Thoughts from a Strayed Attendee: Success -> Complexity -> Society

High Performance Transaction Systems
HPTS 2019, 6-Nov-2019

Dr. Alfred Z. Spector
Chief Technical Officer and Head of Engineering
azs@twosigma.com

TWO SIGMA

Abstract of HPTS 2019 Talk

The multi-trillion-fold (!) increase in the capability of computation over the past 60 years, when coupled with global connectivity and vast data, has made for vibrant fields of research that are growing with no end in sight.

- This explosive growth in computing and data has led to very excellent results in the **core** of the field: e.g., ever more capable and creative algorithms; the capability to build vast, globally networked systems that support most of the world's population; and the effective solutions to grand challenge problems, such as speech and image recognition.
- However, many challenges remain in the the core. For example, we still have trouble designing and building robust, large scale systems, and we need breakthroughs in **software engineering** and **software architecture** to allow applications to progress with greater efficiency and robustness. In particular, our infrastructure is too complex and non-orthogonal in capabilities.
- Finally, with growing import of computing in all aspects of society, important research areas abound at the **intersection of computing and ethics/public policy**. The critical nature of new applications will place a stronger set of requirements, perhaps even regulatory ones, for highly robust systems, further reinforcing the need for improvements in software engineering and systems.

In this talk, I'll discuss the challenges that we have and two outcomes that I predict will ensue: the important growth of cloud computing utilities and —probably— increased regulation.

Framework

- Successes and the Expanding Sphere
- Complexity and the Cloud
- Societal Challenges Beyond the Engineering

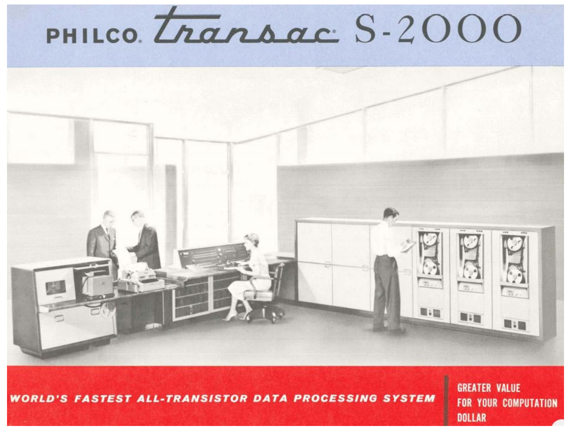
Disclaimers:

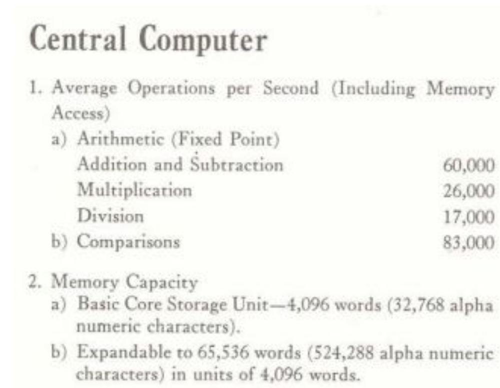
No investment advice -- informational only
My views, not Two Sigma's views
Any use of images, logos is informational only: no endorsement implied

Successes

Computers are ~1 Trillion Times Better over 60 years

Philco TRANSAC 210-212
~1958
~50K FLOPS
~$20M 2018 dollars





Central Computer

1. Average Operations per Second (Including Memory Access)
   a) Arithmetic (Fixed Point)
      - Addition and Subtraction — 60,000
      - Multiplication — 26,000
      - Division — 17,000
   b) Comparisons — 83,000
2. Memory Capacity
   a) Basic Core Storage Unit—4,096 words (32,768 alpha numeric characters).
   b) Expandable to 65,536 words (524,288 alpha numeric characters) in units of 4,096 words.

