

NotesOn: IT HR– Part I – Recruiting - Attributes of a Successful Hire

Introduction (v1.4):

We have all seen HR IT job postings for junior developers (beginning programmers) that state a BS is required but an MBA is preferred. Yet, the Degree may be the least important attribute of a successful IT team member. Does that surprise you?

Two or three years ago, in particular, I noticed a new trend in IT recruiting and it began to concern me. So, I started keeping track of job postings and, lo and behold, the once suspected trend is turning into a reality that will do IT much harm and little good. And. Along the way. Gradually. I discovered that HR (Human Resources) IT has its own fundamentals, too.

Before we go any further, one note is of importance. Below you will find a discussion on the value of college degrees and their importance to IT. Let me say up front, that if you have a chance to obtain a college/university degree, please do so. But. Keep in mind that it is only one facet of learning, only one step in the process of educating yourself, a step that *can be* acquired else wise.

Introduction (V1.4):	1
Background:	1
Hierarchy of Successful Hiring Attributes:	2
Ability:	3
Ability Requirements Of An Engineer:	4
Ability And IT Fundamentals:	4
The Ability Attribute And HR:	5
Passion:	6
A Hiring Fundamental:	7
Passion And Success:	8
Experience:	9
Training:	11
Degrees:	12
Potential Upsides Of A Degree In IT:	12
Potential Downsides Of A Degree In IT:	12
Summary:	13

Background:

When I was a sophomore at the University California at Davis I went to visit a girlfriend and I went to visit her folks over a long weekend. I hit it off with her father, and in talking with him, a very intelligent gentleman who had mastered a number of fields of engineering and applied engineering without the official degrees, we spoke of college educations and he told me that which I had not yet realized for myself. College is not an education in data but an education in how to find data, and somewhat of what to do with it once found. Nor is a college education an end-all but a bare beginning.

In a recent reading moment (and I read a great deal during any moment I can find) I took up *Education of a Wandering Man* by Louis L 'Amour, one of America's greatest authors; of far more than "just Frontier" stories. My wife (though she teases me about the number of books that I have around on shelves and in boxes) bought the 1990 Bantam soft back edition for me as a present. On page 3, Mr. L' Amour states:





“No matter how much I admire our schools, I know that no university exists that can provide an education; what a university can provide is an outline, to give the learner a direction and guidance. The rest one has to do for oneself.”

And. Yet. In this day and age, “The Degree” has become all important, more so than anything else it seems when it comes to judging the worth of any member of an IT team. This is not as it should be, for it is an alteration of what is most important in hiring, raising and grooming IT team members to succeed. Moreover, it is casting aside, as worthless, men and women of genius who have long since added and will continue to add much to our endeavors in pushing the frontiers of IT forward ... if allowed to participate.

“The frontiers of IT?”, you say. Yes. For, as advanced as it may seem at the moment we have but placed a dent in the future of where IT will be, ten, twenty, fifty and a hundred years and more from now. Yet, with a great deal of hubris (excessive and perhaps unwarranted pride) too many of us in IT cast aside the willing, passionate participants who are not only eager to learn and to contribute to the here and now but perhaps they would have been our guide along the path of future history. Yes, we will still get there, one way or another (we have no choice in that), but were they to be allowed to participate to their fullest capability the way might be less painful, less costly and more successful. Merely because they lack a degree in IT, or lack a degree at all, is no reason to “cut our noses off to spite our faces”.

What is absolutely fascinating is that much (though not all) of IT was founded by folks who had no Computer Science degree, nor, in many cases, a degree at all. “Back then” (not so many years ago) it was not how many sheepskins one tacked on a wall but how many lines of code one could write, or how many PC’s one could build, or how many network components one could assemble, or whether one could develop a language to tackle a previously unsolved problem, and so forth, that would stand up to testing and acceptance. It was not how many certifications one had after an e-mail signature that counted but how many ideas one had during design sessions that helped create and improve and advance a product beyond those being generated by competitors.

