MY IT SKILL STACK

One of you reminded me of a good question that she asked in our very first virtual classroom exercise about potential interview questions:

"Which IT skills do you have that could help us to make better business decisions?"

In the spirit of setting a good example, I want to sketch an answer that I might give at an interview. Of course, nobody will ever hire me for anything that will require specific skills… beyond 50 years of age and a rather consummate career, you're only good for upper level management, which means that you don't actually do anything else than supervise and coach others who do the real work (hey, so much for my model of senior management, eh?). I'm only half serious here: the best managers, I believe, are those who could in principle do the jobs that they ask others to do. Even as a teacher, I think you should be able to solve problems and manage concepts at least as good as your students, and quite possibly a lot better, and faster, to have something to offer to the very best and fastest students, too!

Here's my list of IT skill groups with examples:

1. Problem solving skills

I am skilled in identifying, decomposing (for better analysis) and solving complex problems. The mathematician Polya has suggested a generic process for problem solving - especially for mathematical problems (these are the examples he uses in his book, "How to solve it") but it also works outside of maths:

1. Understand the problem: the conditions, the unknowns, the data. Of these, I am particularly good with data.
2. Design a plan of attack (e.g. by modeling - abstracting from the details to identify one or more routes or options)
3. Carry out the plan of attack: this is execution. Probably my least favorite part (often, when I see the solution path, I get bored). But I can do it, and it's satisfying to finish something.
4. Look back, review and discuss your solution. I am especially good at this type of post mortem analysis - it's probably what I use most when it comes to teaching stuff.

2. Computational thinking skills

Computational thinking is best built and trained by dealing with computers - not necessarily as a programmer! Even knowing how computers work, what they can do and what they can't do, is relevant. Computational thinking is a set of skills that allow you to properly assess the possibilities and limitations of current digital technologies. The more of these skills manifest themselves in practical skills the better.

E.g. I "speak" several (=10) programming languages: PL/I (for IBM mainframes), BASIC (my first), Pascal, FORTRAN (standard language for physicists of my generation), Emacs-LISP (because of the editor, Emacs, that I use for everything), C and C++ (learnt it as a graduate student), Python and SQL (learnt it as a professor, first because I had to and then because I thought it was fun), and finally R (which I only started learning this year, and now I do everything with R). Here, I really mean that I can understand the code written in these languages. I have not had practice in most of these for a long time and though you never lose the skill entirely, you do of course get rusty to the point of ignorance if you don't practice your practical skills by solving real problems with coding.

The two programming languages that I would recommend to anyone in business now are: SQL (for databases) and R (for data science).

3. Data literacy skills

These skills are perhaps the most discussed in this grouping. This could be because they're not well-defined or agreed upon, and/or because "literacy" is kind of off-putting for normal people who want to get stuff done (a little like "science" in "data science"). The Wikipedia definition is not bad at all, actually, in this case: "competencies working with data", in particular "the ability to read, understand, create and communicate data as information." Since everyone agrees that information = structured (value-added) data, this sounds O.K. to me.

In practical terms, data literacy shows when you use visualization and story telling techniques, such as: tables, graphs (bloxplots, histograms, heat maps, etc.), models (like diagrammatic models using BPMN, but also informal, non-standard models based e.g. on topology (system diagrams) or geometry (e.g. a pyramid to illustrate hierarchy). Storytelling techniques relate to knowledge about the minimal set of story elements or, for particular examples, good practice rules (e.g. Denning's rules for good organizational storytelling). It

also relates to the ability to be logical (cp. Minto pyramid) and know how to build a plot (e.g. in three acts, like Aristotle).

4. Communication skills

Though I had a natural aptitude for communicating, I believe, I am also a bit of an introvert. Hence, when I went into business after my PhD in physics, I lacked communication skills most of all.

My answer, after I seriously bombed leading a large team of over 40 people (because of my bad communication skills), was to get executive coaching. I liked it so much that I skilled up on coaching and became a coach myself (I only stopped doing that in 2016). You don't just get to be a coach: this also includes skill development. I studied this stuff for about 10 years gaining various professional certificates along the way (Gestalt therapy, Psychodrama, Constellation technique, healing practitioner etc.).

Today, I would count some of the "data literacy" skills as important communication skills, see above.

5. Tool skills

I really love tools. This summer, I am looking forward to digging into a bunch of tools that I left lying around during the teaching term. I am still as excited about new tools as I was when I was 14 and got my first computer!

For example, in my "standard" undergraduate IT course, I teach a lot of models, techniques and concepts, but I also use 20 different digital tools, which the students mostly have never heard of. Of course, this toolset changes all the time. Example: a few years ago, I started with Slack for projects when hardly anyone used it or knew what it was. Fun!

6. CONCLUSIONS

If this list really helps improve business decisions depends on the context, of course. More specificially: it depends on the problems, the processes, and the people on the job.

For a more systematic approach to skill listing and skill building, I recommend the world's best career advice book, "What color is your parachute" by Richard Bolles.