The Times, They Are A Changing

Dennis J. Frailey
(Retired) Principal Fellow, Raytheon Company
Adjunct Professor of Computer Science, SMU
Frailey@ACM.ORG
Frailey@Lyle.smu.edu

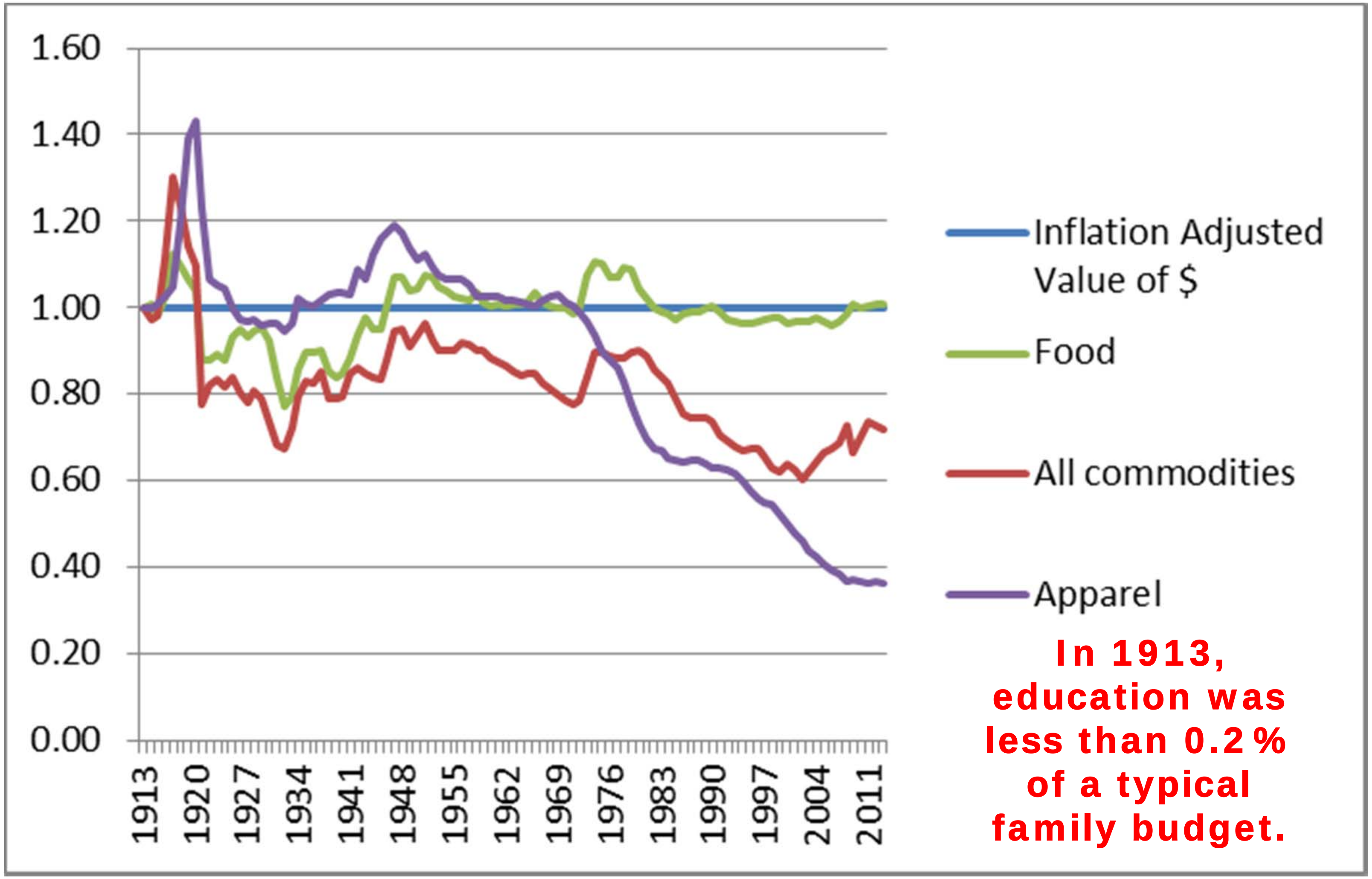
Presented at CSEET 2014

Outline

- **The Need for Change**
  - Economic Data
- **Resistance to Change**
  - Math vs CS
- **Changes in What we Teach**
  - Changes in Computing / SW Engineering
- **Changes in How we Teach**
  - MOOCs, flipped classrooms, and beyond
- **One Way Forward**
  - Cooperation with Industry
  - Academic Freedom vs Voice of the Customer