What counted was what one could DO and the passion with which one did it.

And *that* passion, *the* singular driving force behind *all* that is successful, is being lost to IT in favor of degree requirements that are not a routinely accurate gauge of the quality of the individual, or their knowledge of IT.

By now you all have likely seen job requirements for junior developers (beginning programmers) that state that a BS is required but an MBA is preferred. This strikes me as being pure arrogance, as the degree is the least valuable component, the least valuable attribute of a successful IT team member. If you’ve spent any time in IT you know darned well that a degree is not a guaranteed measure of success, at best it is only a fraction of the story of what makes a valuable IT team member valuable.

But. Believe it or not. This is not a rant, a gripe session, against college degrees. Nor do I raise the issue on a mere whim. Those of you who have read and studied my other posts know better. The subject of degrees in general and education in particular *is very* relevant to the subject of IT fundamentals and thus success in IT. They are two of the fundamental inspection points when making a hiring decision, of both new and already successful IT team members. Just not in the way you may presume.

These days, too many job processing programs automatically are set to filter out, and toss into the electronic dustbin (often called the infinite bit bucket), anyone who is not a precise match to a degree requirement, thereby filtering out extraordinarily competent candidates; just as too often in the “rush for success” fundamentals of all types are tossed aside.

Hierarchy of Successful Hiring Attributes:

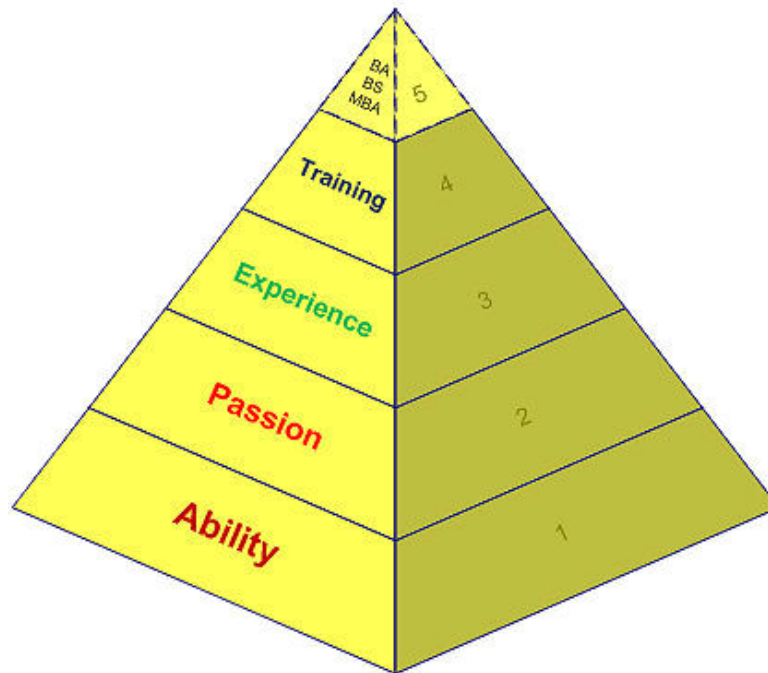
“So”, you might ask, “what is the gauge against which candidates should be measured?” My answer is that you might be surprised by the answer, for the gauge is, in precise order of importance (from most to least):

1. Ability
2. Passion
3. Experience
4. Training / Learning
5. Degree in IT or from any field



Of course the list, save perhaps for highly specialized fields of engineering and science, applies to more than just IT.

If I flip this sequence upside-down and add a bit of “glitz” we have a graphic representation of the fundamental elements, in order of importance, of successful hiring in IT:



© DP Harshman – www.fromtheranks.com

The Hierarchy Of Successful Hiring Attributes (from most to least important)

Figure 1

Let’s start at the bottom and work our way back up.

