Title:

Characterizing Employer's Expectations of the Communication Abilities of New Engineering Graduates

Authors:

Susan Ruff (MIT) and Michael Carter (NCSU)

Abstract:

One of the perennial challenges of writing in the disciplines is how to prepare students to be effective communicators in the professions they will be entering. Communication teachers working in the disciplines are often not aware of what is expected of recent graduates by their employers. To better understand the gap between recent graduates' communication abilities and employers' expectations, the authors surveyed software engineering professionals. They asked which of 67 communication abilities are unimportant for software engineers, which ones are learned on the job, which ones recent graduates are expected to have but lack, and which ones recent graduates possess.

Results showed that employers expect graduates to communicate clearly and professionally, while specific audiences or forms of communication may be learned on the job. Recent graduates meet many of employers' expectations but lack others. For example, most are reported to use English fluently and to use terminology correctly but to lack concision and cohesion. Employers disagree about whether graduates' communication is sufficiently professional.

These results raise interesting questions about the boundaries of communication pedagogy. For example, employers seem to attribute value to politeness in communication; should communication educators attempt to teach students to be nice? Employers also attribute particular value to oral communication; should we decrease emphasis on written communication? We believe these results can inform, but should not dictate, communication pedagogy.

Characterizing Employer's Expectations
of the
Communication Abilities
of
New Engineering Graduates

Susan Ruff, MIT, and Michael Carter, NCSU
IWAC 2016

The full article is available at
http://web.mit.edu/ruff/www/Ruff_Carter.pdf

Special Focus Issue:
Integrating Communication Instruction Throughout STEM Curricula

2008 Chautauqua
Teaching Communication Skills
in the
Software Engineering
Curriculum

Paul Anderson, et al.

MIAMI UNIVERSITY

Thank you to Paul Anderson, et al., for organizing the NSF sponsored Chautauqua at which our research collaboration began.
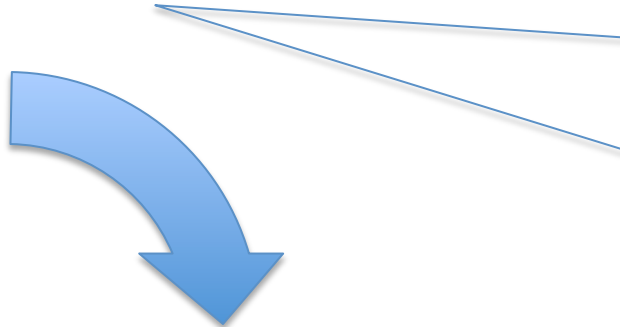
Interviews and focus groups provided a rich picture of engineering communication, but this isn't sufficient to inform teaching since we may not need to teach abilities learned on the job or elsewhere.

38 abilities important for software engineers, e.g., the ability to discern when to ask a question rather than assert an opinion and when to remain silent rather than to speak

TABLE I COMMUNICATION OUTCOMES FROM SOFTWARE ENGINEERING PROFESSIONALS

Software engineers should be able to:

Design communication
Evaluate communication situations and design communication appropriately for different purposes and contexts.
Frame communication in terms of the knowledge & concerns of the audience. Communicate effectively to a variety of audiences, (e.g. managers, peers, across organizational boundaries, customers, & end users).
Recognize the different communication cultures and norms of different countries, organizations, areas within organizations, ethnic groups, and individuals, and adapt to those differences.
Prioritize communication tasks to use time wisely.
Discern when it is more appropriate/effective to keep silent rather than to speak and to ask questions rather than to assert an opinion.

Explain clearly
Present information in a way that goes beyond the specific details of a project to provide the big picture, a higher level of summary.
Explain code, methods, and design decisions by communicating the intent—what was meant to be achieved—and reasons—why key choices were made.
Achieve an appropriate balance between conciseness and explanation; go directly to the point.
Answer questions clearly by going beyond what the questioner has explicitly asked; anticipate what else the questioner might need to know.
Communicate effectively under stress.
Communicate convincingly.
Use consistent and appropriate terminology.

Discuss productively
Lead a productive group discussion.
Deal constructively with conflict: debate/discuss/negotiate/collaborate productively and respectfully.
Support the transition from debate to the formation of a decision; e.g. summarize issues, propose solutions, &/or back down, as needed.
Hear criticism as a constructive contribution to the outcome of a project (without getting defensive).
Give criticism constructively and respectfully.
Collaborate with others within an integrated project team or from different areas of the organization. Demonstrate an understanding of how software engineering decisions affect others by communicating across organizational boundaries to inform, solicit input, and identify win-win solutions.

Receive communication
Solicit help, advice, or information.
Listen actively; ask clarifying questions.
Read with comprehension and evaluate information to determine what is credible and relevant.
Adjust communication based on (non-verbal) reactions of the audience; solicit feedback about the effectiveness of the communication.
Learn & improve communication skills, especially interpersonal skills.

