What does it mean to have an engineering discipline for software?

How far has software engineering progressed toward that goal?

What are the next steps?

with examples from civil engineering and software architecture
What is “engineering”?

Definitions abound

They have in common:

Creating cost-effective solutions ...
... to practical problems ...
... by applying scientific knowledge ...
... building things ...
... in the service of mankind

Engineering enables ordinary people to do things that formerly required virtuosos
What is “engineering”? 

Definitions abound 

They have in common: 

- Creating cost-effective solutions ... 
- ... to practical problems ... 
- ... by applying **codified** knowledge ... 
- ... building things ... 
- ... in the service of mankind

*Engineering enables ordinary people to do things that formerly required virtuosos*
Characteristics of engineering

- limited time, knowledge, and resources force decisions on tradeoffs
- best-codified knowledge, preferentially science, shapes design decisions
- reference materials make knowledge and experience available
- analysis of design predicts properties of implementation
Engineering evolves from craft and commerce; it requires scientific foundations, or at least systematically codified knowledge.

Exploiting technology requires both management and a body of systematic, scientific knowledge.

Science often arises from progressive codification of practice.
Civil Engineering as Model
Civil Engineering

Example:
Bridges and Arches
Great Buildings of the World
Bridges, Derrick Beckett,
Hamlyn Publishing Group, Ltd.,
London, England, pp 10, 12, 16, 19
1st Century CE

Figure 4.4 Two Roman aqueducts, Anio Novus built on Claudia (From Curt Merckel, Die Ingenieurtechnik im Alterthum, 1899; courtesy Julius Springer-Verlag)
Craft of bridges

Romans

Empirical progress via failure and repair

Renaissance & Industrial Revolution

No deliberate application of mathematics to determine size or shape

Scientific Engineering

Little theory, but construction methods lasted until 19th century

Vitruvius: *De Architectura* [about 25 BC]
The Evolution of the Stone-arch Bridge

Ponte di Augusto, Rimini
Waterway below floor level 35%
Roman 14 A.D.

Pont Neuf (North Section), Paris
Waterway below floor level 50%
French 1678-1687

Pont Royal, Paris
Waterway below floor level 55%
French 1685-1687

Pont de la Concorde, Paris
Water below floor level 65%
French 1787-1791

Ring and Western Civilization, James Kip Finch,
Hill Book Company, Inc., New York, NY, 1951, p33
Fig. 28. Arch bridge, according to Leon Battista Alberti.
Ironbridge at Coalbrookdale, 1779
Dee Bridge disaster, 1847
Business of bridges

Romans

Increasingly long spans, lighter structures

Renaissance & Industrial Revolution

Rules of thumb about proportions

Explanation of structures:
- Brunelleschi on arches and domes 15th century
- Galileo on beams 17th century

Scientific Engineering

Introduction of cast iron, wrought iron, steel, and reinforced concrete
Hardest problem was identifying the proper basic concepts, e.g. force.

New mathematics was needed (calculus).
## Properties of Various Sections

<table>
<thead>
<tr>
<th>Sections</th>
<th>Area of Section $A$</th>
<th>Moment of Inertia $I$</th>
<th>Section Modulus $S = \frac{I}{y}$</th>
<th>Radius of Gyration $r = \sqrt{\frac{I}{A}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$b'd - ab$</td>
<td>$y = \frac{d}{2}$</td>
<td>$\frac{1}{12} (6d^2 - 6d^2)$</td>
<td>$\frac{\mu d - ab}{d}$</td>
<td>$\sqrt{\frac{\mu d - ab}{12 (6d - ab)}}$</td>
</tr>
<tr>
<td>$b'd - ab$</td>
<td>$y = \frac{d}{2}$</td>
<td>$\frac{1}{12} (3d^2 + 3d^2 - 6d^2)$</td>
<td>$\frac{\mu d - 3ab}{d}$</td>
<td>$\sqrt{\frac{\mu d - 3ab}{12 (6d - ab)}}$</td>
</tr>
<tr>
<td>$b'd - 2ab$</td>
<td>$y = \frac{d}{2}$</td>
<td>$\frac{1}{12} (3d^2 + 3d^2)$</td>
<td>$\frac{\mu d + 3ab}{d}$</td>
<td>$\sqrt{\frac{\mu d + 3ab}{12 (6d - ab)}}$</td>
</tr>
<tr>
<td>Box + Ar</td>
<td>$y = \frac{d}{2}$</td>
<td>$\frac{1}{12} (5d^2 + 3d^2 - 2a (d - 2a))$</td>
<td>$\frac{\mu d + 3ab}{d}$</td>
<td>$\sqrt{\frac{\mu d + 3ab}{12 (6d - ab)}}$</td>
</tr>
<tr>
<td>Box + Ar</td>
<td>$y = \frac{d}{2}$</td>
<td>$\frac{1}{12} (5d^2 + 3d^2)$</td>
<td>$\frac{\mu d + 3ab}{d}$</td>
<td>$\sqrt{\frac{\mu d + 3ab}{12 (6d - ab)}}$</td>
</tr>
</tbody>
</table>