Ability:

Ability and Passion are neck-and-neck when it comes to being the most fundamental building block of a successful IT candidate. As a matter of fact I’ve had it both ways during the writing of this post, and in the notes that preceded it. In the end, Ability won out as the argument could be made, with some success, that true (not dilettante) passion for a subject is built on a natural bent, a natural leaning, a natural talent, or in rare cases a learned and earned aptitude, that gives one the sense that they *can* be very good at and excel at the subject at hand.

Not everyone who wishes to be an “IT Guy or Girl” has the true ability to be one. Even if they have the desire, which most folks don’t. My wife (a CPA, past CAO, CFO, Consultant, etc.), is absolutely brilliant at managing numbers and finances and financial processes and financial institutions ... but she is hopeless when it comes to “Thinking IT”. She has absolutely no interest in the underpinning’s of any IT system and has a very short fuse regarding, has very little patience with systems that don’t work and don’t do what they’re supposed to do. Being an “IT Guy”, I, of course, have desktops, servers, laptops, networks and so forth around the house and my wife is my own personal, if you’ll excuse the expression, “user from hell”. She knows exactly what she wants, and she



wants it to work *now* exactly the way she needs and wants it to work. And. When it isn't doing any or all of that, I hear about it (as do Help Desks wherever she works, which is the way it should be). But that is as far as it goes when it comes to her and IT. She has her requirements, period. Except. They have no idea how to do it.

Most of our users are like that as well. They know what they want, though sometimes you have to help them realize it with adroit questions, and they want it now and they want it done right the first time, having little patience with bugs and endless releases and sub versions and so forth.

The truth is, to meet that demand, IT requires a special mind-set that is not easily transferable from any other non-engineering disciplines.

And, yes, IT is an engineering field. It is. It is. It is. It has nearly zero to do with philosophy, sociology, psychiatry, the classic arts, music, etc., and very little to do with the hard sciences except applied chip and board development (the chained and channeled flow of electricity and magnetic). The "hard" side of IT is but a small part of the overall IT culture – an important part, a foundational part, but still a small part in the overall scheme. The "soft side" is where it is at for most of us. Even so, the required characteristics are the same for both hard and soft. And since we're on the subject engineering:

Ability Requirements Of An Engineer:

To be an engineer, a good engineer (in any applicable field) there are certain characteristics and skills that must exist, that one must possess. In no particular order for they all must be present to one degree or another, an engineer must:

- be detail oriented ,
- have a keen ability to focus for extended periods of time,
- be capable of extended trains of logic,
- be persistent,
- have patience,
- have an insatiable curiosity as to "how things work",
- have a strong interest in solving puzzles,
- have a deep interest in building "things",
- have a passion for solving problems that get in the way of building "things".

There may be others, but these requirements apply to any and all IT personnel who are in any way related to building "things". Especially the techies, but the business analysts and project managers and managers and, yes, even the executives. None are exempt from the above minimum characteristics if you wish to build a high power IT group.

Ability And IT Fundamentals:

As presented in [PPT: IT Fundamentals Supersede All Methodologies](#) (which can be found on www.fromtheranks.com), what IT is ultimately about is building systems that help our users do some "thing" better. In order to build systems one has to have a solid comprehension of the user's subject matter (most often business or business related) or the skill and willingness to learn it. And one has to be reasonably good at math. And with logic, but an applied logic not a philosophical logic; which I'll address another time. Also, to build systems one must be able to keep a wide range of elements inside one's "skull" at any given time, and stay focused on them for an extended period of time, while aligning them into a logical order that can then be translated into "software business applications", in a way that users' of any type and stripe can understand and apply to their work – to *their* solving of *their* problems.





One must be able to predict future responses to a given set of application circumstances with a fair degree of accuracy and apply that predication in a way that is acceptable to the users of the system – too freeform and the users get lost, too restrictive and they can't do their job, too “engineering mode” and they'll get confused and frustrated when they can't do it the way they would do it.