Communicate professionally
Give opinions with a balance of confidence & humility.
Avoid complaining, by proposing a solution, fixing the problem, or remaining silent.
Be nice to others, through words and tone.
Manage non-verbal communication to avoid sending inappropriate messages.
Make own accomplishments known without arrogance.
Communicate charismatically; be passionate/animated in order to influence people.
Mentor others and help them grow.
Communicate through transparency (make information openly available).
Develop the flexibility to communicate in different roles within an organization.
Inform managers and team members of potential problems before the problems become serious.
Participate in meetings.

Use common forms & tools
Demonstrate a mastery of the kinds of formal and informal communication most often used in the industry (e.g., email, bug reports, meetings, presentations to groups, one-on-one, teleconferences, IM, code comments, documentation, requirements, status reports).
Use digital tools that are beneficial for communication and teamwork (e.g., tools for document control, bitmap and vector illustrations, documentation, web pages, basic video/audio for presentations, intuitive GUI design, and project planning).
Give effective and engaging presentations.
Use email appropriately, demonstrating an understanding of what information should be included and what should not, of when to use "reply all," and of the necessity to read carefully before sending.

978-1-4244-4714-5/09/$25.00 ©2009 IEEE   October 18 - 21, 2009, San Antonio, TX
39th ASEE/IEEE Frontiers in Education Conference

Research Questions

Which communication abilities do employers *expect recent graduates* to have?

Do recent graduates meet expectations?

Which abilities do employers *not* expect?

Why not?
Is the ability *unimportant* or *learned on the job*?

We surveyed software engineers responsible for hiring & review
    = "evaluator" (usually practicing software engineer; perhaps manager)

Do you expect recent graduates to be able to…
    …give clear high-level overviews?

☐ **Yes**, but they usually **lack** this ability.

☐ **Yes**, and they usually **possess** this ability.

☐ **No**, this ability is **unimportant** for software engineers.

☐ **No**, this ability will be **learned on the job**.

The online survey was advertised in at least 7 states

To

- Participants in our prior research on software-engineering communication
- Participants in the 2008 Chautauqua in Teaching Communication Skills in the Software Engineering Curriculum
- Frontiers in Education conference (attendees and exhibitors)
- Employers at 2 career fairs at MIT
- BRAWN, a Boston-area technical communication mailing list

Help educators produce software engineers with **excellent communication skills**.

20 minute survey:
web.mit.edu/ruff/SEcomm

$150 drawing for taking the survey
$150 drawing for forwarding the link

Forwarding and participation were encouraged by two $150 drawings

The sample is not random, so demographics matter.

Results may best represent information industries along the coasts

Results — Do you expect recent graduates to be able to…

…give clear high-level overviews?

Results: **Yes** but they usually **lack** this ability

Percent of evaluators who expect the ability (y-axis, 70 to 100)
% who say grads possess it (x-axis, 0 to 30)

Data points: 1, 2, 3, 4, 5 with region A marked.

Employers expect the abilities in Region A, but graduates are not meeting expectations. We might consider increasing emphasis within curriculum.

Abilities 1 & 2 are meant to capture notions of cohesion and coherence.

1. Connect new information to information that is familiar to the audience.
2. Order information in a way that makes explanations easy to follow.
3. Recognize one's own communication weaknesses and improve.
4. Be concise.
5. Communicate via code comments.

Code comments are a genre specific to software engineering.

Results: **Yes**, and they usually **possess** this ability

Employers expect the abilities in Region C, and graduates are meeting expectations. Perhaps keep curriculum as is.

Percent of evaluators who expect the ability

% who say grads possess it
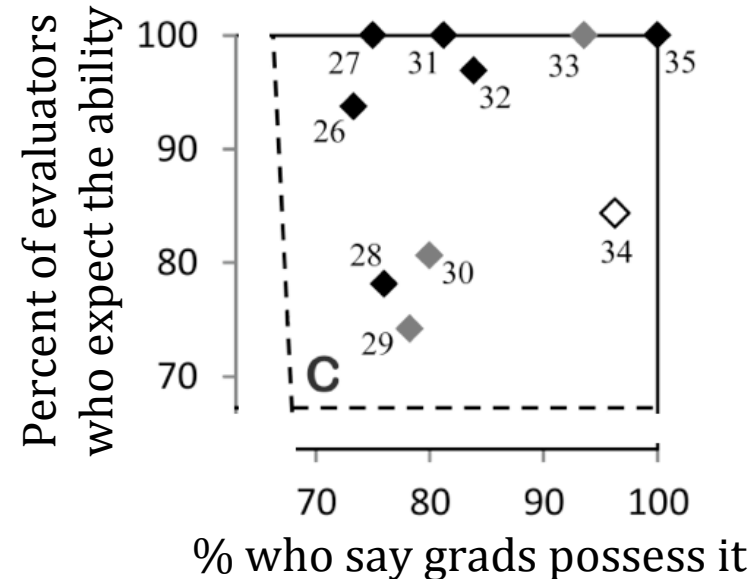
26. Communicate ideas one-on-one