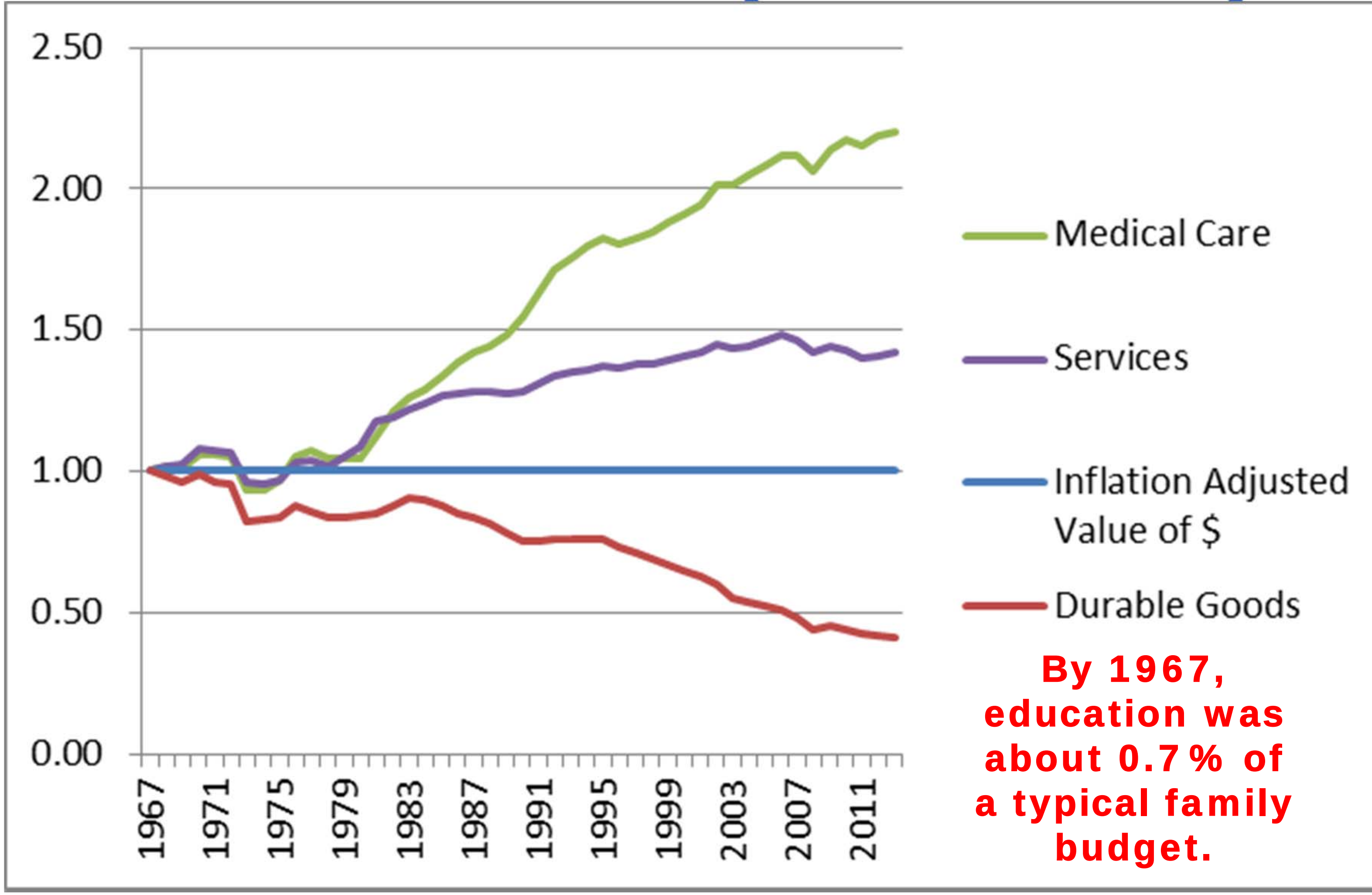
The Need for Change

Inflation-Adjusted Cost of Various Items Since 1913



Source: US Bureau of Labor Statistics -- http://www.bls.gov/cpi/data.htm

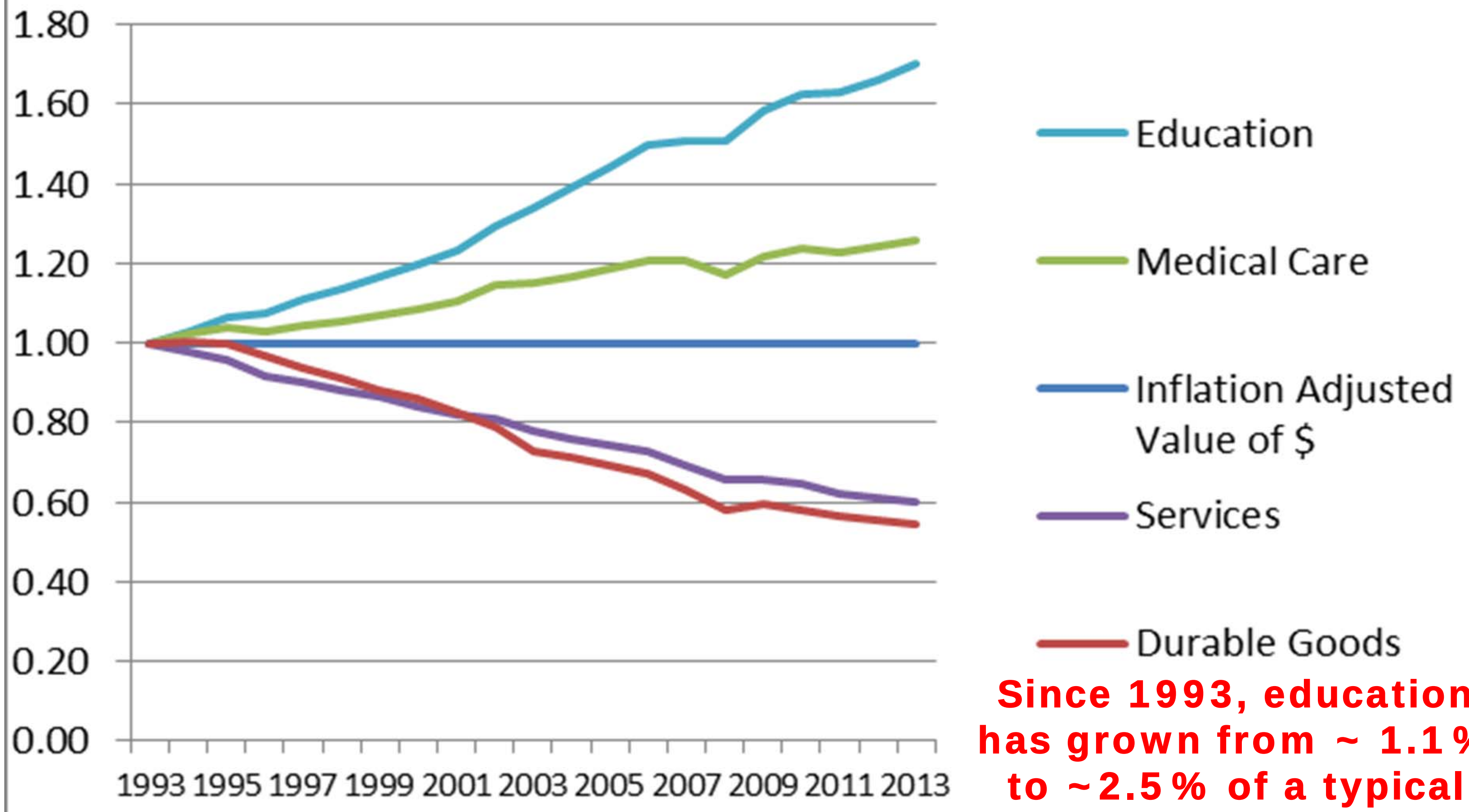
Durable Goods, Services and Medical Care (since 1967)



By 1967, education was about 0.7% of a typical family budget.

Source: US Bureau of Labor Statistics -- http://www.bls.gov/cpi/data.htm

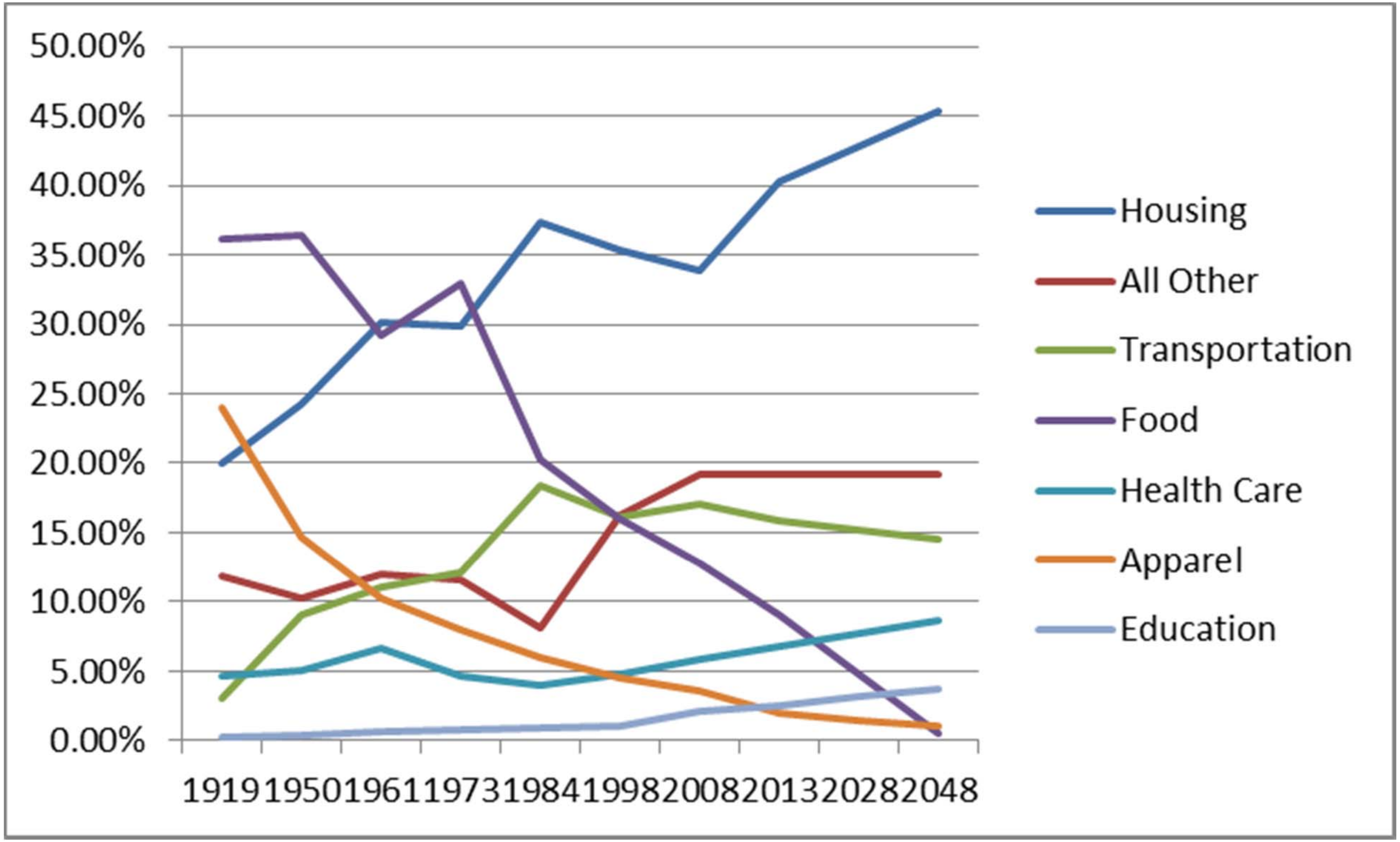
Rates of Growth – 1993-2013



Since 1993, education has grown from ~ 1.1% to ~2.5% of a typical family budget.