Nvidia Titan Xp
~2017
~10 TFlops
~1,000 2018 dollars

GPU Engine Specs:

NVIDIA CUDA® Cores	3840
Boost Clock (MHz)	1582

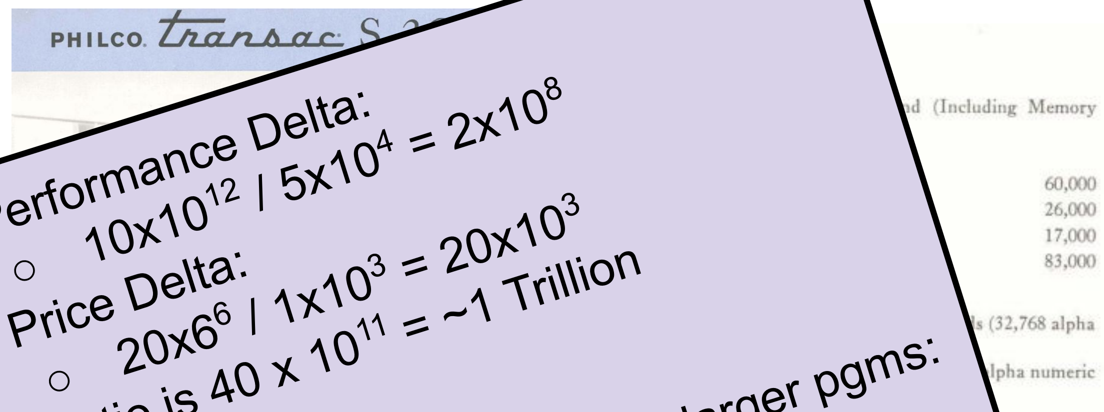
Memory Specs:

Memory Speed	11.4 Gbps
Standard Memory Config	12 GB GDDR5X
Memory Interface Width	384-bit
Memory Bandwidth (GB/sec)	547.7 GB/s

Successes

Computers are ~1 Trillion Times Better over 60 years

Philco TRANSAC 210-212
~1958
~50K FLOPS
~$20M 2018 dol



- Performance Delta:
  - $10x10^{12}$ / $5x10^{4}$ = $2x10^{8}$
- Price Delta:
  - $20x6^{6}$ / $1x10^{3}$ = $20x10^{3}$
- Ratio is 40 x 10^{11} = ~1 Trillion
- 62% CAG
- And, we can construct much larger pgms: Two Sigma runs ~256 Gbyte images

Nvidia Titan Xp
~2017
~10 TFlops
~1,000 2018 dollars

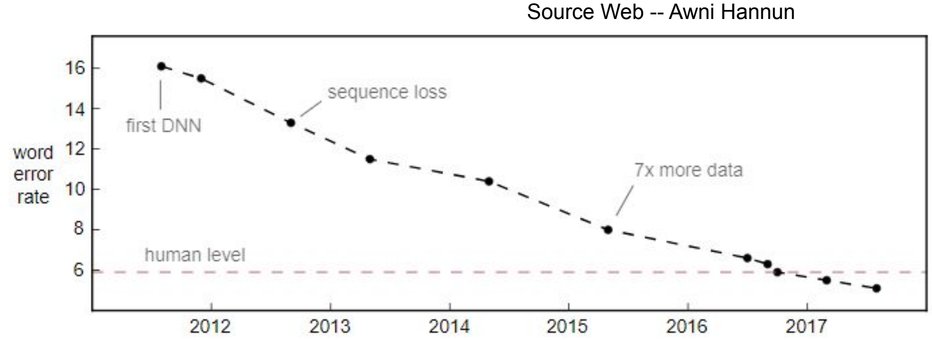
	3840
...ck (MHz)	1582

Memory Specs:

Memory Speed	11.4 Gbps
Standard Memory Config	12 GB GDDR5X
Memory Interface Width	384-bit
Memory Bandwidth (GB/sec)	547.7 GB/s