27. During discussion, treat others with respect

28. Communicate to an audience of other software engineers

29. Communicate via small talk / social conversation

30. Communicate via telephone

31. Be nice to others, though words and tone

32. Use correct and consistent terminology

33. Use English fluently

34. Communicate via instant messaging

35. Communicate via e-mail

Results: **Yes**, but they **may lack or possess** this ability

Employers expect the abilities in Region B, but disagree as to whether graduates meet expectations. Perhaps provide individualized teaching for those students who need it, if possible.
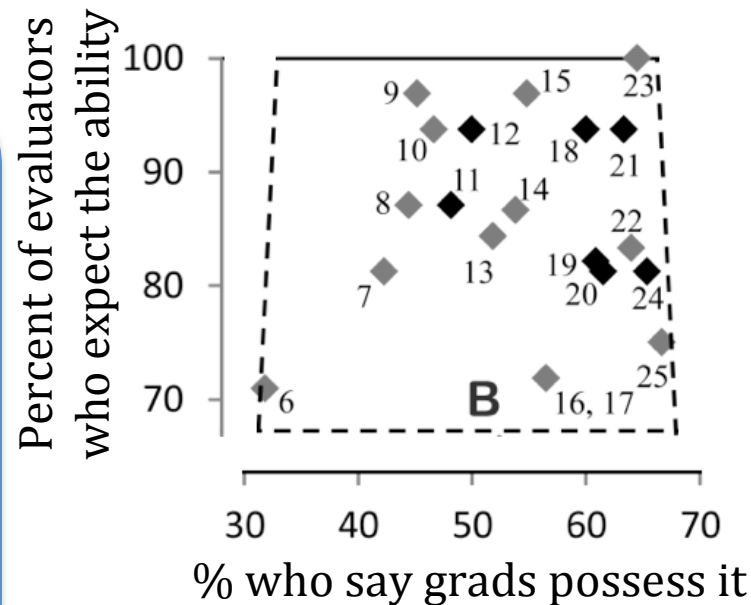
These are abilities for communicating professionally—of being *nice*. Should we be in the business of teaching students to be nice? If we decide yes, *how* can we do so?

For example,
   7. Adjust communication based on non-verbal reactions
   8. Discern when to ask questions rather than to assert an opinion
   9. Communicate with a balance of confidence and humility
   12. Listen actively
   13. Avoid taking debate, feedback, or others' opinions personally
   14. Discern when to keep silent rather than to speak
   15. Avoid complaining
   20. Respond professionally to one's own mistakes

Results: Abilities **not expected** of recent graduates

Learned on the job:

- Experience with document management systems
- Communicate via online meetings
- Experience with tools for project planning
- Flexibility to communicate in different roles within an organization
- Be aware of the knowledge and concerns of customers of the company
- Be aware of the knowledge and concerns of business &/or marketing

Employers do *not* expect recent graduates to have these abilities because they're learned on the job. That's not surprising, since many of these abilities are specific to the job or company. We might choose to teach some in order to graduate particularly competitive engineers, but if time is tight, these abilities could perhaps be deemphasized or omitted from the curriculum.

Results: Abilities **not expected** of recent graduates

Learned on the job:
- Experience with document management systems
- Communicate via online meetings
- Experience with tools for project planning
- Flexibility to communicate in different roles within an organization
- Be aware of the knowledge and concerns of customers of the company
- Be aware of the knowledge and concerns of business &/or marketing

Unimportant:
None!