### Notes:
- $\mu$ is the modulus of elasticity.
- $y$ is the distance from the neutral axis to the extreme fiber.
- $A$ is the area of the section.
- $I$ is the moment of inertia.
- $S$ is the section modulus.
- $r$ is the radius of gyration.
Figure 10.2 Types of arch bridge.

Figure 10.29 Coefficients for in-plane buckling of parabolic arch [59] $M_R = C_1 (E / L)^2$.
Engineering of bridges

Romans

Renaissance & Industrial Revolution

Scientific Engineering

1700: good theories (statics, strength of materials)

1750: tabulations of properties of materials

1850: formal analysis of a bridge structure

1900: structural analysis worked out

1950: systematic theory

2000: design automaton
21st century

PennDOT now requires use of its software for automated design of simple bridges
  o PennDOT’s Bridge Automated Design and Drafting Software (BRADD) automates bridge design from problem definition through CAD drawing.
  o BRADD designs concrete, steel, and concrete bridges with spans of 18 feet to 200 feet.
<table>
<thead>
<tr>
<th>Superstructure Type</th>
<th>Traditional</th>
<th>Integral</th>
<th>SuperOnly High/Stub/Wall</th>
<th>SuperOnly Integral</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prestressed Concrete Adjacent Box Beam</td>
<td>❒</td>
<td>❒</td>
<td></td>
<td>❒</td>
</tr>
<tr>
<td>Prestressed Concrete Spread Box Beam</td>
<td>❒</td>
<td>❒</td>
<td>❒</td>
<td>❒</td>
</tr>
<tr>
<td>Prestressed Concrete I-Beam</td>
<td>❒</td>
<td>❒</td>
<td>❒</td>
<td>❒</td>
</tr>
<tr>
<td>Steel Rolled Beam</td>
<td>❒</td>
<td>❒</td>
<td>❒</td>
<td>❒</td>
</tr>
<tr>
<td>Steel Plate Girder</td>
<td>❒</td>
<td>❒</td>
<td>❒</td>
<td>❒</td>
</tr>
</tbody>
</table>
Evolution of civil engineering

1700: statics
1700: strength of materials
1775: hydraulics

1750: properties of materials
1850: full analysis of a bridge

Romans: 1st century
Software Engineering
Software engineering as engineering

From the definition of engineering:

Creating cost-effective solutions ...
... to practical problems ...
... by applying codified knowledge ...
... building things ...
... in the service of mankind
Software engineering as engineering

From the definition of engineering:

The branch of computer science that ... 
... creates cost-effective solutions ... 
... to practical computing problems ... 
... by applying codified knowledge ... 
... developing software systems ... 
... in the service of mankind

Software is design-intensive -- manufacturing costs are minor

Software is symbolic, abstract, and constrained more by intellectual complexity than by fundamental physical laws
"Software Engineering"

Rallying Cry

Phrase coined in 1968 to draw attention to software problems

Aspiration, not description
Craft practice, 1968

- Monolithic development, merging research, development, production
- Software fine in many areas, but not for life-critical applications
- Widening gap between ambitions and achievement, increasing risk
- Software is late, over cost estimate, doesn’t meet specifications
- Too much revolution, not enough evolution
Figure 2. From Selig: Documentation for service and users. Originally due to Constantine.
Production techniques

Systematic **software development methods** bring order and predictability to projects via structure and project management (1970-1990s)

- Structured programming
- Waterfall models
- Incremental and iterative development
- Cost/schedule estimation
- Process maturity
- Extreme, agile processes
Codified knowledge

Data structures, algorithms
Programming languages and semantics
Verification and model checking
Computability and computational models
Objects and abstract data types
Canonical structures for many applications
Software architectures
Model-based engineering
Pattern languages

...
Fundamental ideas

**Abstraction** enables control of complexity

Imposing **structure** on problems makes them more tractable; **canonical solutions** are available

**Symbolic representations** are necessary and sufficient for solving information-based problems

Precise models support **analysis and prediction**

**Exponential growth** creates opportunities and limits
Commerce drives science

Science is often stimulated by problems in commercial practice

- safety-critical applications ➔ safety analysis
- large systems ➔ architectural patterns
- concurrency ➔ parallel logics and languages
- large state spaces ➔ model checking
- many versions ➔ program families, inheritance
- large-scale search ➔ MapReduce
- adaptive systems ➔ MAPE
Research and development stimulates creation of innovative ideas and industries.
Research and development stimulates creation of innovative ideas and industries.
Increasing Abstraction Scale
Design guidance

Choosing among algorithms based on the problem setting
Design guidance

Choosing among algorithms based on the problem setting

Use a bubble sort

Are there < 11 items or are you unwilling to write 20 lines instead of 10 lines of program to get an $n \log n$ sort time instead of $n^2$?

