GBL-ITEA
35th International Test & Evaluation Symposium

Michael Soltys
CSU Channel Islands

Wed Dec 13, 2018
Academia Day Panel

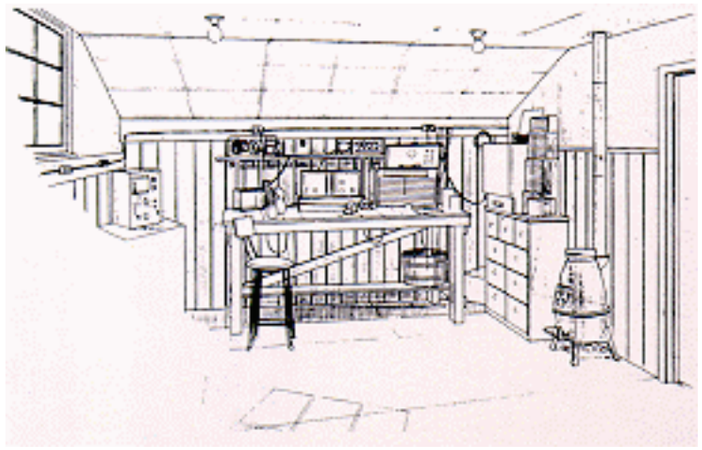
University of Toronto

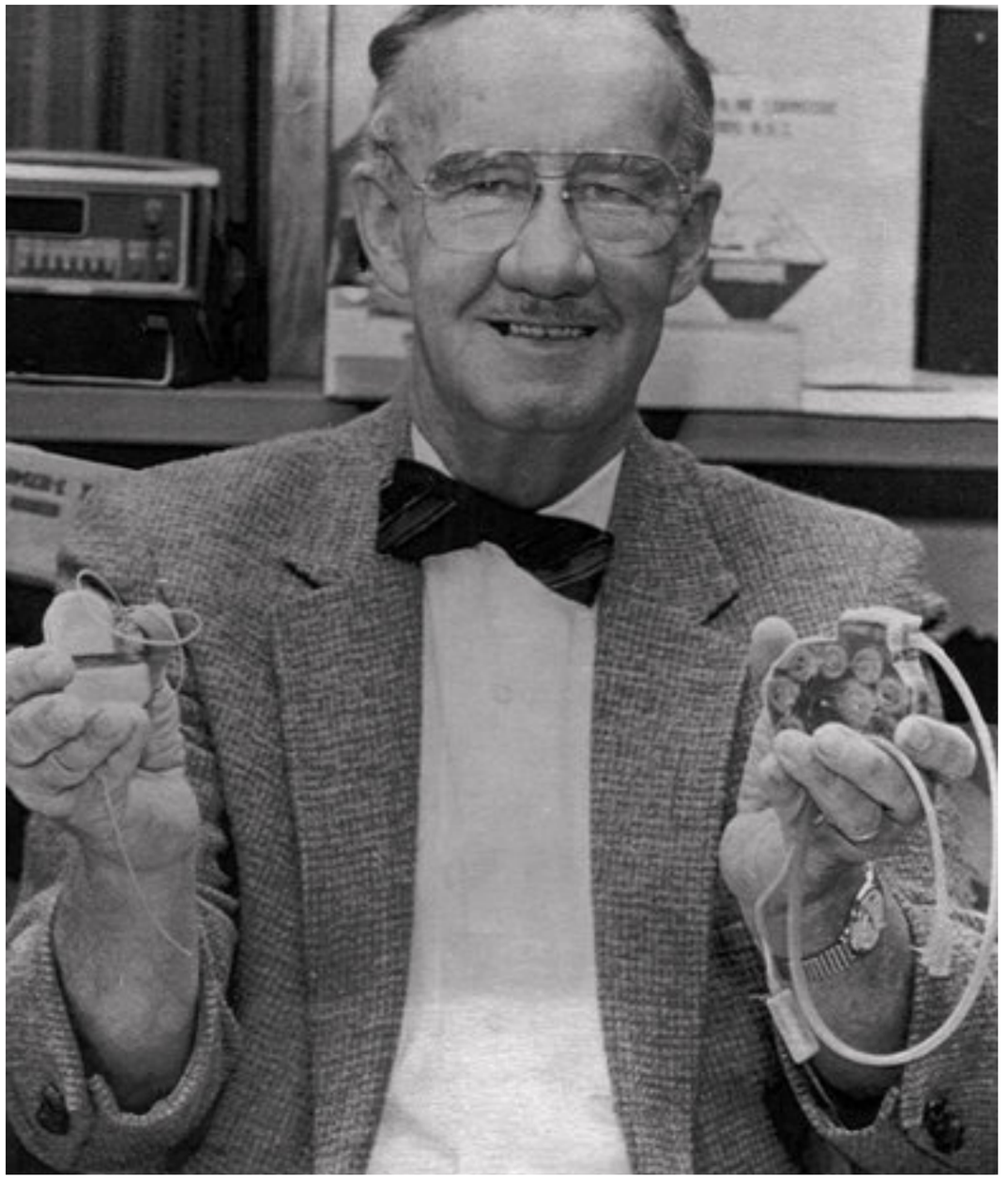
PhD at Toronto 2001

- Stephen Cook, one of the founds of theoretical CS