One must be able to plot out a logical sequence of events, then start that sequence, continue that sequence, tweak that sequence as needed, and finish that sequence; all while keeping a half a hundred or so “mental objects” juggling in the “air” at one time. That may be a slight exaggeration but not by much for complex business applications and integrated suites have a seemingly unlimited number of “threads” that must be accounted for and kept track of ... in sometimes excruciating real time detail.

To do all of that right requires (not optional, requires) discipline, attention to basics / fundamentals, the ability to focus (really focus) for hours on end and the willingness to spend uncounted hours in front of a keyboard and display screen.

Many have the *desire* to be “IT folks”, few have the engineering *ability*. Fewer still have the ability to build and successfully lead successful IT teams. By the way, “desire” is not a characteristic of (at least not a primary characteristic of) a successful IT candidate. Desire is not enough. Ability and Passion must be there, first and foremost.

Again, IT Leadership is not exempt from *any* of the above requirements. Additional learning and experience is required to be a good manager and executive, but no manager, no IT executive can long survive if they do not have the same personal skills and abilities and passion as their “techie”. I've seen a few exceptions, very few, but as a rule the best IT managers and IT executives were “dyed in the wool” techies first. In a later example you will hear of one such junior executive who had no business being in, let alone over IT for he had neither the Ability that IT demands, nor the Passion for IT.

The Ability Attribute And HR:

The problem with the Ability attribute is that it can be hard to measure and thus hard on HR. When jobs are aplenty, HR has time to do at least initial phone interviews but in difficult economic periods when there are far more applicants than jobs, too often “short cuts” are taken, including a heavy dependence on automated resume and qualifications filtering systems, based on checkboxes. As hard as it is on HR to *not* use these, in the long run this method is deadly to IT groups. For instance:

Of late it has been spread around that an assessment can be made of one's ability based primarily on the number of certifications and/or degrees one has obtained. While beneficial to business that provide training, that is neither fair nor true. Many IT team members, successful IT team members, have studied endlessly on their own, the volume often far exceeding the information studied in formal college or training classes.

Another factor used to form a judgment is the resume, but as we all know this can be “tweaked for the occasion” and can create an impression that is not entirely (or necessarily at all) accurate.

At one company, we developed and used tests to gauge an applicant's ability, and it worked, up to a point. The problem with tests is that some of the old applicants shared them (at least verbally) amongst their friends and new applicants (in other words they cheated), so tests (manual or on-line) while of some value are not the entire answer either.

Testing Note: as most languages and environments have at least six ways of doing the same thing, i.e. creating the same result, developing excruciatingly detailed tests with only one “correct” answer is a waste of time. It is better to focus the test on key concepts (and fundamentals) and have an experienced IT person grade the test. Check boxes and true or false generally won't do. The PMP certification test is an example of a good one: experience is required to take the test and experience, not just rote memorization, is required to correctly answer many of the 200 questions.

The first best way to judge ability is via one-on-one meetings ... the candidate meets with several individuals in the company (techies, managers and execs), who know the job's responsibilities cold. Again, the same rule applies as regards testing, there are so many ways to “slice a banana” in IT that looking for just one answer could eliminate genius from your ranks. So primarily interview against concepts.





But, as the one-on-one interviews come *after* the initial filtering process, what is HR to do before?

First, unless it is a very precise engineering role (say down at the chip fabrication level) I would recommend eliminating the degree requirement from the job postings and application processing filters. Too many otherwise extraordinarily talented IT team members are intimidated by the “absolute” requirement of an IT degree and the loss is not theirs but your company’s. (How many companies were started by inventors and entrepreneurs with no degree whatsoever? The answer is: most or nearly all.)

Then, by all means, study the resume, but with an emphasis on the overall presentation of the fundamental hiring attributes listed herein, rather than an exclusionary “lightning round” that eliminates the entire application based on one single factor. And. Assume, initially, the applicant is honest with an understanding that as the process continues the “exaggerators” will filter themselves out.