Successes

Speech Recognition:

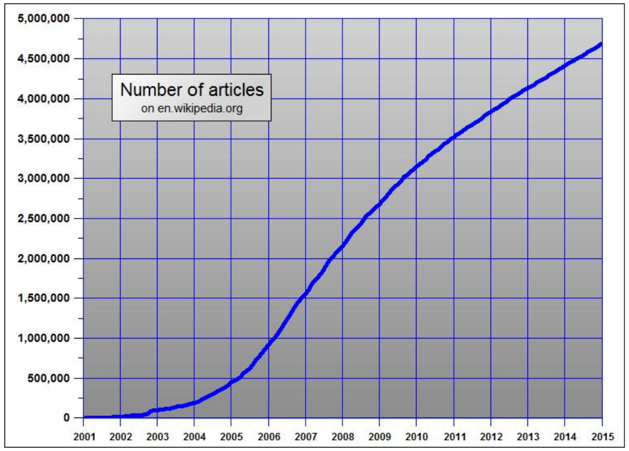


Similar results in: Image recognition, translation, and more.

Library of Alexandria:

Google ~5 x 10^9 searches/day[1]

>100 Trillion unique web pages[2]



Facebook alone has 2 billion users

[1]SearchEngineLand derived from Google Statement.[2]Extrapolating from 2013 data. [2][Facebook Data]

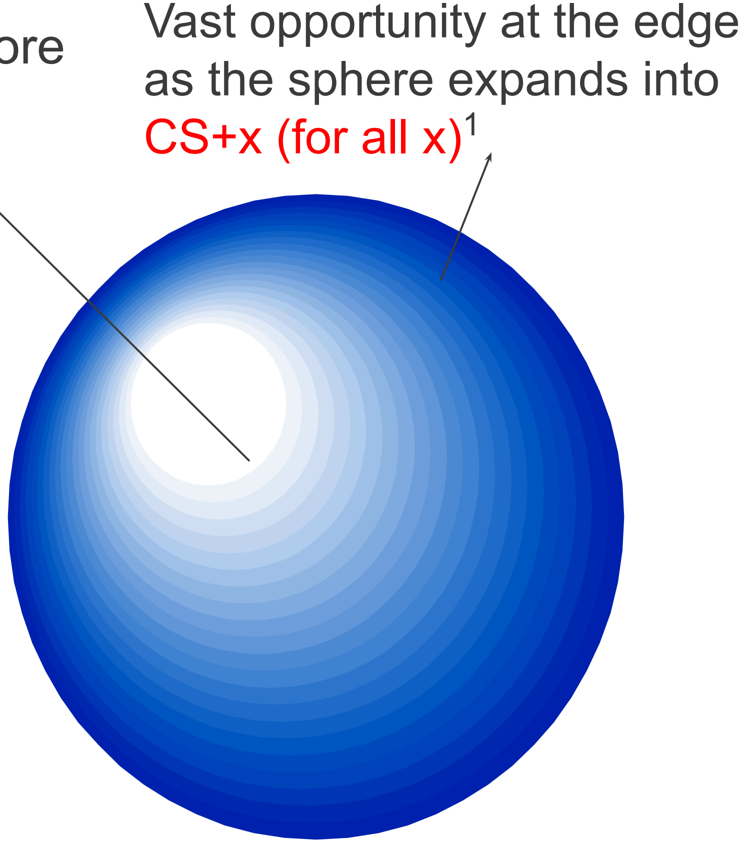
Success and the Expanding Sphere

A survey of CS-induced change could continue for the full hour...