Yes

Are the items less than an average of $\log_2 n$ positions out of place?

Use Treesort 3 or a Shell sort

No

Are there storage available besides what the items currently occupy?

Is there storage available besides what the items currently occupy?

Yes

Use a distributive sort.

No

Try to use an address calculation sort

Are six or fewer tape drives and disks available?

Yes

Use a poly-phase merge.

No

Use an oscillating merge.

Are items to be added or deleted from the sorted order?

Is the CPU space-time product more important than I/O time?

Yes

Is the sort to be done on two disks?

Interleave the strings to be merged.

No

Use a tournament replacement sort.

Can 7 or more drives be read in both directions and quickly re-

No

Use a simple replacement sort or Heapsort for short records.
Software Architecture
Software architecture ...

- ... is principled understanding of the large-scale structure of software systems as collections of interacting elements
- ... emerged 1990s from informal roots
- ... codifies a vocabulary for software system structures based on types of components and connectors
- ... provides guidance for explicit design choices bridging requirements to code
with a program transformation
A layered system!!

http://www.multicians.org/architecture.html
Craft practice

Software has always had structure
  o Informal vocabulary
    - Objects, pipes/filters, interpreters, repositories ...
  o Intuitions and folklore about fitness to task

Ancient examples (since NATO69):
  o Software bundled with hardware
  o Compilers, layered operating systems
  o Databases for accounting
FIGURE 7. Flight Computer Operating System (The FCOS dispatcher coordinates and controls all work performed by the on-board computers.)

<table>
<thead>
<tr>
<th>Client Layer*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access domain management</td>
</tr>
<tr>
<td>Buffering and record-level I/O</td>
</tr>
<tr>
<td>Transaction coordination</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Agent Layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Implementation of standard server interface</td>
</tr>
<tr>
<td>Logger, agent, and instance tasks</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Helix Directories</th>
</tr>
</thead>
<tbody>
<tr>
<td>Path name to FID mapping</td>
</tr>
<tr>
<td>Single-file (database) update by one task</td>
</tr>
<tr>
<td>Procedural Interface for queries</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Object (FID directory)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification and capability access (via FIDs)</td>
</tr>
<tr>
<td>FID to tree-root mapping; table of (FID, root, ref_count)</td>
</tr>
<tr>
<td>Existence and deletion (reference counts)</td>
</tr>
<tr>
<td>Concurrency control (file interlocking)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Secure Tree</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic crash-resistant file structure</td>
</tr>
<tr>
<td>Conditional commit</td>
</tr>
<tr>
<td>Provision of secure array of blocks</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Commit and restart authority</td>
</tr>
<tr>
<td>Disk space allocation</td>
</tr>
<tr>
<td>Commit domains</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cache</th>
</tr>
</thead>
<tbody>
<tr>
<td>Caching and performance optimization</td>
</tr>
<tr>
<td>Commit support (flush)</td>
</tr>
<tr>
<td>Frame allocation (to domains)</td>
</tr>
<tr>
<td>Optional disk shadowing</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Canonical Disk</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physical disk access</td>
</tr>
</tbody>
</table>

*Also called client Helix.

Figure 2. Abstraction layering.

Figure 2. Display PostScript interpreter components.
A7E avionics architecture, as shown in Bachman et al
Software Documentation in Practice, SEI 2000
Commerce stimulates science

informal vocabulary ➔ styles/patterns
communication need ➔ notations, ADLs
ad hoc analysis ➔ style-specific analysis
multiple versions ➔ product lines
maintenance issues ➔
architecture as high-level documentation
Sample idioms / styles / patterns

- layers
  - virtual machines <hierarchy of abstractions>
  - client-server systems <decomposition of function>

- data flow
  - batch sequential <indep. programs, batch data>
  - pipes and filters <transducers, data streams>

- interacting processes
  - communicating processes <processes, messages>
  - event systems <processes, implicit invocation>
Explanations for practitioners

N-Tier architecture

Virtual machine

http://www.codeproject.com/Articles/430014/N-Tier-Architecture-and-Tips

http://www.pcmag.com/encyclopedia/term/53927/virtual-machine
Commercial practice

1970s: batch processing
   o modules and procedure calls, Cobol

1980s: informal “architecture” in papers
   o colloquial use of architectural terms