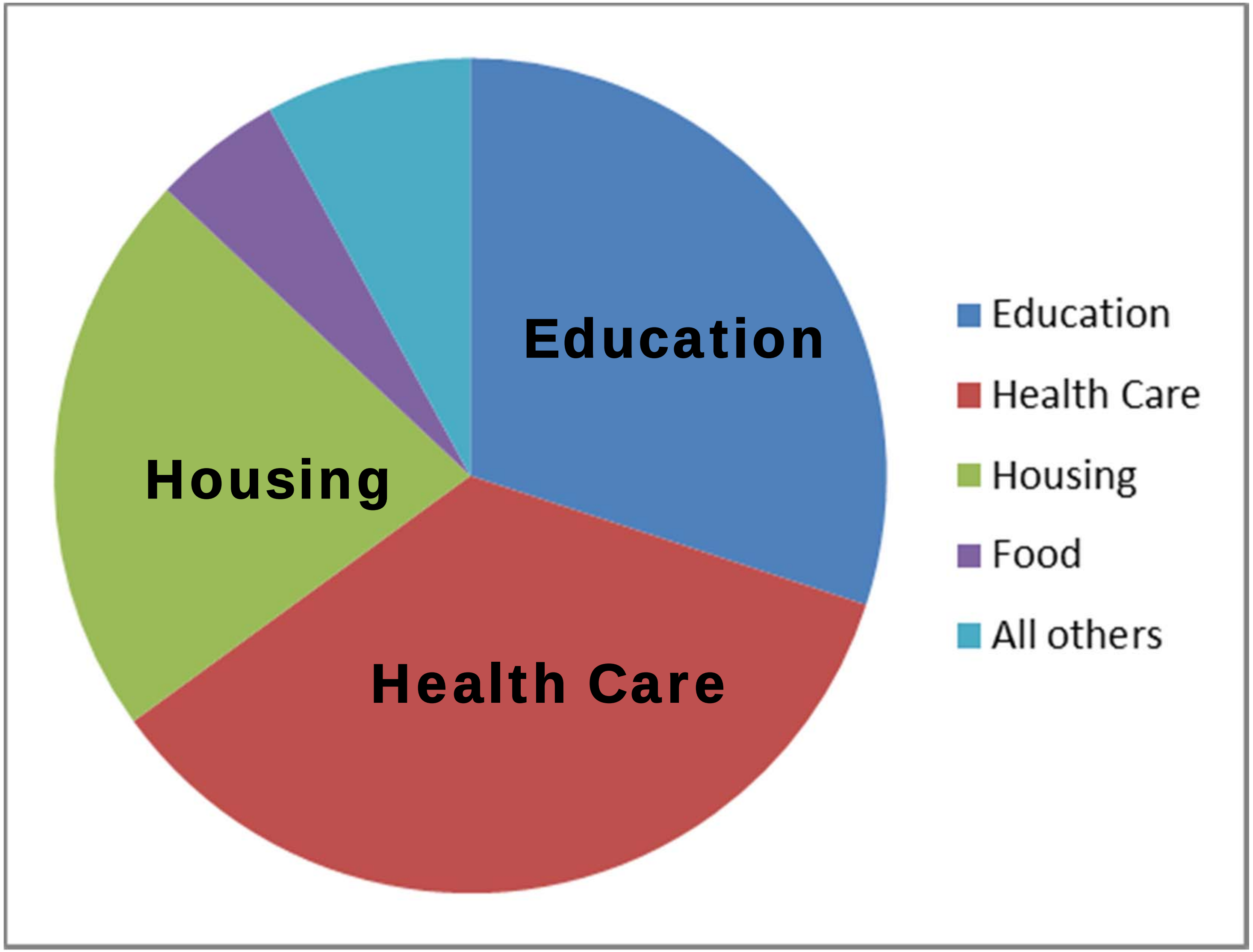
Source: US Bureau of Labor Statistics -- http://www.bls.gov/cpi/data.htm

Elements of a Typical US Family Budget 1919 through 2048 (projected)



Source data: US Dept of Labor Statistics and Johnson, Rogers and Tan, "A century of family budgets in the United States", *Monthly Labor Review*, May, 2001

The Future Family Budget?



The Future Family Home?

Other Factors Motivating Change

- **People are more mobile**
  - They expect to access things from mobile devices and from anywhere
- **Expansion of knowledge**
  - The computing field continues to grow
  - Students can't learn everything
  - "Just in Time" training and education are more common
- **Expansion of knowledge sources**
  - Internet, for-profit education providers, etc.
  - Students need ways to integrate the knowledge into a cogent understanding

Resistance to Change

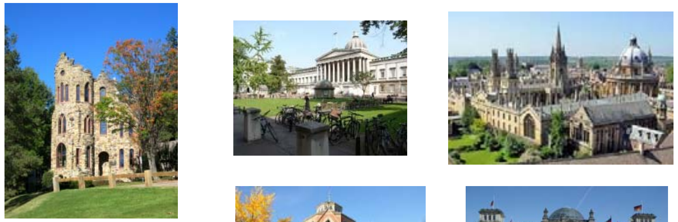
John Silbur's Research on Organizations that Resist Change

- **Of 68 institutions that have survived the past 500 years**
  - 2 are churches (Catholic, Lutheran)

  - 2 are governments (Iceland, Isle of Man)

Silber, John, "Straight Shooting: What's Wrong with America & How to Fix it", 1990

64 are Universities

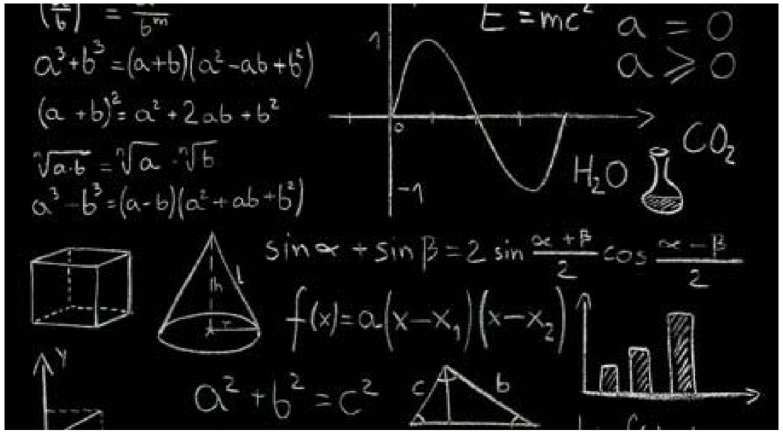


Choosing a Career in 1966 – Mathematics or Computer Science

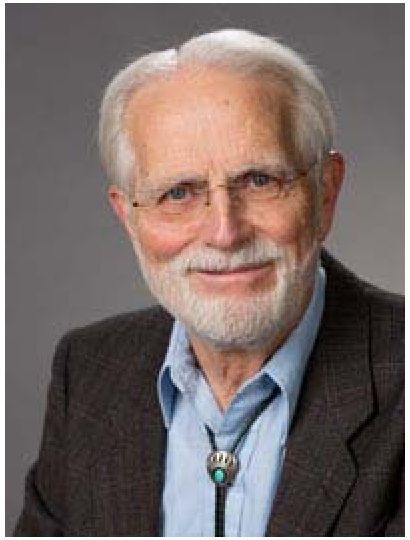
Math Faculty

The Math Curriculum

- **Real / Complex Analysis**
- **Topology**
- **Differential Topology**
- **Complex Manifolds**
- **Algebra**
- **Differential Equations**
- **Combinatorics**
- **Probability and Statistics**
- **Algebraic Number Theory**
- **Field Theory**
- **Etc.**



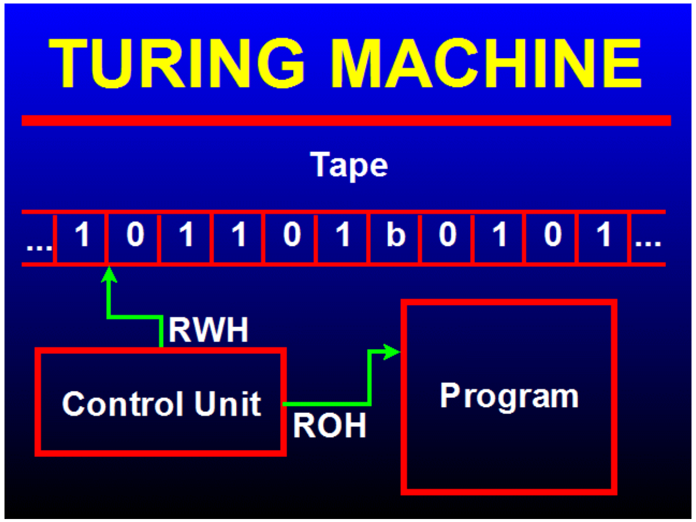
What Allen H Brady Taught



- Turing Machines
- Symbolic Logic
- Automata Theory
- Formal Languages
- Computational Complexity

$$T = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$$

Where :
Q = Set of states
Σ = Input alphabet
Γ = Tape alphabet $(\Sigma \subseteq \Gamma)$
δ = Transition function
$\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$
q_0 = Start state
B = Blank symbol
F = Set of accepting states $(F \subseteq Q)$

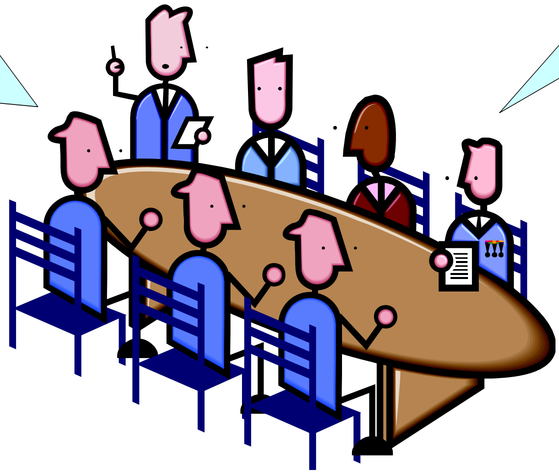


Reasons to Choose Math over Computer Science

Computing lacks the intellectual depth of mathematics and you would waste your talents by going into that field

Computer science is a fad, likely to die out quickly and you would have a worthless degree

You can be assured of a **stable, interesting, comfortable career in mathematics**



So What Happened?