The core of CS

1. Mathematical Analysis
2. Engineering Methodologies
3. **Empiricism (relatively recent, increase in emphasis)**

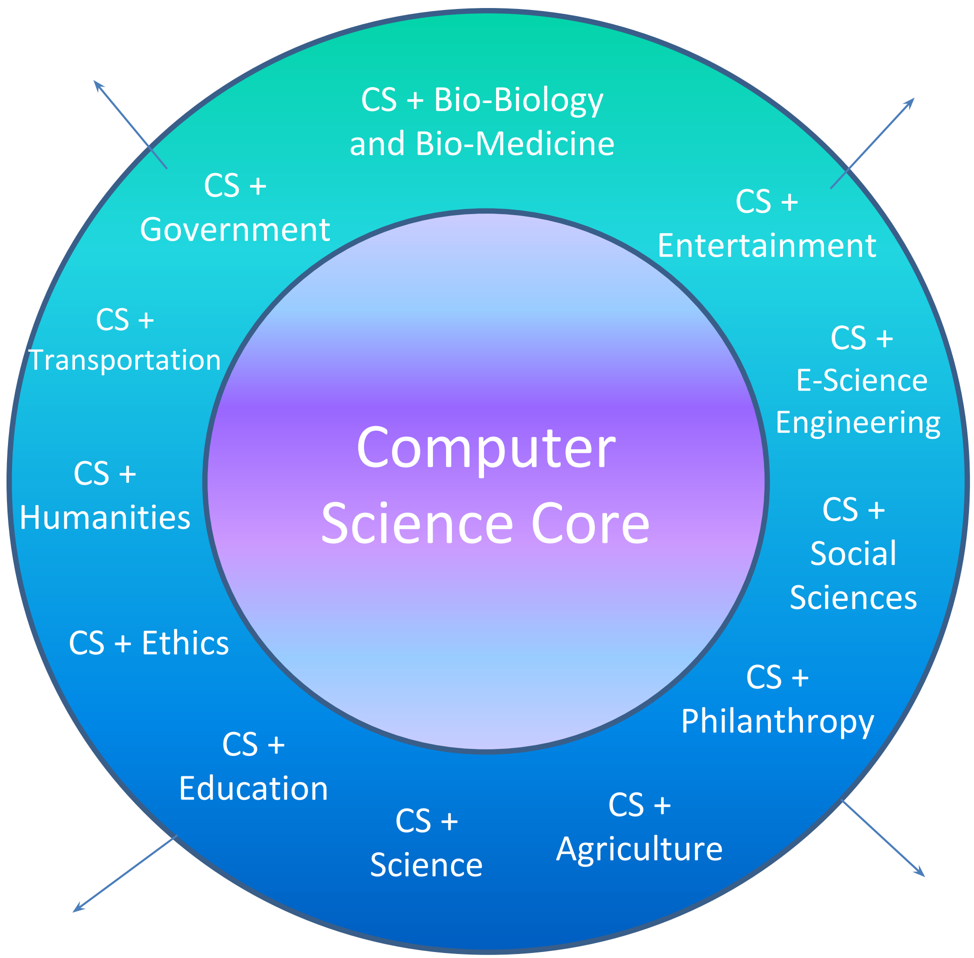
Note: Empiricism, Big Data, Data Science, are closely related terms.



Expanding Sphere of Computer Science

[1]from AZS Presentation to Harvard in 11/2004
See, UIUC,Caltech, Northwestern, for more examples

Outside the Core of Computer Science



[1] Of course, some of these were emerging even in ’76

The Challenge of Complexity

- As a field, we are building and operating our systems near the edge of feasibility
  - Large systems
  - And simple ones, alike.
- We haven’t really become an engineering discipline in the way I would like.
- I’ll illustrate two types of challenges:

I. Conundrum of Abstraction, Encapsulation, Reuse

- Abstraction, Encapsulation, Reuse is the core paradigm of our field...
- Approaches:
  - Libraries
  - Object-oriented computing
  - Services-oriented architectures
  - Open source
  - ...