1990s: early structure
   o software product lines

2000s: architecture research enters practice
   o company-specific overall architectures
   o frameworks, UML
   o objects everywhere
Maturation of scientific ideas

Key Idea

Basic Research
Recognize problem, Invent ideas

Concept Formation
Refine ideas, publish solutions

Development & Extension
Try it out, clarify, refine

Internal Exploration
Stabilize, port, use for real problems

External Exploration
Broaden user group, extend

Popularization
Propagate through community

Seminal paper or system
Usable capability
Outsiders use it
Production quality, commercial support

15-20 years

Sam Redwine, Jr. and William Riddle: Software Technology Maturation, Proc ICSE-8, May 1985
Maturation of software architecture

Garlan and Shaw. Software architecture: reflections on an evolving discipline. ESEC/FSE keynote 2011
Foundations

information hiding, abstract data types, objects
layered systems
influence of structure on properties
model-driven approaches
computational models and logics
system implementation languages
…etc …

These concepts evolved on their own 15- to 20-year cycles; related concepts continue to evolve.
Basic research, 1985-1993

Deliberately designed structures for specific problems
Product line architectures for domains such as avionics, oscilloscopes, missile control
Catalogs of common idioms and architectural styles to support a design vocabulary
Balanced emphasis on components and connectors

Concept formation 1992-1996

Elaboration of basic models

Architectural description languages
Early formalization plus language development
architectural analysis reconciling multiple views
Taxonomies of styles based on early narrative
catalogs, becoming systematized as patterns
Architectural views
Workshops, first books

Foundations

Basic Research

Unification and refinement; second generation concepts

Development

Interoperability and integration among ADLs (Acme interchange language)
Refined taxonomies
Institutions, conference tracks, conferences
Increased attention to architecture in design; informal use of styles/patterns to guide design

Some designs formally analyzed

HLA for distributed simulation

Connecting architecture decisions to quality attributes

Analysis and evaluation techniques, SAAM >> ATAM

Books on practice and on specific aspects

Explicit attention to architecture in design; architecture’s role in quality attributes
External exploration: 1998-present

- UML (for better or worse) as standard ADL
- RUP as industrialization of 4+1 views
- O-O frameworks (also for better or worse)
- Component-based software engineering
- Company-specific end-to-end architectures

Technologies useful beyond development group
Popularization: 2000-present

Production-quality, supported, commercialized marketed versions of the technology

Architectural patterns fueled by web
  e.g. n-tier client-server, agent-based, service-oriented
  with ecosystem of services, tools, platforms, training

Frameworks and platforms

Standards for interfaces and system families

Architect as senior technical leader

Courses and place in core curriculum

Conferences and organizations for practitioners

Production-quality, supported, commercialized marketed versions of the technology

Architectural patterns fueled by web
  e.g. n-tier client-server, agent-based, service-oriented
  with ecosystem of services, tools, platforms, training

Frameworks and platforms

Standards for interfaces and system families

Architect as senior technical leader

Courses and place in core curriculum

Conferences and organizations for practitioners
AN x64 PROCESSOR IS SCREAMING ALONG AT BILLIONS OF CYCLES PER SECOND TO RUN THE XNU KERNEL, WHICH IS FRANTICALLY WORKING THROUGH ALL THE POSIX-SPECIFIED ABSTRACTION TO CREATE THE DARWIN SYSTEM UNDERLYING OS X, WHICH IN TURN IS STRAINING ITSELF TO RUN FIREFOX AND ITS GECKO RENDERER, WHICH CREATES A FLASH OBJECT WHICH RENDERS DOZENS OF VIDEO FRAMES EVERY SECOND!

BECAUSE I WANTED TO SEE A CAT JUMP INTO A BOX AND FALL OVER.

I AM A GOD

http://xkcd.com/676/
Systematically Organized Knowledge

SEI Series organizes knowledge about architecture and its analysis
## Architectural styles and reasoning