> This isn't surprising because most of the abilities in the survey had been previously identified as important for software engineers.
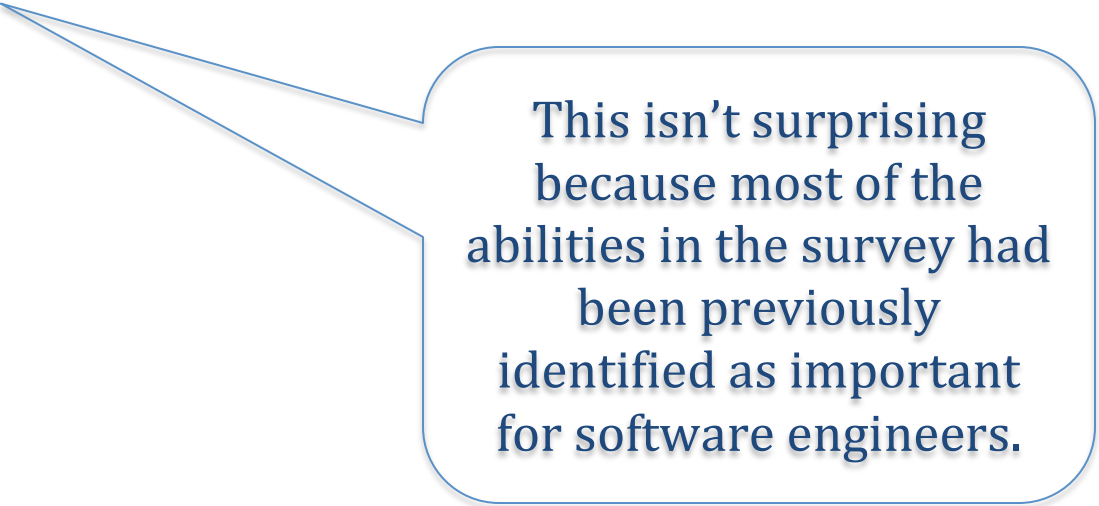
Results: Abilities **not expected** of recent graduates

Learned on the job:
- Experience with document management systems
- Communicate via online meetings
- Experience with tools for project planning
- Flexibility to communicate in different roles within an organization
- Be aware of the knowledge and concerns of customers of the company
- Be aware of the knowledge and concerns of business &/or marketing

Unimportant:
  None!

Relatively unimportant:
- Communicate via conference posters
- Communicate via journal articles
- Be aware of the knowledge and concerns of lawyers

Employers do *not* expect these abilities, but they disagree as to whether the abilities are unimportant or learned on the job.

Academic genres are relatively unimportant for engineers

Summary of Results
(Software engineering, especially in information industries along coasts)

Employers don't expect
- o academic genres, e.g.,
  - ▪ journal article, conference poster
- o company-specific knowledge, e.g.,
  - ▪ document-management systems or project-planning tools
  - ▪ concerns of customers or business/marketing

Employers expect clarity and professionalism.
- o Clarity expectations are met in some ways, e.g.,
  - • using terminology correctly
- but not others, e.g.,
  - • communicating concisely & cohesively
- o Employers disagree whether grads communicate professionally.

These results should not dictate curriculum, but can be used as a starting point for discussion of which aspects of communication to prioritize.

Implications vary by institution and individual

Thank you!

The remaining slides are backup:
- The meaning of the colors of the data points
  [i.e., the second half of the data from the split survey]
- The statistical analyses
- The graph for the abilities employers *don't expect*
- A cross sectional look at the data that reveals abilities to teach students who would like to be particularly competitive graduates.
- A resource for teaching professional communication
- Implications re: oral vs. written communication
- A few motivational quotations

We also surveyed software engineering "practitioners"
"practitioner" = software engineer not involved in hiring or review

How important is each ability for software engineers in your workplace?

(0=Not important)

1=Somewhat   5=Essential

	0	1	2	3	4	5	Don't know
Be nice to others, through words and tone (e.g., in daily interactions)	○	○	○	○	○	○	○
During discussion, treat others with respect (e.g., when giving an opinion, debating potential solutions, reviewing code)	○	○	○	○	○	○	○
Avoid taking debate, feedback, or others' opinions personally	○	○	○	○	○	○	○