Software Scale Challenges

- Supposedly, abstraction allows for multiple implementations, but this barely works…
  - Performance
  - Dependencies
  - Bugs
  - Other specification Incompleteness

- Software Engineering techniques problematic
  - Mono-repo in an open source world
  - Library maintenance
  - Libraries vs. services-oriented architectures
  - Code Review/Automated Testing/Continuous build/release
  - So many approaches…

Is this really an engineering *discipline*?

2. Insufficient Progress in Software Architecture

- We build things, with minor differences, repetitively
  - Costly
  - Distracting
  - Incompatible
- Our potential, user-programmer community:
  - Doesn't understand
  - Finds our systems hard to configure, manage, use
  - Fears risk of overly sophisticated technology
  - Therefore, often generate their own application-specific solutions
  - (so, we aren't gaining the leverage we should)

We Should and We Do Know a Lot, but…

- **...**
- **But,**
  - Why are the technical leaders so overwhelming key to major, new systems design and implementation?
  - Why does (or can) the industry develop many system with minimal advances, but significant incompatibilities?
  - That is, why are there so many different ways to do the same thing?

    Why do we regularly re-invent the wheel?
  - Why is it virtually impossible to get the details right?
  - Where is the regularity in system design?
  - Why are achieving reuse, failure analysis, security, etc. either so difficult or requiring of genius?

Over Time, Approaches Have Proliferated

- A vast number of approaches to building systems
  - 361 Apache projects as of Nov. 2018
- Many language possibilities

Just a Few of Our Tools at Two Sigma

Java, C++, Javascript, Python 2, Python 3, ...

Postgres

No-SQL

Distributed File Systems

Block-Storage

gRPC

REST

Map Reduce

Spark

Mesos

Micro-services

Kubernetes

Cassandra

Zookeeper

Many event managers...

- Given each system is probably Turing Complete
- Where is the scientific/engineering discipline?