- NP-completeness result from 1971, for which he got the Turing award

Wilson Greatbatch



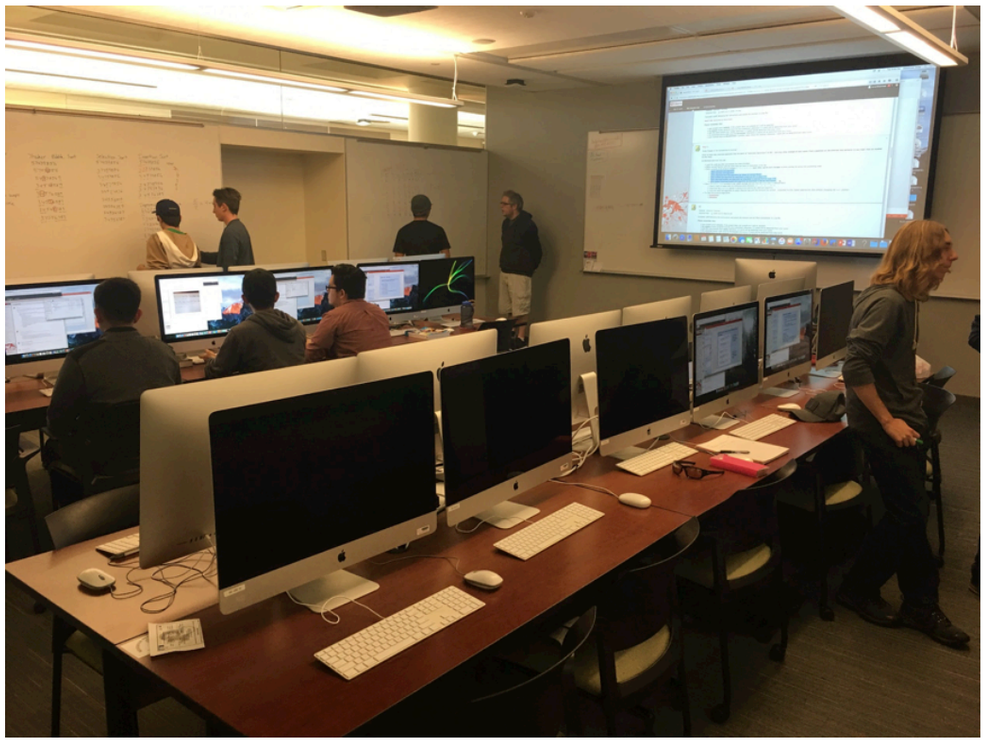


- Inventor of 1st implantable pacemaker
- workshop in barn in Clarence, NY
- 10 year old Stephen Cook a precocious neighbor