For example, here is an edited down job posting from my files (it was actually much longer):

Java, OOA/D, DDL & SQL, HTML, XML, UNIX, Linux, Windows, MSSQL Server Enterprise (Transact-SQL, StoredProcedures), Oracle, PostgreSQL, JBoss, InstallShield/MSI, JUnit, LoadRunner, Rational Rose, ClearCase, C/C++/C#, AJAX, Tapestry, Hibernate, Servlets, Applets, Tomcat, Weblogic, Apache Maven, Eclipse, Eclipse RCP framework, TCP/IP, HTTP, HTTPS, SSL/TLS, X.509 certificates, ...

Now, if you were fortune enough to find someone who had a verifiable history of working with *all* of these tool sets (and someone would have to be experienced indeed to have successfully used them all) would you or your system eliminate them if they didn’t have a degree?

Because, you see, this job posting had an additional non-negotiable requirement of a “Bachelor degree in Computer Science or related discipline”.

What would your job application processing system do with our hypothetical non-degreed, otherwise highly qualified potential candidate? Discard their application? What would you (as an HR representative or interviewer) have done with them?

In many HR systems you wouldn’t even know you had such a candidate on your doorstep; your automated review process would have filtered them out causing you to waste months if not years looking for the perfect, ideal, candidate with an irrelevant degree.

I spoke with an HR recruiter some while ago who had this exact problem. She told me my resume was perfect, my experience was ideal and exactly what they needed, and they had no doubt that I could do the job ... but ... as I didn’t have an MBA they wouldn’t make me an offer ... because ... are you ready? ... the company’s IT executives had laid down a policy that even their most junior developers had to have an *absolute minimum* of a BS in Computer Science but *preferably* an MBA. ... Needless to say, there was a stunned silence on my side of the conversation ...

Was I disgusted? Of course. Is that software company still in business? No, of course not. (Out of curiosity I kept track). They had their priorities 180° out of sync with what is most important. The company was on a “status trip” having *nothing* to do with success.

The moral is: don’t focus on degrees when they are irrelevant to a job. That is an expensive waste of time and resources. Focus, sharply, first, on Ability. Your customers will appreciate it. Your stockholders and/or investors will appreciate it. And, trust me, your IT group will appreciate it. When crunch time comes it is not a Degree but Ability that gets “it” done.

(Don’t get disgusted or impatient with me, we will get to the value of college degrees in a short while.)

Passion:

When I left college I started my career in a field that I thought I wanted to spend the rest of my life in, architecture and engineering. I did pretty well at it, at the end of my ten years in the trade I ended up as the General Manager of an architectural firm in Southern



California with a million plus square foot of homes and buildings behind me. But. I often found myself bored. There was not, for me, sufficient challenge – I have a very low threshold of boredom. Then, quite by accident, I found my true passion.

It happened as a result of my being charged with automating the back office; business was going well enough that manual book keeping, invoicing and so forth was becoming a problem. We had already dipped our toes into CAD (computer aided design) but all else in the offices were done manually. “Back then” IT was in even more of its infancy than it is now and everything was “cast in concrete” as we used to say; you either took the application and system the way it was, *exactly* the way it was, or you did without ... unless, you wanted to pay a LOT of money for modifications, usually far more than the total initial cost of the system.

After months of painstaking research I was getting fairly disgusted with the entire subject. I was not at all impressed with what I was finding in the marketplace. Until, I found a company building software that could be custom tailored for nearly every need without constantly re-coding the various accounting, distribution and inventory control modules. The idea of “nearly on the fly” flexibility seemed an absolutely brilliant notion so I went to work for Business Software Products (dba Distributor Business Systems). Although they don’t seem to be in business these days (could be wrong, might have gone through a name change) at the time it was a cutting edge notion and we did quite well. It wasn’t too long before other software houses started following in our footsteps.

More to the point, I wasn’t six months on the job before I was starting to write code, which I had never done before. My new journey started with my curiosity as to “how” the customization was being done, followed by how the configuration points could be further customized, and then how could additional requested features be added, and then ... Soon I was not only helping to complete the unfinished application modules but I was also defining, coding and creating “standard” reports for each module, as well helping to install entire systems from scratch (running network cables, building and burning-in servers, etc.), training users on those systems, giving classes and seminars on the theory behind and use of the systems (and the theory behind the theory), providing consulting services to executives, and so on.