<table>
<thead>
<tr>
<th>Style class</th>
<th>Characteristic</th>
<th>Reasoning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data flow</td>
<td>Styles dominated by motion of data through the system, no “upstream” content control by recipient</td>
<td>Functional composition, latency</td>
</tr>
<tr>
<td>Closed loop control</td>
<td>Styles that adjust performance to achieve target</td>
<td>Control theory</td>
</tr>
<tr>
<td>Call-and-return</td>
<td>Styles dominated by order of computation, usually with single thread of control</td>
<td>Hierarchy (local reasoning)</td>
</tr>
<tr>
<td>Interacting processes</td>
<td>Styles dominated by communication patterns among independent, usually concurrent, processes</td>
<td>Nondeterminism</td>
</tr>
<tr>
<td>Data sharing styles</td>
<td>Styles dominated by direct sharing of data among components</td>
<td>Representation</td>
</tr>
<tr>
<td>Data-centered repositories</td>
<td>Styles dominated by a complex central data store, manipulated by independent computations</td>
<td>ACID properties, transaction rates, data integrity</td>
</tr>
<tr>
<td>Hierarchical</td>
<td>Styles dominated by reduced coupling, with resulting partition of the system into subsystems with limited interaction</td>
<td>Levels of service</td>
</tr>
</tbody>
</table>

Rules of thumb on data flow

If your problem is decomposed into sequential stages, consider batch sequential or pipeline architectures. If each stage is incremental, so that later stages can begin before earlier stages finish, consider a pipeline architecture. But avoid if there is a lot of concurrent access to shared data.

If your problem involves transformations on continuous streams of data (or on very long streams), consider a pipeline architecture. However, if your problem involves passing rich data representations, avoid pipelines restricted to ASCII.

If your system involves controlling continuing action, is embedded in a physical system, and is subject to unpredictable external perturbation so that preset algorithms go awry, consider closed loop architectures.

Generality-power trades

Styles, Platforms, and Product Lines

- Bosch Engine Control
- Siemens Healthcare for 3D

Product Lines
- AUTOSAR
- HLA
- IOS

Domain-Specific
Component Integration Platforms

Generic Component Integration Platforms

- CORBA
- COM
- JavaBeans
- Android
- ...

Specializations

- Generic Styles
- Pipes & Filters
- Process Control
- Data Flow
- Call-Return
- Events
- ...

Low

Power

Low

High

Specialization

Status of results

- Not all results are established scientific truths
  - observations and generalizations can be useful
- Brooks proposes recognizing three kinds of results, with individual criteria for quality:
  - **findings** -- well-established scientific truths -- judged by **truthfulness** and **rigor**
  - **observations** -- reports on actual phenomena -- judged by **interestingness**
  - **rules-of-thumb** -- generalizations, signed by an author but perhaps not fully supported by data) -- judged by **usefulness**
    with **freshness** as criterion for all
But is it “Engineering” yet?
But is it “Engineering” yet?

“Engineering” is associated with a level of assurance that protects the public health, safety, and welfare.

Consider, though . . .

- Toyota unexpected acceleration
- Retail data breaches (Home Depot, Staples, ...)
- HealthCare.gov rollout
- Sony cyberattack
- TurboTax vulnerability
- . . .

Illustrated London News, 1847
Toyota unintended acceleration

- **Throttle stuck open, driver couldn’t stop car**
  - Hundreds died/injured in 2002-2010 models
  - Toyota denied claims but settled for $1.6++ Billion

- **Electronic Throttle Control System (ETCS)**
  - wide open throttle $\rightarrow$ brakes won’t stop car
  - single-bit failure could kill critical subtask

- **Software didn’t follow known good practices**
  - watchdog didn’t detect major task failure
  - cyclomatic complexity often over 50
  - poor coding practice, $\sim$10,000 global variables
  - recursion could cause uncaught stack overflow
  - poor development/testing process compliance

# Identity Theft Resource Center

## 2014 Data Breach Category Summary

### How is this report produced? What are the rules? See last page of report for details.

### Report Date: 1/5/2015

### Page 1 of 1

<table>
<thead>
<tr>
<th>Totals for Category</th>
<th># of Breaches</th>
<th># of Records</th>
</tr>
</thead>
<tbody>
<tr>
<td>Banking/Credit/Financial</td>
<td>43</td>
<td>1,198,492</td>
</tr>
<tr>
<td>% of Breaches: 5.5%</td>
<td>% of Records: 1.4%</td>
<td></td>
</tr>
<tr>
<td>Business</td>
<td>258</td>
<td>68,237,914</td>
</tr>
<tr>
<td>% of Breaches: 33.0%</td>
<td>% of Records: 79.7%</td>
<td></td>
</tr>
<tr>
<td>Educational</td>
<td>57</td>
<td>1,247,812</td>
</tr>
<tr>
<td>% of Breaches: 7.3%</td>
<td>% of Records: 1.5%</td>
<td></td>
</tr>
<tr>
<td>Government/Military</td>
<td>92</td>
<td>6,649,319</td>
</tr>
<tr>
<td>% of Breaches: 11.7%</td>
<td>% of Records: 7.8%</td>
<td></td>
</tr>
<tr>
<td>Medical/Healthcare</td>
<td>333</td>
<td>8,277,991</td>
</tr>
<tr>
<td>% of Breaches: 42.5%</td>
<td>% of Records: 9.7%</td>
<td></td>
</tr>
</tbody>
</table>

### Totals for All Categories:

- # of Breaches: 783
- # of Records: 85,611,528
- % of Breaches: 100.0%
- % of Records: 100.0%

### 2014 Breaches Identified by the ITRC as of: 1/5/2015

- Total Breaches: 783
- Records Exposed: 85,611,528

http://www.idtheftcenter.org/
Ineffective Planning and Oversight Practices Underscore the Need for Improved Contract Management

GAO-14-694: Published: Jul 30, 2014.