Evaluators & practitioners agree on what's important
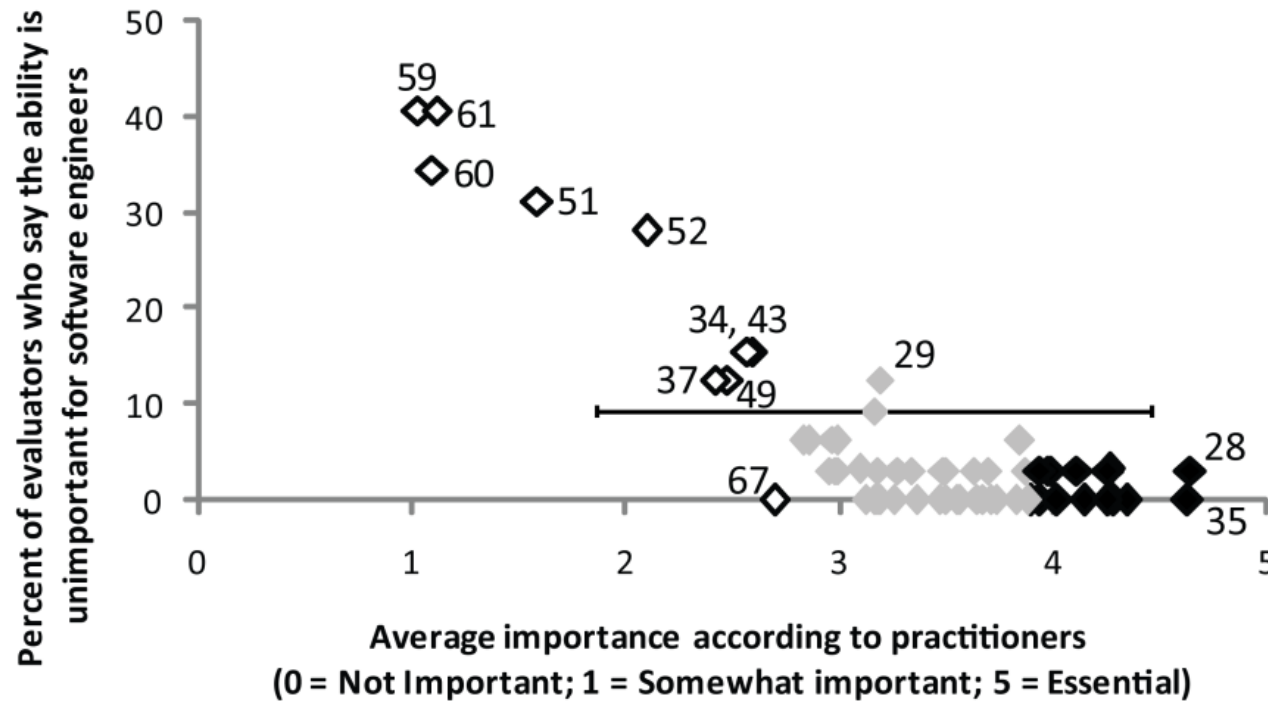
Strong correlation between evaluators and practitioners: $R^2 = 0.69$.
Agreement among practitioners is weak: error bar $\sigma \approx 1.3$. (0.7-1.9)

Average importance of ability, per practioners:
- ◆ most important   (average importance > 3.9)
- ◆ somewhat important   (between 2.8 and 3.9)
- ◇ least important   (< 2.8).

Statistical analyses

67 abilities

Evaluators: N=32; 4 categories + Don't Know

0-5 "Don't Know" responses for each ability (mean=0.5)

2-tailed exact binomial tests with p=0.05

First round N=27-32 (mean=31.5)

Second round N = 20-32 (mean=26.5)