PhD in Computer Science in 1971

Computer Science Professor for 7 years

Adjunct Professor of Computer Science & Software Engineering for 37 years (so far)

Started computing career in 1962 as a Fortran programmer

Worked in industry for 40+ years:

compiler design, computer design, operating systems, many software applications

Changes in What We Teach

1966-67 Grad Catalog

658. THEORY OF TESTS: ESTIMATION AND DECISION II. Sem. 2. Class 3, cr. 3. Prerequisite: STAT 657.
Decision theoretic approach to statistical problems, complete class theorems, Bayes solutions, minimax theorem, sequential decision problems, optimality proof of sequential probability ratio test, connections with game theory.

690. SEMINAR. SS. Sem. 1 and 2, cr. 1 to 3.

691. SEMINAR IN PROBABILITY THEORY. SS. Sem. 1 and 2, cr. 1 to 3.

695. SEMINAR IN MATHEMATICAL STATISTICS. SS. Sem. 1 and 2, cr. 1 to 3.

698. RESEARCH. M.S. Thesis.

699. RESEARCH. Ph.D. Thesis.

Graduate Courses in Computer Sciences

500. COMPUTING AND PROGRAMING SYSTEMS. Sem. 1 and 2. Class 3, cr. 3. Prerequisite: CS 400 or CS 580.
Computer organization as it affects programing. Magnetic tape systems. Error detecting and error correcting codes. Disc, drum, and other random access systems. Input-Output programs. Buffering, simultaneous operation, interrupt handling. Introduction to the design of compiling systems. Programing languages based on ALGOL. Backus Normal Form. Recursive procedures, dynamic storage allocation and other ALGOL features. Professor Rosen.

514. NUMERICAL ANALYSIS. Sem. 1, Class 3, cr. 3. Prerequisite: MA 510; corequisite: CS 200 or equivalent knowledge of programing.
Finite difference calculus, finite difference equations; iterative methods for solving nonlinear equations; differentiation and integration formulas; the solution of ordinary differential equations; the solution of linear systems; round-off error bounds. Professor Maybee.

515. NUMERICAL ANALYSIS OF LINEAR SYSTEMS. Sem. 2. Class 3, cr. 3. Prerequisites: MA 511 or MA 351, and MA 441; corequisite: CS 200 or equivalent knowledge of programming.
Computational aspects of linear algebra; linear equations and matrices, direct and iterative methods; eigenvalues and eigenvectors of matrices; error analysis. Professor Rice.

520. MATHEMATICAL PROGRAMING. Sem. 1. Class 3, cr. 3. Prerequisites: MA 511 or MA 351, MA 441.
Fundamental theorems from the theory of linear inequalities. Simplex Method and variants; gradient methods; special techniques for solving integer programing problems; survey of applications; survey of methods in current use on various computer systems; consideration is directed throughout to that class of algorithms created to solve mathematical programming problems which are appropriate for implementation on modern digital computers. Professor Pyle.

560. INFORMATION STORAGE AND RETRIEVAL. Sem. 1. Class 3, cr. 3. Prerequisites: CS 400 or CS 580.
Computer-based information storage and retrieval systems. Selective dissemination of information; document retrieval. Indexing; file organization; search techniques. Automatic classification and abstracting. The structure of information systems.

580. INTRODUCTION TO DATA PROCESSING. Sem. 1 and 2. Class 3, cr. 3. May not be used in the primary area in mathematics.
Problem solving and programing techniques and languages; students use computer-oriented and problem-oriented languages in solving problems common to the non-physical sciences. Professor Pyle.

(**581. INTRODUCTION TO LOGIC AND BOOLEAN ALGEBRA.** See MA 581.)

582. MATHEMATICAL THEORY OF FINITE AUTOMATA. Sem. 2. Class 3, cr. 3. Prerequisite: MA 581.
Structure theory of finite transition algebras and semi-groups, lattices of congruence relations on words; periodic sets of words; behavior of finite automata; transition graphs and regular canonical systems; Kleene's theory of regular expressions; Church's solvability-synthesis alogorithm and its extension to wider classes of design requirements; decision procedure for sequential calculus and its significance to the algorithmic design of sequential machines. Professor Büchi.

583. SIMULATION AND INFORMATION PROCESSING. Class 3, cr. 3. Prerequisites: MA 261 or MA 223, CS 400 or CS 580, STAT 311 or equivalent.
Simulation and modeling. Monte Carlo techniques; use of special simulation languages to simulate actual systems; structuring and use of information systems; information storage and retrieval.

584. RECURSIVE FUNCTIONS I. Sem. 1. Class 3, cr. 3. Prerequisite: MA 581.
Post canonical systems, computability, decision problems; recursive functions, Turring machines, Markov algorithms, recursive algorithms. Professors Büchi and Korfhage.

(**585. MATHEMATICAL LOGIC I.** See MA 585.)

590. TOPICS IN COMPUTER SCIENCES. Sem. 1 and 2. Class 1-5, cr. 1-5.
Directed study for students who wish to undertake individual reading and study on approved topics.

600. ADVANCED PROGRAMING SYSTEMS I. Sem. 2. Class 3, cr. 3. Prerequisite: CS 500.
Design of assemblers, compilers, and interpretive systems. Libraries of subroutines, macro-instructions, and generators. Operating systems for debugging and job sequencing. Systems storage allocation. Programing languages and their translators. List processing languages. Generalized compiling systems for producing compilers. Professor Rosen.

601. ADVANCED PROGRAMING SYSTEMS II. Sem. 2. Class 3, cr. 3. Prerequisite: CS 600.
Continuation of CS 600. Professor Rosen.

614. NUMERICAL SOLUTION OF ORDINARY DIFFERENTIAL EQUATIONS. Sem. 1. Class 3, cr. 3. Prerequisite: CS 514, MA 525, 543 or consent of instructor.
Numerical solution of initial-value problems by Runge-Kutta methods, general one-step methods, and multistep methods. Analysis of truncation error, discretization error, and rounding error. Stability of multistep methods. Numerical solution of boundary- and eigen-value problems by initial-value techniques and finite difference methods. Professor Gautschi.

615. NUMERICAL SOLUTION OF PARTIAL DIFFERENTIAL EQUATIONS. Sem. 2. Class 3, cr. 3. Prerequisite: CS 515. Desirable: MA 523 or MA 520.
The numerical solution of hyperbolic, parabolic alliptic equations by finite difference methods; iterative methods (Gauss-Seidel, over-relaxation, alternating direction) for solving elliptic equations; discretization and round-off errors; explicit and implicit methods for parabolic and hyperbolic systems; the method of characteristics; the concept of stability for initial value problems. Professors Maybee and Conte.

(**616. THE THEORY OF APPROXIMATION.** See MA 616.)

(**681. ARTIFICIAL INTELLIGENCE.** See EE 681.)

(**682. THEORY OF AUTOMATA.** See EE 682.)

684. RECURSIVE FUNCTIONS II. Sem. 2. Class 3, cr. 3. Prerequisites: MA 553, CS 584.
Kleene hieararchies, of nonrecursive sets; advanced theory of recursivity, recursive unsolvability types; advanced topics on Turing machines. Professor Buchi and Korfhage.

(**685. MATHEMATICAL LOGIC II.** See MA 685.)

690. SEMINAR ON TOPICS IN COMPUTER SCIENCES. Sem. 1 and 2. Class 1-5, cr. 0-5.

698. RESEARCH. M.S. Thesis.

699. RESEARCH. Ph.D. Thesis.

MECHANICAL ENGINEERING

OFFICERS OF INSTRUCTION
P. W. McFadden, Head of the School

Professors: V. E. Bergdolt, Ph.D.; C. L. Brown, Ph.D.; J. B. Chaddock, Sc.D.; P. F. Chenea, Ph.D.; D. S. Clark, M.S.E.; Raymond Cohen, Ph.D.; W. B. Cottingham, Ph.D.; W E. Fontaine, M.S.M.E.; R. W. Fox, Ph.D.; R. J. Grosh, Ph.D.; A. S. Hall, Ph.D.; G. A. Hawkins, Ph.D.; A. R. Holowenko, M.S.E.; R. H. Kohr, Ph.D.; Wolfgang Leidenfrost, Dr. Ing.; P. W. McFadden, Ph.D.; C. W. Messersmith, M.S.E.; Joseph Modrey, Ph.D.; F. B. Morse, M.S.M.E.; Rufus Oldenburger, Ph.D.; J. R. Osborn, Ph.D.; B. E. Quinn, Ph.D.; B. A. Reese, Ph.D.; Y. S. Touloukian, Ph.D.; R. Viskanta, Ph.D.; C. F. Warner, Ph.D.

