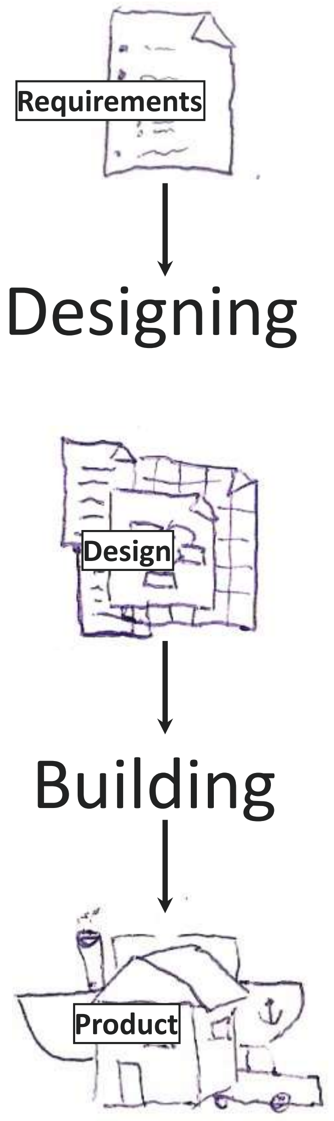
Software Engineering?

„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]

Dangerous Analogy

- Software = Design = Plan
- Programming is design, not production
  - Production (copying/loading a program) is automated
  - Simulation not necessary
- Agile technologies possible
- Quality measurement?


Requirements
Designing
Design
Building
Product