Total expected Type I errors: 5.55

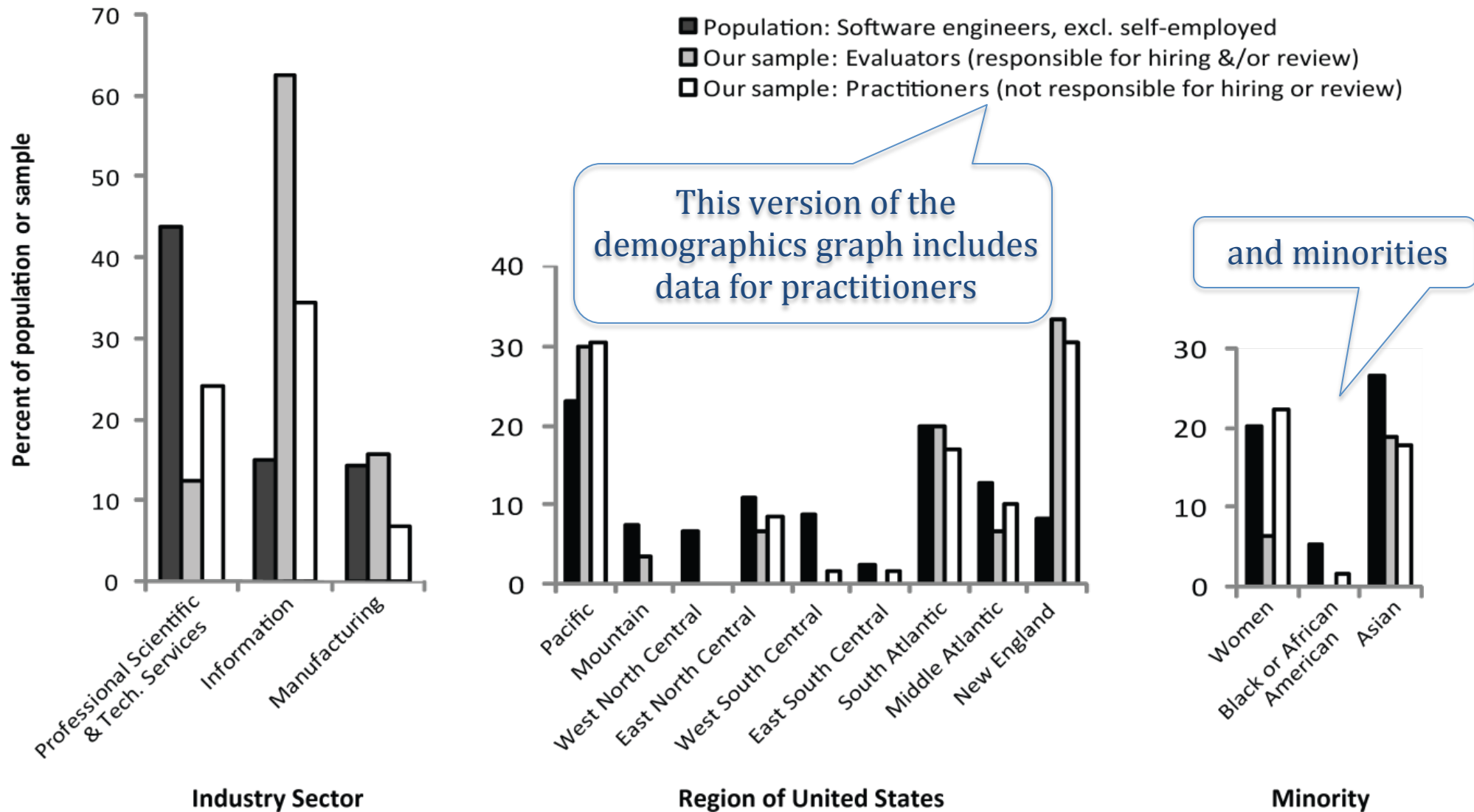Practitioners: N=64; 6-item Likert scale

0=not important

1=somewhat important

5=extremely important

standard deviation typically about 1.3

least/most important ≥ 2 standard deviations from mean (of means)

The sample is not random, so demographics matter.

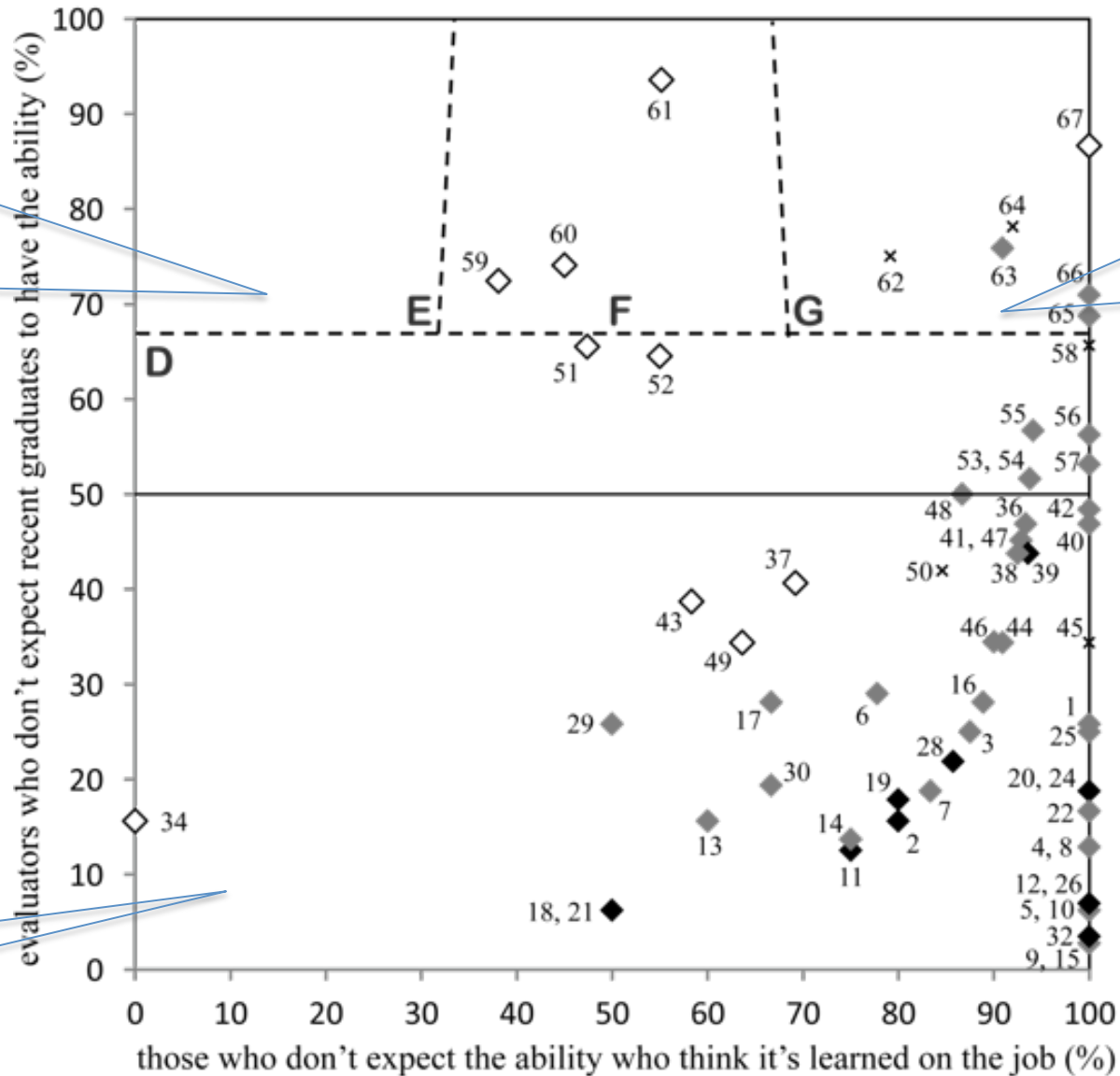Results may best represent information industries along the coasts