CS @ CI

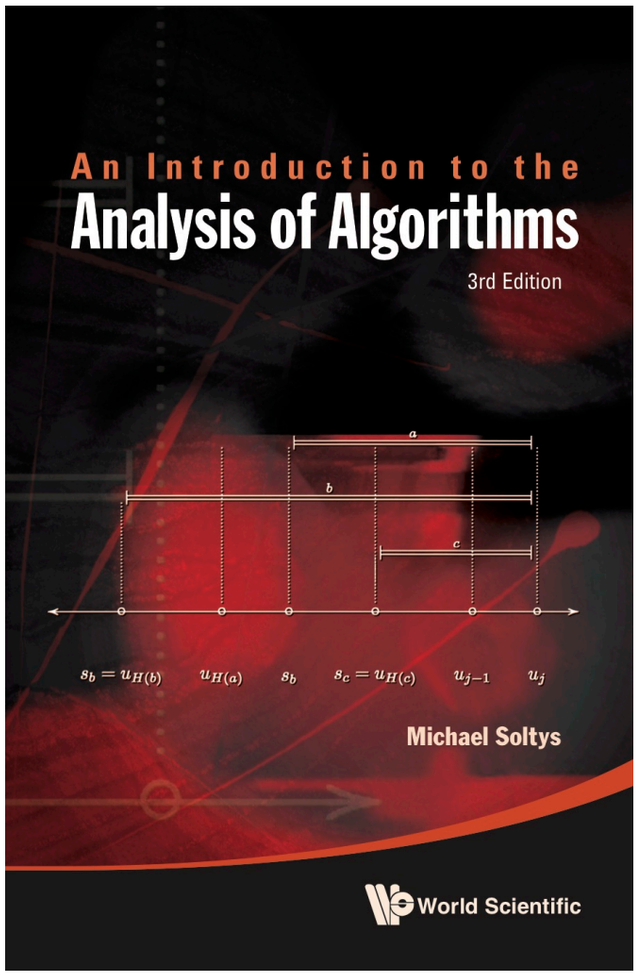
Sierra Hall

Lab



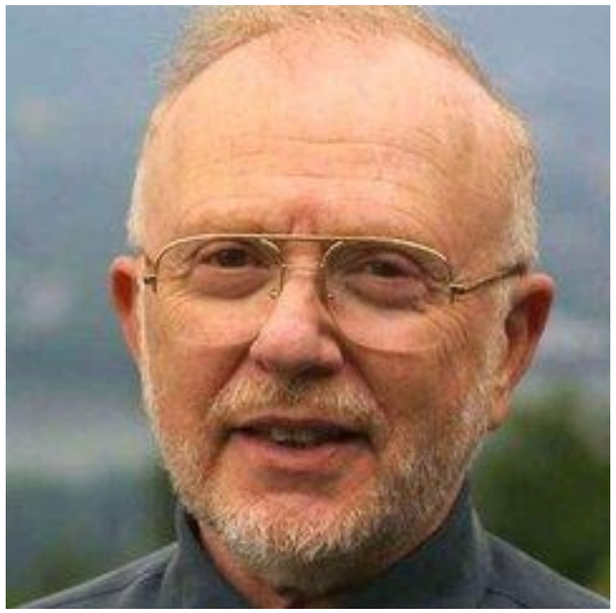
Algorithms

- Written for Software Engineers
- Perspective:
  proofs of correctness
- 3rd edition



Cybersecurity

Why test and evaluate?



Not an “add on”
Intrinsic part of software development

Testing is hard

Two methods in Software Engineering:

- Run on test cases
- Formal methods

Algorithm 9.5 `mult(A,B)`

Pre-condition: $B \geq 0$

$a = A;$
$b = B;$
$y = 0;$
while $b > 0$ **do**
    $y = y + a;$
    $b = b - 1;$
end while

Post-condition: $y = A \cdot B$

1 $\{y + a(b-1) = AB \wedge (b-1) \geq 0\}$`b=b-1;`$\{y + ab = AB \wedge b \geq 0\}$
assignment
2 $\{(y+a)+a(b-1) = AB \wedge (b-1) \geq 0\}$`y=y+a;`$\{y+a(b-1) = AB \wedge (b-1) \geq 0\}$
assignment
3 $(y + ab = AB \wedge b - 1 \geq 0) \rightarrow ((y + a) + a(b-1) = AB \wedge b - 1 \geq 0)$
theorem
4 $\{y + ab = AB \wedge b - 1 \geq 0\}$`y=y+a;`$\{y + a(b-1) = AB \wedge b - 1 \geq 0\}$
consequence left 2 and 3
5 $\{y + ab = AB \wedge b - 1 \geq 0\}$`y=y+a;b=b-1;`$\{y + ab = AB \wedge b \geq 0\}$
composition on 4 and 1
6 $(y + ab = AB) \wedge b \geq 0 \wedge b > 0 \rightarrow (y + ab = AB) \wedge b - 1 \geq 0$
theorem
7 $\{(y + ab = AB) \wedge b \geq 0 \wedge b > 0\}$`y=y+a; b=b-1;`$\{y + ab = AB \wedge b \geq 0\}$
consequence left 5 and 6