Associate Professors: D. E. Abbott, Ph.D.; R. E. Goodson, Ph.D.; J. F. Hamilton, Ph.D.; J. H. Hoffman, Ph.D.; H. G. Laughlin, M.S.M.E.; C. C. Oliver, Ph.D.; R. A. Olsen, M.S.M.E.; K. R. Purdy, Ph.D.; C. W. Rezek, Ph.D.; R. J. Schoenhals, Ph.D.; J. K. Stene, Ph.D.; Kenneth Wark, Ph.D.; E. J. Wellman, Ph.D.; E. R. F. Winter, Ph.D.

Assistant Professors: R. D. Gustafson, Ph.D.; M. R. L'Ecuyer, Ph.D.; J. B. Lusk, M.S.M.E.; J. G. Skifstad, Ph.D.; H. D. Thompson, Ph.D.

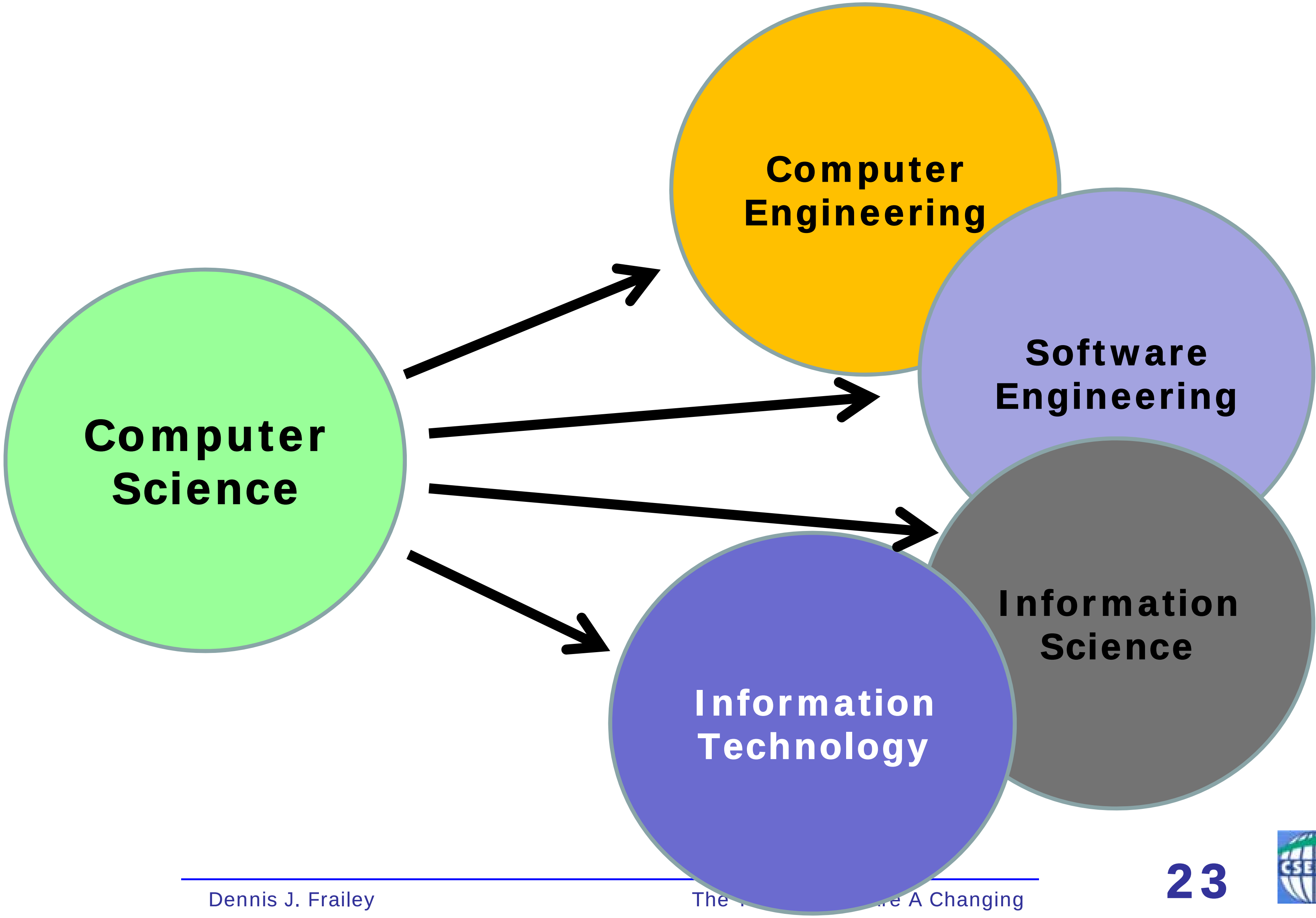
Areas of Graduate Study. Graduate students in mechanical engineering may select a primary area of concentration in one or more of the following areas: automatic control; kinematics; dynamics; vibration, stress analysis; design; cli-

1966-67 Graduate Computer Science Courses

PURDUE UNIVERSITY

- Computing and Programming Systems
- Advanced Programming Systems
- Information Storage and Retrieval
- Simulation and Information Processing
- Artificial Intelligence
- Numerical Analysis
- Numerical Analysis of Linear Systems
- Mathematical Programming
- Introduction to Logic and Boolean Algebra
- Mathematical Theory of Finite Automata
- Recursive Functions
- Mathematical Logic
- Numerical Solution of Ordinary Differential Equations
- Numerical Solution of Partial Differential Equations
- Theory of Approximation

The Growth of Computer Science



1966-67 Graduate Computer Science Courses

- Computing and Programming Systems
- Advanced Programming Systems
- Information Storage and Retrieval
- Simulation and Information Processing
- Artificial Intelligence
- Numerical Analysis
- Numerical Analysis of Linear Systems
- Mathematical Programming
- Introduction to Logic and Boolean Algebra
- Mathematical Theory of Finite Automata
- Recursive Functions
- Mathematical Logic
- Numerical Solution of Ordinary Differential Equations
- Numerical Solution of Partial Differential Equations
- Theory of Approximation

2014 Graduate Computer Science Courses

- Systems I and II
- Databases
- Simulation and Modeling
- Artificial Intelligence
- Numerical Computing
- Algorithms
- Bioinformatics
- Complexity
- Distributed Systems
- Geometric Modeling, Visualization & Graphics Recursive Functions
- Distributed Systems
- Data Mining
- Parallel and Distributed Computing Theory of Approximation
- Security
- Software Engineering

PURDUE UNIVERSITY

The Changes in Computing

	1950's	1960's	1970's	1980's	1990's	2000's	2010's
Hardware Technology	Vacuum Tubes	Transis tors	Integrated Circuits	LSI	VLSI	ULSI	Nano-systems
Programming Languages	Binary Assembly	Fortran Cobol	Pascal Algol	Ada C Lisp	C++ GUI Java	C# PhP XML	J2EE, EJB
Computing Paradigm	1 user Mainframe	Batch	Time Sharing	Personal Computer	LAN, WEB	.NET, SOA	Mobile
Operating System	none	1 user	multi user	multi user linked	networked	Web, Open source	Cloud, Android, iPhone
Data Base Methods	none	Linear (tapes)	Hier-archical	Relational	Object Oriented	SQL, X Query	SQLJ, OLAP, JDBC
Software Design	pad and pencil	Flow Charts	Structured Design	Data Flow	Object Oriented	RAD, XP, RUP	MDE

Computing is a Changing Field

- Every ten years the field is very different
- The half-life of computer knowledge is five years
- One must plan on a career of continuous learning