Results

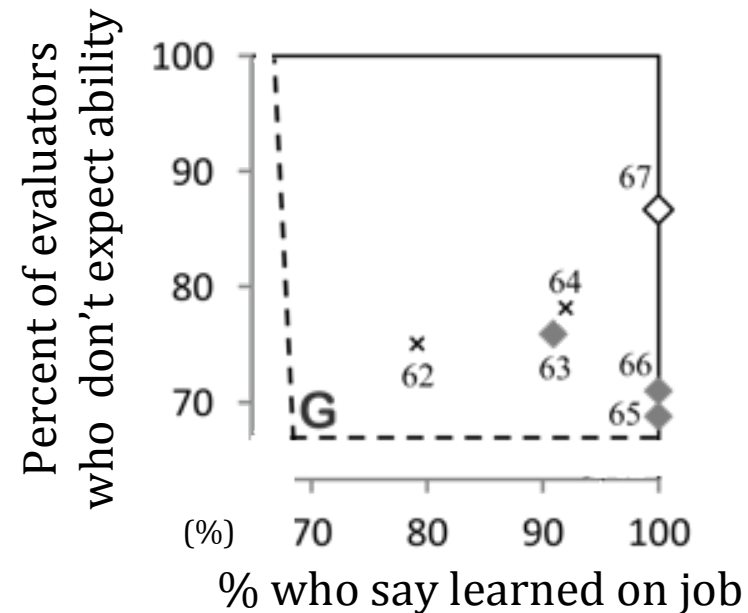Do you expect recent graduates to be able to…
…give clear high-level overviews?

Results: **No**, the ability is **learned on the job**

Chart: Y-axis: "Percent of evaluators who don't expect ability" from 70 to 100. X-axis: "% who say learned on job" from 70 to 100. Points: G at lower left, 62 (×), 63, 64 (×), 65, 66, 67 (◇).

62. Experience with document management systems

63. Communicate via online meetings

64. Experience with tools for project planning

65. Flexibility to communicate in different roles within an organization

66. Be aware of the knowledge and concerns of customers of the company

67. Be aware of the knowledge and concerns of business &/or marketing

Results: **No**, the ability may be either **learned on the job or unimportant** for software engineers
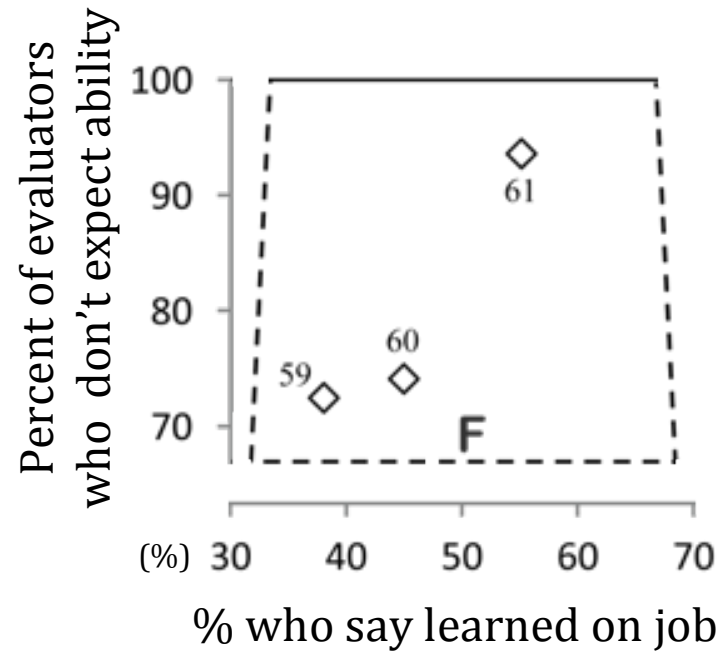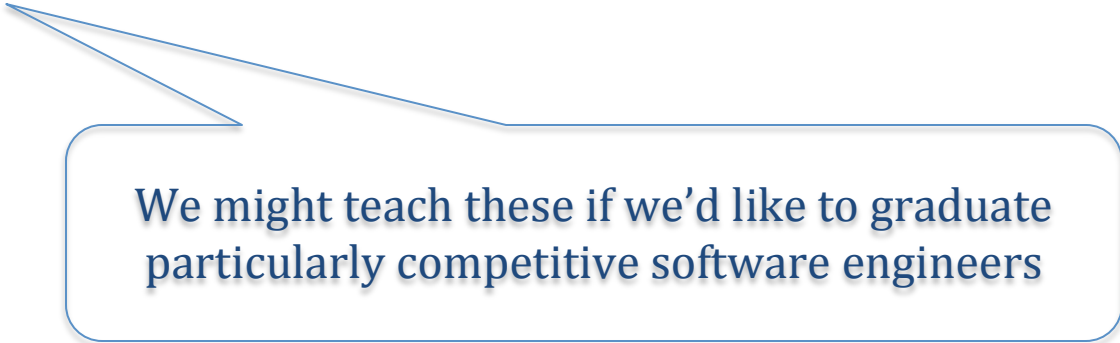