Contract Planning and Oversight Practices Were Ineffective Given the Challenges and Risks

GAO-14-824T: Published: Jul 31, 2014.

Actions Needed to Address Weaknesses in Information Security and Privacy Controls

GAO-14-730: Published: Sep 16, 2014.

Information Security and Privacy Controls Should Be Enhanced to Address Weaknesses

GAO-14-871T: Published: Sep 18, 2014.

CMS Has Taken Steps to Address Problems, but Needs to Further Implement Systems Development Best Practices

Characteristics of engineering

- limited time, knowledge, and resources force decisions on tradeoffs
- best-codified knowledge, preferentially science, shapes design decisions
- reference materials make knowledge and experience available
- analysis of design predicts properties of implementation
software development methods

PRODUCTION

~ 1990, adoption of development methods

COMMERCIAL

Emerging, but spotty

SCIENCE

PROFESSIONAL ENGINEERING
Making Progress
Adapting to evolving technology

- Technology outruns traditional manuals
  - Understand how search supplants indexing
- Agility, “perpetual beta” vs overall design
  - Exploit power end of generality tradeoff, embedding knowledge in task-specific tools
- Scaling cost to consequence
  - High stakes applications have rigorous engineering, mashups are fine for throwaways – but where is middle ground?

*How do we bring codified knowledge to design?*  
*Exhortation won’t work*
Architectures at scale

- Highly distributed, dynamically-formed task-specific coalitions of distributed autonomous resources (fix “mashups”)
- Balance among privacy, data quality, data mining (address “credibility” of data)
- Pervasive cyber-physical systems: control, security, adaptation (“Internet of Things”)
- Socio-technical ecosystems: platforms, extensions, and people as part of system (“wicked” problems, end user development)
Civilize the electronic frontier

Infrastructure and amenities
Civil order, good manners, rule of law
Empowerment of citizens to manage their own affairs
Clarity on personal security/responsibility

This requires widespread understanding of the technology and shared expectations about its use
There are *lots* of casual developers

Estimated counts in American workplace

<table>
<thead>
<tr>
<th>Role</th>
<th>Count (in millions)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Professional, non-software, managerial</td>
<td>40.5M</td>
</tr>
<tr>
<td>Non-professional, non-managerial</td>
<td>47M</td>
</tr>
</tbody>
</table>

Education

- Self-taught: 41.8%
- BS in CS (or related): 37.7%
- On-the-job training: 36.7%
- MS in CS (or related): 18.4%
- Online class: 17.8%
- Some univ, no degree: 16.7%
- Industry certification: 6.1%
- Other: 4.3%
- Boot-camp: 3.5%
- PhD in CS (or related): 2.2%
- Mentorship program: 1.0%

“Professional and enthusiast programmers” (international)


# Demographics of US Internet users

<table>
<thead>
<tr>
<th>Overall</th>
<th>Total adults</th>
<th>87%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Women</td>
<td>87</td>
<td></td>
</tr>
<tr>
<td>Men</td>
<td>86</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Age</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>18-29</td>
<td>97%</td>
<td></td>
</tr>
<tr>
<td>30-49</td>
<td>93</td>
<td></td>
</tr>
<tr>
<td>50-64</td>
<td>88</td>
<td></td>
</tr>
<tr>
<td>65+</td>
<td>57</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Geography</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>urban</td>
<td>88%</td>
<td></td>
</tr>
<tr>
<td>suburban</td>
<td>87</td>
<td></td>
</tr>
<tr>
<td>rural</td>
<td>83</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Education</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;= high school</td>
<td>76%</td>
<td></td>
</tr>
<tr>
<td>some college</td>
<td>91</td>
<td></td>
</tr>
<tr>
<td>college +</td>
<td>97</td>
<td></td>
</tr>
</tbody>
</table>