Changes in How We Teach

Challenges / Opportunities

- Cost / Productivity
- Effectiveness
- Global Reach & Needs

Assets

- Technology
- Creativity
- Software Engineers

Cost and Productivity

Classroom Evolution

Classroom Evolution

Classroom Evolution

Classroom Evolution

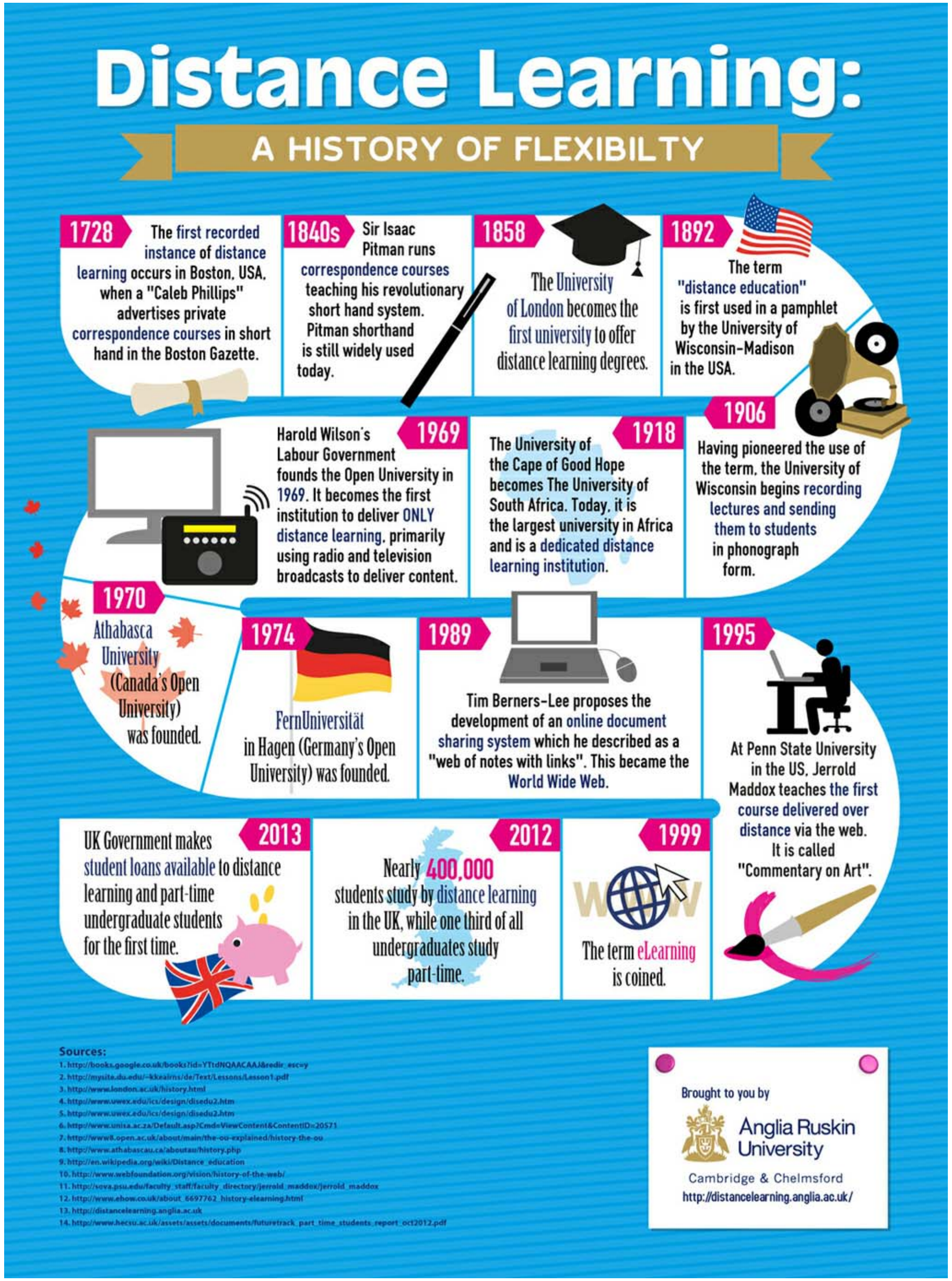
Classroom Evolution

Classroom Evolution

Classroom Evolution?

Distance Education

It's been around for a while



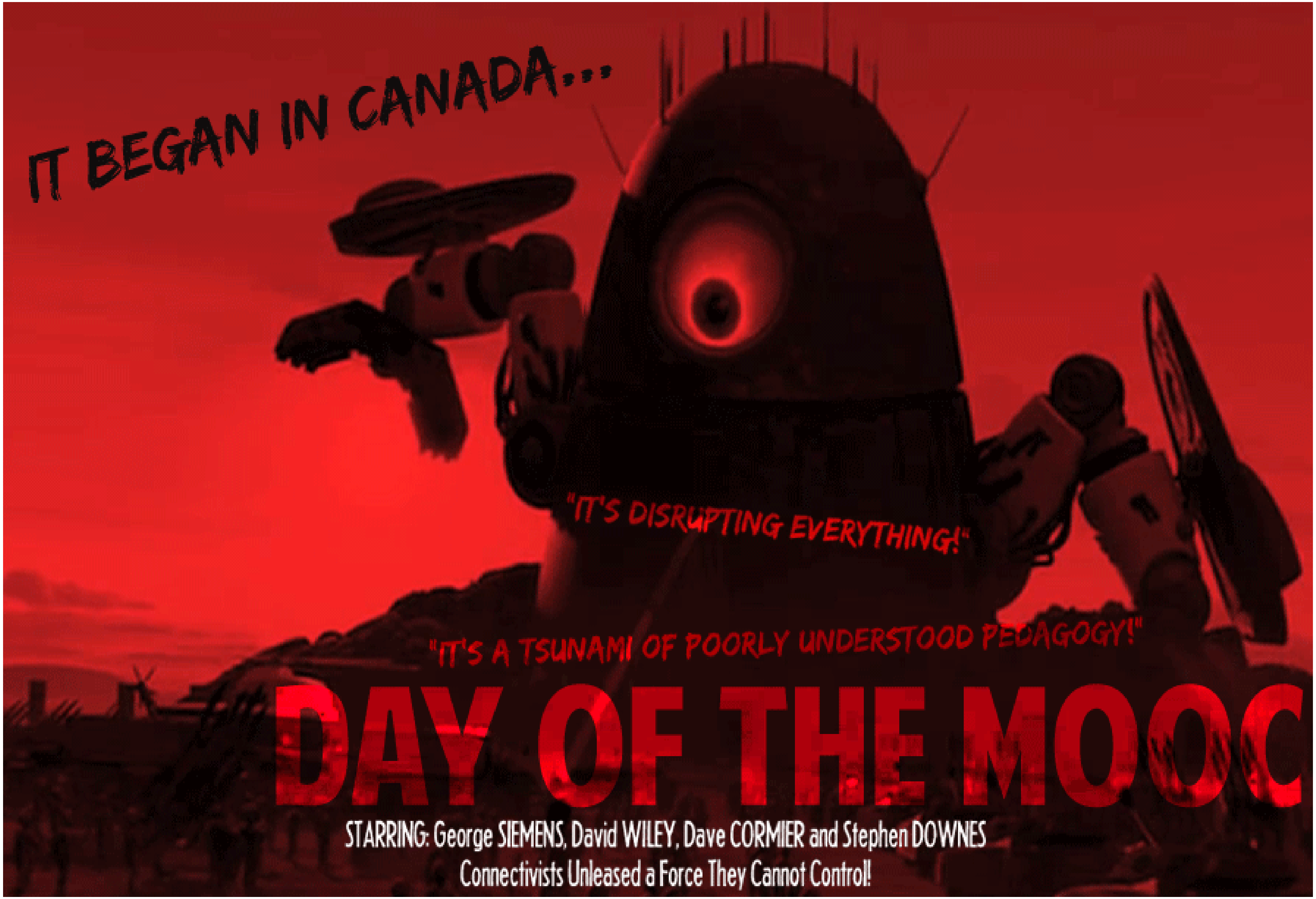
Student Observations

Recorded is better than live because I can watch it any time I want to.

I can rerun the recording if I did not understand.

Better Organized

I can schedule around personal obligations.