59. Communicate via conference posters
60. Communicate via journal articles
61. Be aware of the knowledge and concerns of lawyers

Either learned on the job or expected but lacking

- Use metaphors to communicate a system's purpose
- In conflicts, collaborate to identify win-win solutions
- Communicate across organizational boundaries
- Communicate to an audience of managers
- Communicate to an audience of UI designers
- Communicate to an audience of software architects
- Communicate to an audience of end users of the software
- Communicate effectively via conference calls
- Communicate effectively via formal requirements / specifications
- Communicate effectively via formal documentation
- Communicate effectively via code check-in notes
- Communicate effectively via bug reports

We might teach these if we'd like to graduate particularly competitive software engineers
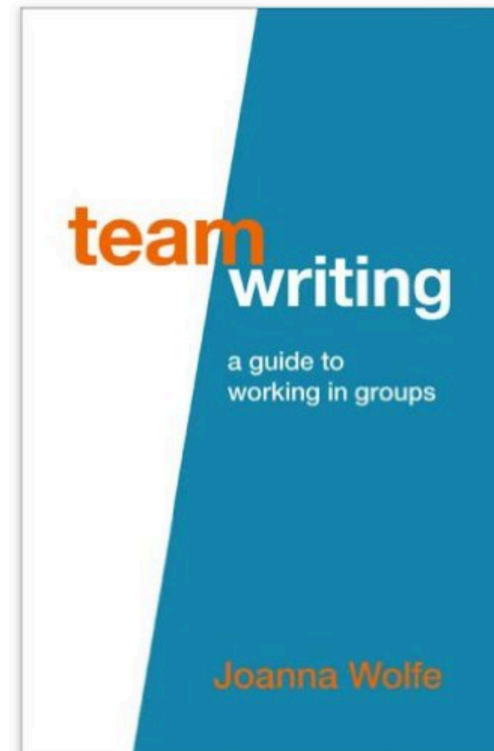
Implications?

Should we teach students to be nice?

For example,

    7. Adjust communication based on non-verbal reactions
    8. Discern when to ask questions rather than to assert an opinion
    9. Communicate with a balance of confidence and humility
  12. Listen actively
  13. Avoid taking debate, feedback, or others' opinions personally
  14. Discern when to keep silent rather than to speak
  15. Avoid complaining
  20. Respond professionally to one's own mistakes
  21. During discussion, treat others with respect
  31. Be nice to others, through words and tone

Can we?

e.g., *Team writing: a guide to working in groups*
by Joanne Wolfe

Implications?

Should we shift focus toward oral communication?

Oral

formal & informal presentations to a group	**expected**
one-on-one & group meetings	**expected**
small talk & discussion	**expected**
nonverbal communication	**expected**

Written

e-mail, instant messaging, code comments	**expected**
correct spelling	**expected**
formal documentation, specifications, other job genres	**disagree**
structure & formatting for fast reading	**disagree**
journal articles	**not expected**

Implications vary by institution and individual

The program must enable students to attain, by the time of graduation, an ability to communicate effectively with a range of audiences.

–ABET 2015-2016

Once again employers report that soft skills represent a more critical shortcoming of job applicants than technical skills. Communication remains the most cited shortcoming.

–State of St. Louis Workforce 2013

There is a widely held belief that 'soft skills' are greatly in demand in Computing graduates. There is less consensus on whether students lack them, and whether (indeed, how) the curriculum should provide them.

–*Computing Graduate Employability*, 2016
Council of Professors & Heads of Computing/Higher Ed. Academy