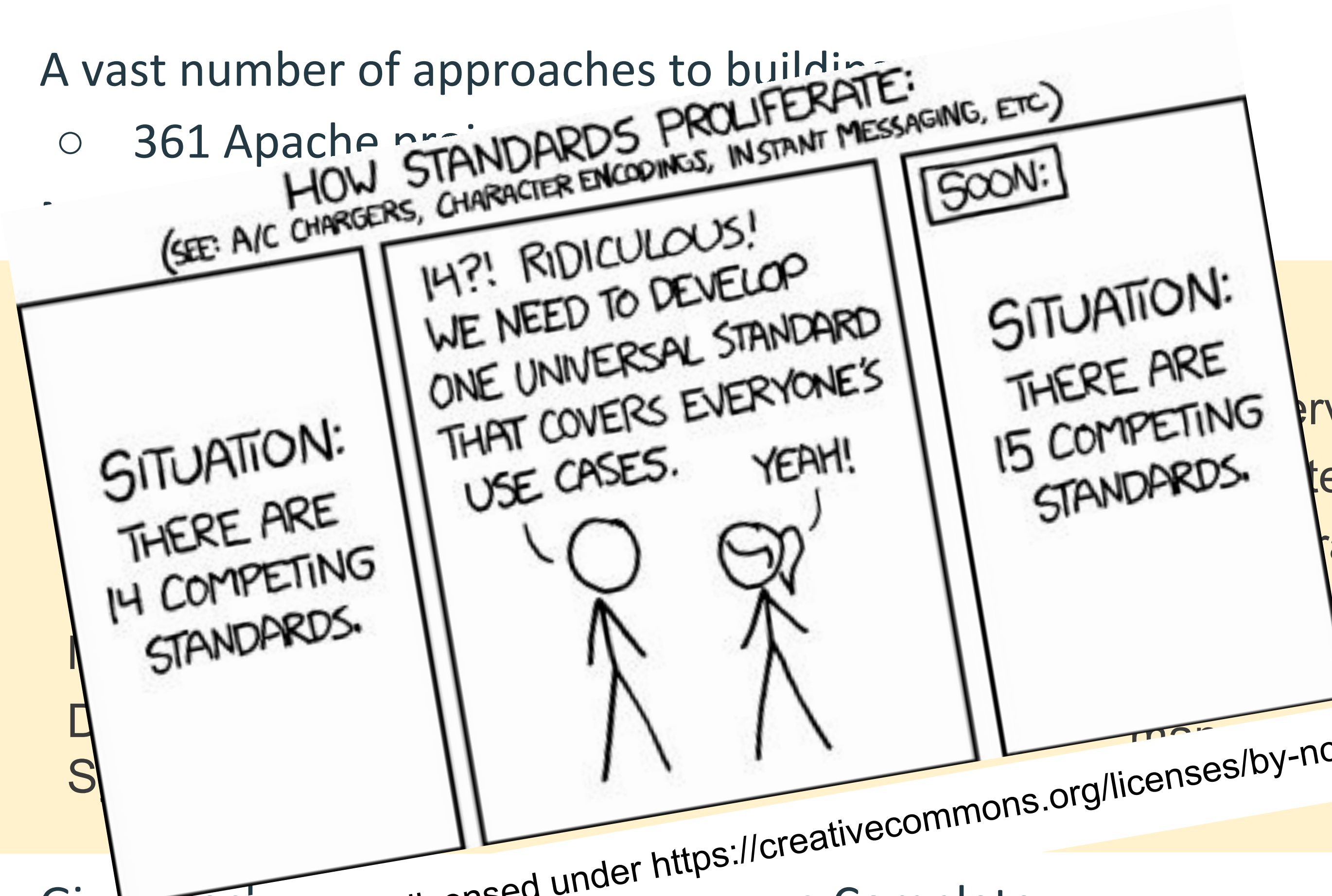
Over Time, Approaches Have Proliferated

- A vast number of approaches to buildin
  - 361 Apache pr
-

ervices
tes
ra
r
nt

Give , uring Complete

W e scientific/engineering discipline?



Reference, XKCD, licensed under https://creativecommons.org/licenses/by-nc/2.5/

Complexity, Technical Debt, ... is Serious

Example - Distributed Transactions

- Hypothesis: *Many believed distributed transactions are a fundamental building block that would greatly simplify the development of reliable distributed systems*
- Result:
  - Mostly used w/homogeneous databases or applications
  - Systems vendors may use them within their infrastructure
  - Marginal other user adoption as a core architectural enabler
- Why: - Many challenges
  - Difficult to implement: generality and depth (e.g., nesting, sagas, dependencies, …)
  - Difficult to install, configure, and manage
  - Difficult to standardize
  - Possibly, performance (?)
  - Possibly, window of vulnerability (?) ..

Example - Distributed Transactions

- Hypothesis: D*istributed transactions are a f* … *building block that would greatly simplify the de* … *le distributed systems*
- Result:
  - We do use them with …
  - Systems vendor … eir infrastructure
  - Marginal ot …
- Why: - Ma …
  - Di … erality and Depth
  - … manage
  - … rdize
  - … (?)
  - W … of Vulnerability (?

40+ years later, do we really know how important general purpose distributed transactions truly are?

Example - Distributed Transactions

- Hypothesis: D*istributed transactions are a f... building block that would greatly simplify the de... le distributed systems*
- Result:
  - We do use them with...
  - Systems vendors ... eir infrastructure
  - Marginal ot...
- Why: - Ma...
  - Dif... erality and Depth
  - ... manage
  - ... rdize
  - ... (?)
  - W... of Vulnerability (?

The same question could be said about many different abstractions and algorithms we've invented.

To the rescue, the Cloud Vendors (?)