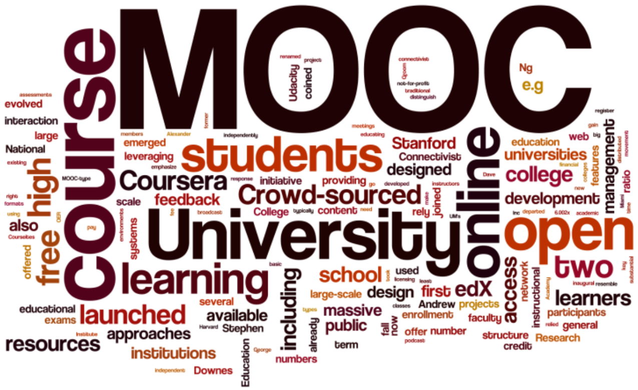


http://www.michaelbransonsmith.net/blog/wp-content/uploads/2012/12/DAY-OF-THE-MOOC.gif

Effectiveness

Innovations in Distance Education

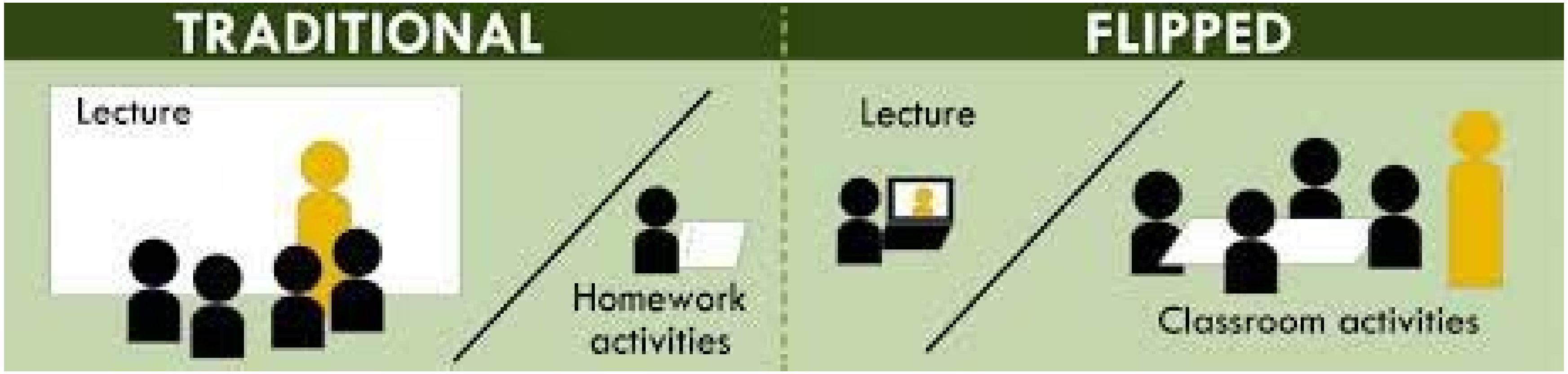
- **DuoLingo.com**
  - Language education
- **Code.org**
  - Programming education
- **TED.com and TED-ed.com**
  - TED is a nonprofit devoted to spreading ideas, usually in the form of short, powerful talks (18 minutes or less).
  - TED-ed is developing courses based on TED topics
- **...**

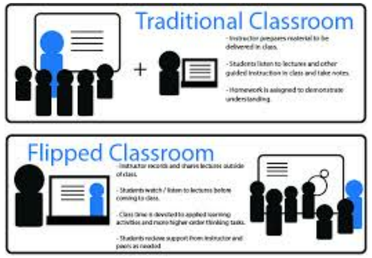


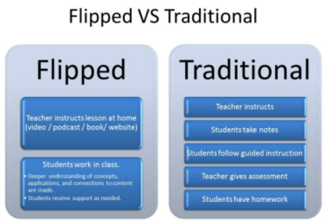
My Lessons Learned about Distance Education

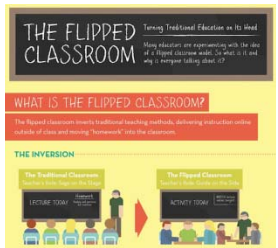
- **Audio is More Important than Video**
  - They don't care what you look like
  - But they have to clearly understand what you say
- **"Real" Time Isn't Very Important**
  - Students prefer the ability to time shift
- **You Have to be Well Prepared**
  - Materials ready in advance
  - FAQs and other helpful aids
- **You have to Change the Way You Teach**
  - Flexible schedules and due dates
  - Assignments tailored to the needs of distance students
  - Lots of grading and interactive support

Classroom Flipping



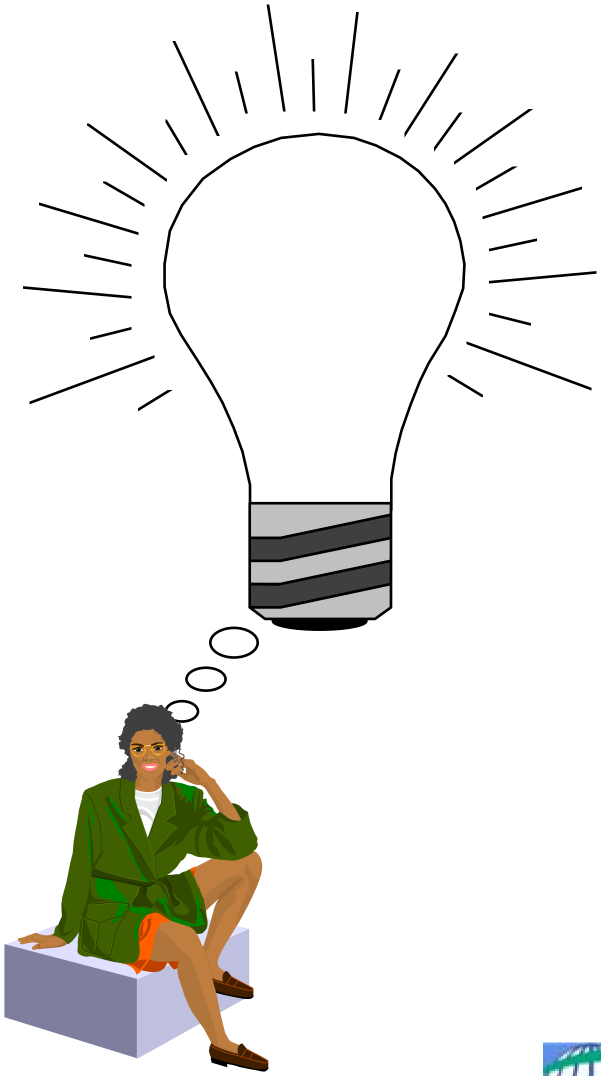






We Must Innovate!

- **Re-invent**
- **Re-engineer**
- **Re-think the way we educate**
- **… or someone else will do it for us**
  - they already are!

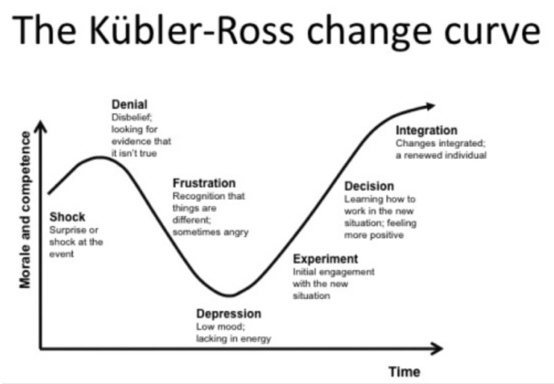


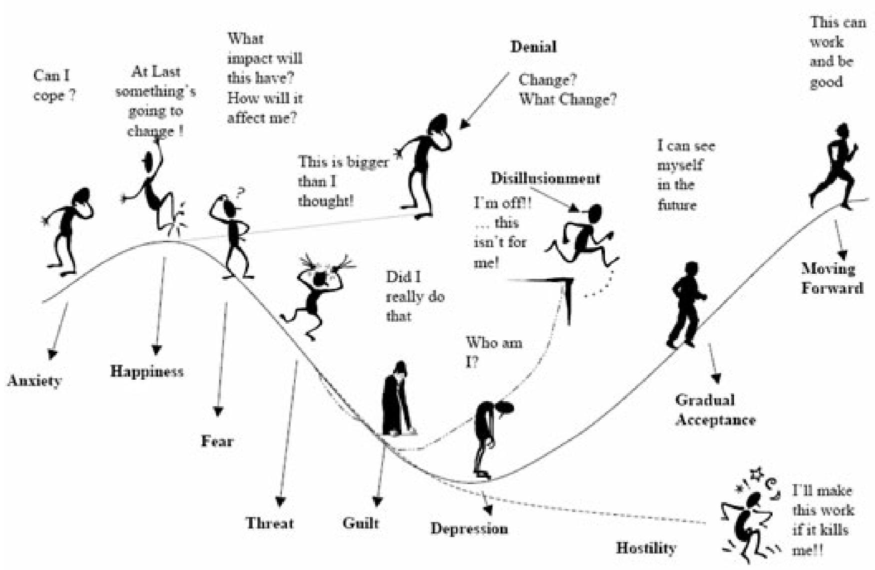
What We Teach
How We Teach

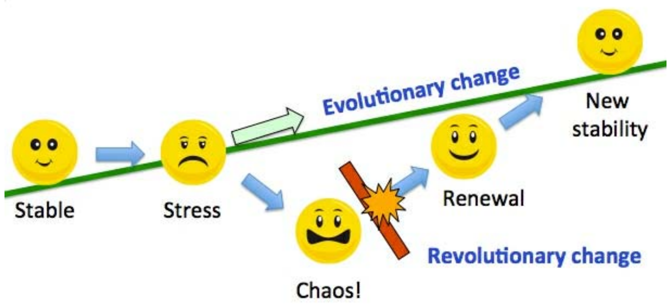
Challenges

The Change Process?