### Generations Online 2010

This chart shows the popularity of internet activities among internet users in each generation.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Search</td>
<td>Search</td>
<td>Search</td>
<td>Search</td>
<td>Search</td>
<td>Search</td>
<td>Search</td>
</tr>
<tr>
<td>Health info</td>
<td>Health info</td>
<td>Health info</td>
<td>Health info</td>
<td>Health info</td>
<td>Health info</td>
<td>Health info</td>
</tr>
<tr>
<td>Social network sites</td>
<td>Get news</td>
<td>Get news</td>
<td>Get news</td>
<td>Get news</td>
<td>Get news</td>
<td>Get news</td>
</tr>
<tr>
<td>Watch video</td>
<td>Govt website</td>
<td>Govt website</td>
<td>Govt website</td>
<td>Govt website</td>
<td>Govt website</td>
<td>Govt website</td>
</tr>
<tr>
<td>Get news</td>
<td>Travel reservations</td>
<td>Travel reservations</td>
<td>Travel reservations</td>
<td>Travel reservations</td>
<td>Travel reservations</td>
<td>Travel reservations</td>
</tr>
<tr>
<td>Buy a product</td>
<td>Watch video</td>
<td>Buy a product</td>
<td>Bank online</td>
<td>Watch video</td>
<td>Bank online</td>
<td>Bank online</td>
</tr>
<tr>
<td>IM</td>
<td>Bank online</td>
<td>Watch video</td>
<td>Financial info</td>
<td>Financial info</td>
<td>Financial info</td>
<td>Financial info</td>
</tr>
<tr>
<td>Listen to music</td>
<td>Social network sites</td>
<td>Bank online</td>
<td>Social network sites</td>
<td>Social network sites</td>
<td>Social network sites</td>
<td>Social network sites</td>
</tr>
<tr>
<td>Travel reservations</td>
<td>Bank online</td>
<td>Watch video</td>
<td>Religious info</td>
<td>Religious info</td>
<td>Religious info</td>
<td>Religious info</td>
</tr>
<tr>
<td>Online classifieds</td>
<td>Online classifieds</td>
<td>Online classifieds</td>
<td>Online classifieds</td>
<td>Online classifieds</td>
<td>Online classifieds</td>
<td>Online classifieds</td>
</tr>
<tr>
<td>Bank online</td>
<td>Listen to music</td>
<td>Financial info</td>
<td>Financial info</td>
<td>Financial info</td>
<td>Financial info</td>
<td>Financial info</td>
</tr>
<tr>
<td>Govt website</td>
<td>IM</td>
<td>IM</td>
<td>IM</td>
<td>IM</td>
<td>IM</td>
<td>IM</td>
</tr>
<tr>
<td>Play games</td>
<td>IM</td>
<td>IM</td>
<td>IM</td>
<td>IM</td>
<td>IM</td>
<td>IM</td>
</tr>
<tr>
<td>Read blogs</td>
<td>Financial info</td>
<td>Religious info</td>
<td>Religious info</td>
<td>Religious info</td>
<td>Religious info</td>
<td>Religious info</td>
</tr>
<tr>
<td>Financial info</td>
<td>Rate things</td>
<td>IM</td>
<td>IM</td>
<td>IM</td>
<td>IM</td>
<td>IM</td>
</tr>
<tr>
<td>Rate things</td>
<td>Read blogs</td>
<td>Play games</td>
<td>IM</td>
<td>IM</td>
<td>IM</td>
<td>IM</td>
</tr>
<tr>
<td>Religious info</td>
<td>Play games</td>
<td>Read blogs</td>
<td>IM</td>
<td>IM</td>
<td>IM</td>
<td>IM</td>
</tr>
<tr>
<td>Online auctions</td>
<td>Online auction</td>
<td>Online auction</td>
<td>Online auction</td>
<td>Online auction</td>
<td>Online auction</td>
<td>Online auction</td>
</tr>
<tr>
<td>Podcasts</td>
<td>Online auction</td>
<td>Online auction</td>
<td>Online auction</td>
<td>Online auction</td>
<td>Online auction</td>
<td>Online auction</td>
</tr>
<tr>
<td>Donate to charity</td>
<td>Online auction</td>
<td>Online auction</td>
<td>Online auction</td>
<td>Online auction</td>
<td>Online auction</td>
<td>Online auction</td>
</tr>
<tr>
<td>Virtual worlds</td>
<td>Virtual worlds</td>
<td>Virtual worlds</td>
<td>Virtual worlds</td>
<td>Virtual worlds</td>
<td>Virtual worlds</td>
<td>Virtual worlds</td>
</tr>
</tbody>
</table>