While there, I not only helped to sell, gather requirements for, design, build / customize, install, and train folks on over 40 systems, I discovered my true passion. And that passion drove me to learn everything I could about the subject of computers and IT, *no matter the source*. No longer the least bit bored (for every day in IT is different) I moved on to other companies and absorbed other subjects and skills, becoming adept at, among other things, UNIX and Windows, servers, networking, proper database design and tuning, a host of software development languages, audit controls, project management, IT management, etc., etc. I practically inhaled IT, taking every class I could get my hands on and reading and studying books when I couldn’t.

One of the takeaways from that journey was the value of a logical progression of increased skills, one built on top of the other. Had I tried to shortcut that learning process (educational and experiential) anywhere along the way, my effectiveness as an IT member would have been reduced. (There is a lot to be said for apprenticeships, formal or otherwise).

But the primary discovery was that I had a true passion for IT.

A Hiring Fundamental:

As a tech lead, project manager, manager and director I observed, learned early and thoroughly, remember always, and apply regularly, one key fundamental when it comes to hiring and managing personnel:

Those who have the *passion* for IT will almost always excel in IT and are almost always very easy to manage. (Of course you have to continue to feed that passion to keep it burning brightly.)

Those who do not have the passion for IT and thus are in IT for the wrong reasons (and there are many wrong reasons) are most usually problem children and do some of the screwiest most disruptive things one could hardly imagine.

You think I am joking? I am not. In fact, allow me to throw down a gauntlet:



I will put a team of dedicated, passionate IT team members (without a single degree amongst any of them) up against any equivalently experienced team of non-passionate IT members with as many degrees as you wish ... and my team will “shut them down” time after time – my team will provide superior quality, superior performance and superior usability.

Why? Because the passionate person *will* find a way to solve problems, they *will* come in early and stay late until the completion target is met, they *will not* limit themselves to what was taught in some class, any source of information is fair game, and they *will* have pride in their work such that they won't dump “half baked” software over the wall, letting the “users” test it for them. Here is an IT motto worthy of being painted on every wall in the area:

Pride counts. Pride comes from quality work that exceeds expectations.

Quality counts. Quality comes from dedication (and focus).

Dedication counts. Dedication comes from the passion to succeed.

Passion counts. Passion comes from the joy of doing.

Time and time again I have seen the passionate IT team member out-perform the degreed individual who brings to the table an attitude of superiority, or sense that as they have a degree they are automatically “owed” something (unwarranted salaries, titles, preferences, etc.). This is not always the case, of course, for when you combine passion with a superlative college education you can *also* have a super star on your hands. But. It is not the degree that matters so much as the passion to put what has been learned to good use. Degrees do not equal passion. Passion equals passion. Ability plus passion equals increased potential for success.

Passion And Success:

A couple of years ago I was a presenter at an IT conference in San Diego, and I offered this “thought for the day” to the audience of 200+ IT “techies”, managers and executives (mostly the latter). I said, right up front, without equivocation, early on in my presentation that one of the primary fundamentals, one of the driving forces behind successful IT members and organizations is their passion for IT ... then I stuck my neck out by adding ... that if they did not truly have a passion for IT then perhaps they should find a field where they did. I received a couple of dirty looks, okay more than a couple, but I also received quite a few compliments ... from those in attendance who truly possessed the passion for IT.