The Change Process







It gets worse before it gets better

One Way Forward

Cooperation Between Universities and Industry

- **U. of Maryland w Northrop Grumman**
  - New specialization in Cyber Security
  - (A special track in the Computer Science program)
- **Ohio State U w IBM**
  - Big Data Analytics center
- **Murray State University, Kentucky w Local Industry**
  - Retooled Engineering Program
- **State University of New York w Private Donors**
  - College of Nanoscale Engineering

– Wall Street Journal, April 8, 2014, pp A1, A4

Q&A

- **What's different about these programs?**
  – Aimed at **undergraduates**
- **Why are corporations doing this?**
  – They are "**concerned about a mismatch between their needs and graduates' skills**"
- **What are the benefits to students?**
  – "Pathways to good internships and high paying jobs."

What About Academic Integrity?

Academic Integrity

- Freedom from bias and biased influence
- Research integrity
- Honesty and rigor in the pursuit of knowledge
- Adherence to Moral and ethical principles
- Honor codes

Voice of the Customer

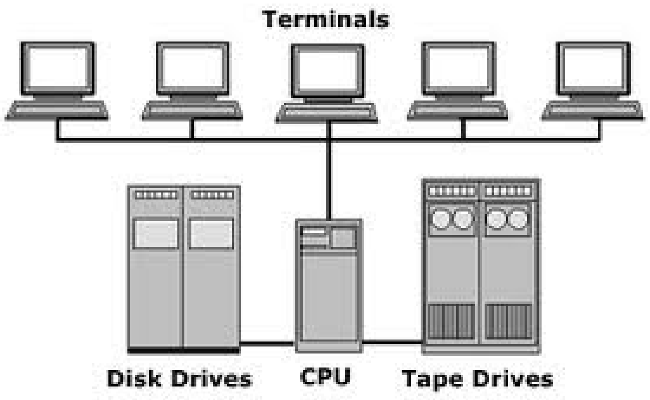
- Understanding the customer's needs, expectations, preferences and aversions
- Proper prioritization
- Common language with customer
- Avoiding "engineering arrogance"

Engineering Arrogance

The tendency of engineers to think they know better than the customer.

A Little Bit of History

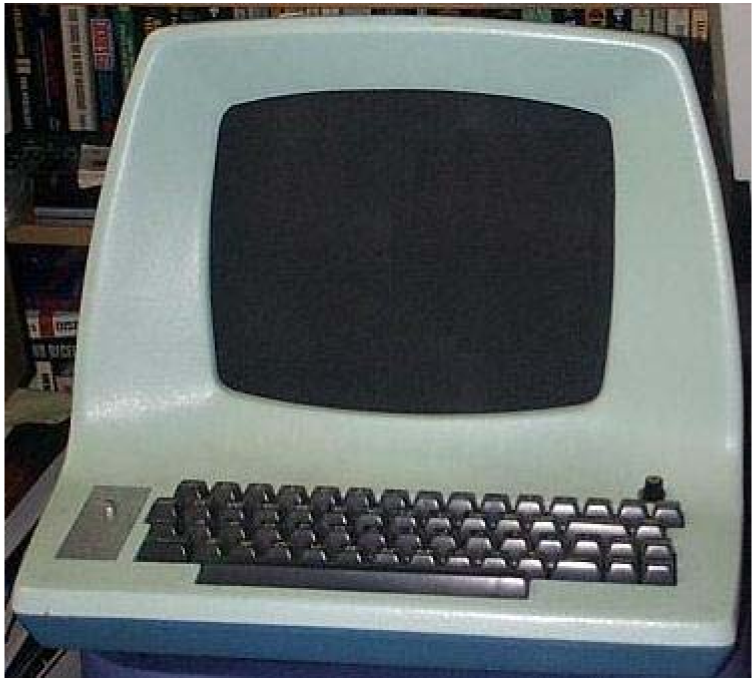
- In the 1970's many universities used "dumb" terminals to access "minicomputers" or "mainframe" computers.



- Many of those terminals were Teletypes, which had no graphics capability and whose keyboards resembled those of typewriters.

First Generation Video Terminals

- **Keyboards resembled typewriter keyboards**
- **<u>Did not have cursor control keys</u>.**



Typical Cursor Control in Early Video Terminals

- “control” U for “up”
- “control” D for “down”
- “control” L for “left”
- “control” R for “right”

Bit Mapped Graphics

- **These terminals used "bit mapped" graphics**

- **Due to limitations on memory size and speed, they displayed mostly text**
- **<u>More sophisticated graphics were difficult to display</u>**

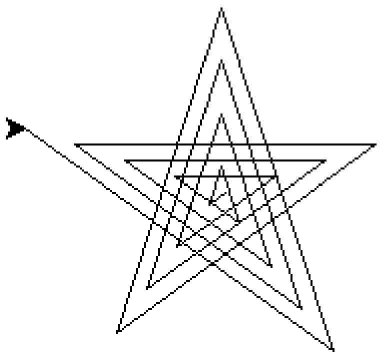
Vector Graphics Terminals

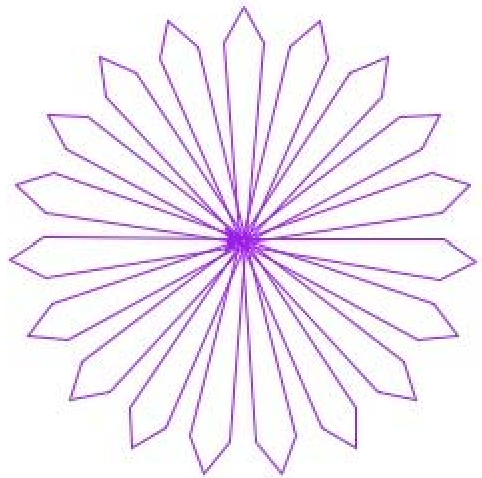
- **Could draw lines and other primitive geometric shapes automatically**
  - Lines
  - Curves
  - Polygons
  - ...

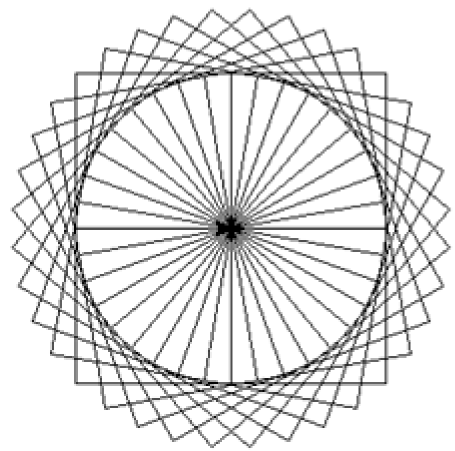
- **But they were very expensive.**
  - Too expensive for most consumer applications
  - Few students had access to them
- **And the early ones still lacked cursor control keys**

Logo and Turtle Graphics

- In the 1960's, the MIT AI lab developed a graphics programming language called Logo
- … and a method of drawing pictures called Turtle Graphics
- … resulting in software that would allow drawing of pictures on the screens of vector graphics terminals







Logo and Turtle Graphics on Texas Instruments' Home Computer

- Intended for young children
- To draw pictures on an inexpensive computer with bit mapped graphics.
- In the 1970's, TI contracted with MIT to port the LOGO language and many applications to a home computer

Emerging Personal Computers had Cursor Control Keys

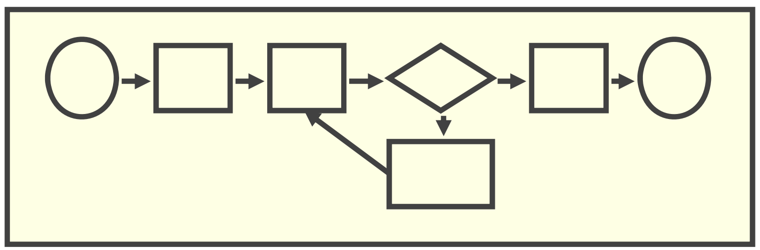
But the MIT students and faculty initially refused to support cursor control keys because "'control' R is more intuitive than the → cursor control key"

Two Keys instead of One?

Our Students ~~Will Make~~ it Happen

A Special Opportunity for Software Engineers

- We are immersed in the technology
- We understand the problems
- We know how to make applications faster, smaller, & more efficient
- We use techniques such as re-engineering, optimization, etc.
- And our students are sometimes further ahead than their teachers!



Concluding Remarks

- ✓ **Education must innovate and change**
- ✓ **Software engineers and software engineering educators are uniquely well qualified to help make this happen**
  - ➢ We know the technology
  - ➢ We understand processes
- ✓ **If we don’t, somebody else will**
- ✓ **We can lead the way**

As the present now
Will later be past
The order is rapidly fadin'
And the first one now
Will later be last
For the times they are a-changin'.

Bob Dylan, 1964

Questions?