**Source:** Pew Internet surveys.

http://www.pewinternet.org/2010/12/16/generations-2010/

**Key:** % of internet users in each generation who engage in this online activity
<table>
<thead>
<tr>
<th>Email</th>
<th>Search</th>
<th>Email</th>
<th>Search</th>
<th>Email</th>
<th>Search</th>
<th>Email</th>
<th>Search</th>
</tr>
</thead>
<tbody>
<tr>
<td>Health info</td>
<td>Health info</td>
<td>Health info</td>
<td>Health info</td>
<td>Health info</td>
<td>Health info</td>
<td>Health info</td>
<td>Health info</td>
</tr>
<tr>
<td><strong>social network sites</strong></td>
<td>Get news</td>
<td>Get news</td>
<td>Get news</td>
<td>Get news</td>
<td>Get news</td>
<td>Get news</td>
<td>Get news</td>
</tr>
<tr>
<td><strong>Watch video</strong></td>
<td>Govt website</td>
<td>Govt website</td>
<td>Govt website</td>
<td>Govt website</td>
<td>Govt website</td>
<td>Govt website</td>
<td>Govt website</td>
</tr>
<tr>
<td>Get news</td>
<td>Travel reservations</td>
<td>Travel reservations</td>
<td>Travel reservations</td>
<td>Travel reservations</td>
<td>Travel reservations</td>
<td>Travel reservations</td>
<td>Travel reservations</td>
</tr>
<tr>
<td>Buy a product</td>
<td>Watch video</td>
<td>Buy a product</td>
<td>Travel reservations</td>
<td>Buy a product</td>
<td>Travel reservations</td>
<td>Buy a product</td>
<td>Travel reservations</td>
</tr>
<tr>
<td>IM</td>
<td>Listen to music</td>
<td>IM</td>
<td>Bank online</td>
<td>IM</td>
<td>Bank online</td>
<td>IM</td>
<td>Bank online</td>
</tr>
<tr>
<td><strong>Social network sites</strong></td>
<td>Bank online</td>
<td><strong>Social network sites</strong></td>
<td>Bank online</td>
<td><strong>Social network sites</strong></td>
<td>Bank online</td>
<td><strong>Social network sites</strong></td>
<td>Bank online</td>
</tr>
<tr>
<td>Online classifieds</td>
<td>Online classifieds</td>
<td>Online classifieds</td>
<td>Online classifieds</td>
<td>Online classifieds</td>
<td>Online classifieds</td>
<td>Online classifieds</td>
<td>Online classifieds</td>
</tr>
<tr>
<td>Bank online</td>
<td>Listen to music</td>
<td>Bank online</td>
<td>Financial info</td>
<td>Bank online</td>
<td>Financial info</td>
<td>Bank online</td>
<td>Financial info</td>
</tr>
<tr>
<td>Govt website</td>
<td>IM</td>
<td>Govt website</td>
<td>Financial info</td>
<td>Govt website</td>
<td>Financial info</td>
<td>Govt website</td>
<td>Financial info</td>
</tr>
<tr>
<td><strong>Play games</strong></td>
<td>IM</td>
<td><strong>social network sites</strong></td>
<td>IM</td>
<td><strong>social network sites</strong></td>
<td>IM</td>
<td><strong>social network sites</strong></td>
<td>IM</td>
</tr>
<tr>
<td>Read blogs</td>
<td>Financial info</td>
<td>Religious info</td>
<td>Religious info</td>
<td>Religious info</td>
<td>Religious info</td>
<td>Religious info</td>
<td>Religious info</td>
</tr>
<tr>
<td>Financial info</td>
<td>Religious info</td>
<td>Rate things</td>
<td>IM</td>
<td>Religious info</td>
<td>Rate things</td>
<td>IM</td>
<td>Religious info</td>
</tr>
<tr>
<td><strong>social network sites</strong></td>
<td>IM</td>
<td><strong>social network sites</strong></td>
<td>IM</td>
<td><strong>social network sites</strong></td>
<td>IM</td>
<td><strong>social network sites</strong></td>
<td>IM</td>
</tr>
<tr>
<td><strong>social network sites</strong></td>
<td>IM</td>
<td><strong>social network sites</strong></td>
<td>IM</td>
<td><strong>social network sites</strong></td>
<td>IM</td>
<td><strong>social network sites</strong></td>
<td>IM</td>
</tr>
</tbody>
</table>
Civilizing the electronic frontier

- **Policy, requiring technology**
  - Balance anonymity and accountability
  - Balance security and privacy
  - Balance individual and corporate objectives
  - Address product liability

- **Technology**
  - Apply known best practices and designs
  - Address new forms of information access (search) and software creation (independent parts)

- **User models**
  - Improve the explanations and intuitions we provide the public at large
Recapitulation

Engineering evolves from craft and commercial practice via science

Ideas evolve over time from pure research to practical production

The greatest need for engineering is in the most critical applications