- Hypothesis: Many of the challenges could be solved by the cloud vendors, if they focus on this.
  - Internal universality and standardization feasible
    - Services cheaper to deploy than software products
  - Economies of scale on many tasks
    - Integration
    - Configuration
    - Monitoring/management
    - …
  - Similar to the traditional Mainframe value proposition
    - The mainframe was far simpler so relatively feasible for customer management
    - New technology is more complex so better off as a service

What's needed by the Cloud Vendors

- A really coherent internal architecture and clear
  - Discipline
  - Not a potpourri
- Standardization of key elements
  - Kubernetes, Linux, … are a start but more
  - Recognition: many customers cannot avail themselves of core, sophisticated services if they are proprietary
- True robustness
  - Tools for security and privacy
    - E.g., enclaves
    - Fault tolerant of configuration errors
  - Appropriate techniques to inspire performance & availability

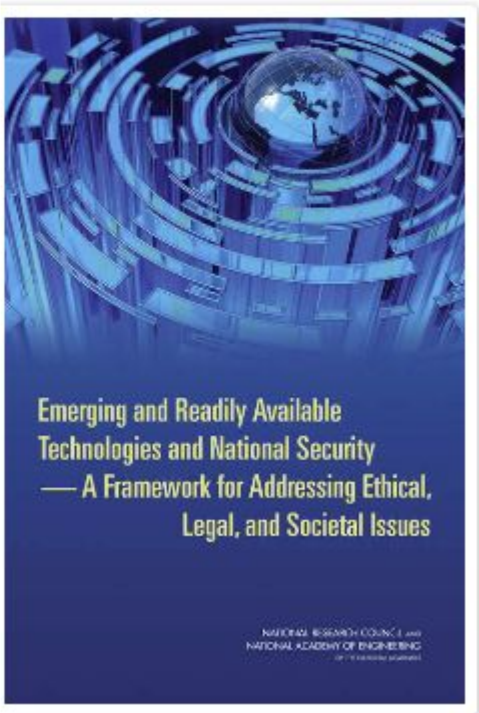
What's needed by the Cloud Vendors

- A really coherent internal architecture
  - Discipline
  - Not a potpourri
- Standardization of key eleme
  - Kubernetes, Linux, a
  - Many customer
    sophisticat
- True robus
  - To
  - guration errors
  - A es to inspire performance & availability

If the vendors do this, this would recreate similar structure of mainframe business of days yore, with the many pro's… and some con's.

Perspective on Ethics (1)

- Technology will advance
- As always, good and bad applications
  - E.g., the encryption debate
    - Search with Probable Cause vs.
    - Absolute privacy
  - Influence
    - Nudging for societal benefit
    - Collectivist tyranny
- It's very hard to comprehend unintended consequences in advance.
  - But, we need to educate ourselves
  - At least, to mitigate near term issues



Many lessons from a variety of domains, H. Lin, Ed..

Perspective on Ethics (2)

- Regulatory regimes must understand and come to grips w/**application** of technology
- Regulation needs to address problems today and not jump ahead to problems that don't or won't exist for a very long time.
  - Technical skill needed
  - Breadth of perspective is needed, as well
- Artificial General Intelligence is not likely "around the corner."
  - E.g., "Future of Work" is probably "work," for the foreseeable future
  - So, we need to solve real problems, and we must be precise in their identification and analysis

Perspective on Ethics (2)

- Regulatory regimes must understand and co
  w/**application** of technology
- Regulation needs to address proble o
  problems that don't or won't e
  - Technical skill needed
  - Breadth of persp
- Artificial Genera e corner."
  - E.g., "F for the foreseeable
    fu
  - s, and we must be precise in
    ysis

It is likely that the systems side of software development will become more regulated and held to higher standards: we should adapt now

Recap

Successes and the Expanding Sphere

-> Complexity (possible, and driven by applications)

-> The Cloud (should reduce complexity)

-> Societal Challenges (but society will care)

-> Likely increased regulatory burden (and enforce)

Thank you

These slides will available as:
http://www.azs-services.com
Also, see companion piece on data science:
https://azs-services.com/ opportunities-and-perils/

Questions?