8 $\{(y+ab = AB) \wedge b \geq 0\}$
```
while (b>0)
   y=y+a;
   b=b-1;
```
$\{y+ab = AB \wedge b \geq 0 \wedge \neg(b > 0)\}$
while on 7
9 $\{(0 + ab = AB) \wedge b \geq 0\}$ `y=0;` $\{(y + ab = AB) \wedge b \geq 0\}$
assignment

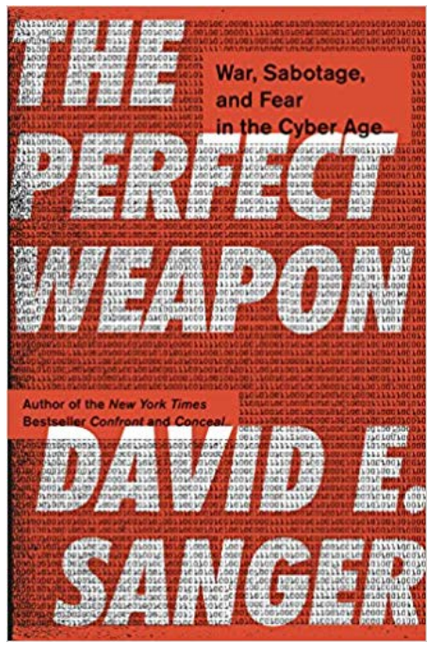
10 $\{(0+ab = AB) \wedge b \geq 0\}$
```
y=0;
while (b>0)
   y=y+a;
   b=b-1;
```
$\{y+ab = AB \wedge b \geq 0 \wedge \neg(b > 0)\}$

composition on 9 and 8
11 $\{(0 + aB = AB) \wedge B \geq 0\}$ `b=B;` $\{(0 + ab = AB) \wedge b \geq 0\}$
assignment

12 $\{(0+aB = AB) \wedge B \geq 0\}$
```
b=B;
y=0;
while (b>0)
   y=y+a;
   b=b-1;
```
$\{y+ab = AB \wedge b \geq 0 \wedge \neg(b > 0)\}$

composition on 11 and 10
13 $\{(0 + AB = AB) \wedge B \geq 0\}$ `a=A;` $\{(0 + aB = AB) \wedge B \geq 0\}$
assignment
14 $\{(0 + AB = AB) \wedge B \geq 0\}$ `mult(A,B)` $\{y + ab = AB \wedge b \geq 0 \wedge \neg(b > 0)\}$
composition on 13 and 12
15 $B \geq 0 \rightarrow ((0 + AB = AB) \wedge B \geq 0)$
theorem
16 $(y + ab = AB \wedge b \geq 0 \wedge \neg(b > 0)) \rightarrow y = AB$
theorem
17 $\{B \geq 0\}$ `mult(A,B)` $\{y + ab = AB \wedge b \geq 0 \wedge \neg(b > 0)\}$
consequence left on 15 and 14
18 $\{B \geq 0\}$ `mult(A,B)` $\{y = AB\}$
consequence right on 16 and 17

F-35



“Systems designed with 1970s technology couldn’t be easily upgraded, because the process of testing to make sure they are ‘military grade’ takes years - by which time the technology is out of date. This is why we have 50 year old aircraft carriers, with 30 year old software [running on them].”

–David E. Sanger, “The Perfect Weapon”, pp. 256-257

Exciting and important question:

how to test and evaluate software

Sobering Stats

- NSB: engineering degrees dropped 20% since 1985
- ACT: less than 6% high school seniors plan to take engineering
- AFS: 0.8% students plan to major in math
- Intel Science Fair: 6million Chinese students applied; 65K US students did (~100:1)
- 50% of US Engineering PhDs go to foreign students
- In next decade, 90% of world scientists & engineers will reside in Asia

Michael Soltys

MSCS Grad Advisor and Chair of CS/IT

michael.soltys@csuci.edu
@MichaelMSoltys
http://www.msoltys.com