Last year I attended the 3rd Annual Angel Capital Summit in Denver, CO, sponsored by the [Rockies Venture Club](#). At the close of the summit a gentleman by the name of [Jack Zufelt](#) spoke and he echoed what I had long believed, but never put into words, as he laid out the true key to success. I won't try to recapitulate, summarize, the entire presentation (if you get the opportunity to see him in person do so) but, in very brief outline form here is what he had to say (these are from my notes so any errors are mine):

- Success is not a technique
- Procrastination = you don't want to do it
- Discipline is from the root word 'disciple' = if you are dedicated to “it” you don't have to be pushed
- Your heart must be in it
- Success is based on a core desire
- A core desire overcomes every obstacle = 100% dedication, on a scale of 0 to 100 it is 100

I don't recall Jack using the word “passion”, though he may have, but that is how I mentally translated and condensed what he said in my head, in part because I had both felt and used the term and concept of passion before in relationship to success in IT.

If you don't have the passion for a subject, you may muddle along (being miserable or at least unhappy) but you won't succeed, certainly not to the degree you wish. If you have the passion for a subject you will have the discipline and drive to succeed. How you define what success is, is up to you of course, but, among other items, if based on passion your success will be founded on honestly examining and trying out what works and what doesn't work; throwing out what doesn't and keeping and building on what does. And what does work is hiring folks who have both Ability and Passion for the subject.





Again, it does not matter your desired IT role, whether it be “techie”, business analyst, project manager, manager or executive, if you have a passion for IT you have put your feet on the pathway to success in IT. Ability and Passion alone are not an absolute guarantee of success but if you stick with it “a core desire overcomes every obstacle” thereby increasing your odds of succeeding dramatically.

And. As a bonus. Your passion will transmit itself to those with whom you work and to those whom work for you. Passion alone can lift an otherwise mediocre IT team to levels of excellence that it had no true idea existed.

As I told the conference, if you do NOT have the passion for IT then I invite you to find another field of endeavor. You will be, and IT will be, far happier.

So, passion is the number two success criteria when hiring IT personnel. You will find it coupled closely with Ability, or in the case of brand new IT team members (junior programmers, analysts, etc.) *latent* ability. Look for the passion (after some practice you will see it) and then, in the case of a new hire, new to IT, monitor their latent ability.

It is more than their saying to you: “I have a passion for IT”. You will not only see it in their resumes, but in their interviews. You will hear echoes of it when speaking with their references. And, once hired, you should (given a reasonably logical organizational structure and rational processes and procedures that foster excellence) quickly see it in their work product no matter what level they start at. True passion leaks out. It shines through. If the hiring process doesn’t obfuscate it, cloud it over.

If they don’t pan out, maybe they threw up a good smoke screen (and this sometimes happens to the best HR recruiters); if you somehow misjudged their ability and their passion ... you simply recommend they find another avenue towards success as you escort them out the door.

Experience:

This attribute is also senior to college degrees, and to training, although from some job postings one certainly would begin to wonder if that were true.

For instance, I just saw one not very long ago for an IT Manager. All of the requirements were reasonable ... until it stated that a Masters was required but a PhD was preferred. My goodness. A PhD for an IT Manager? In a very, very specialized field, maybe, but otherwise? Curiously, there was little mentioned about the depth of experience required.

Applied experience is at least equal to if not superior to any number of degrees and such a candidate should not be filtered out because of the lack of a piece of paper.

An experienced IT professional, at any level of an IT organization, is worth their weight in platinum. They have the “been there done that” glow of competence that will help your IT group dodge a lot of nonsense and help your company short circuit potentially destructive tendencies and “latest trends”.

I know of one “executive” who had a boat load of degrees but zero useful experience in IT. He promptly proceeded, and I do mean promptly proceeded, to destroy his entire department, by:

- offering neither leadership, mentor-ship nor coherent direction,
- establishing an adversarial relationship with the department from his first day (the concept of meeting the team half way was held in complete disdain, complete contempt, by this individual),
- constantly shooting from the hip (i.e. issuing unfounded orders, unreasonable demands and unsound expectations based on ever changing whims), often overriding previously established (and workable) processes and procedures,





- going directly to the “techie” and issuing orders and making changes; completely bypassing his managers and technical leadership without informing them of the orders or changes,
- trying to be both a “techie” and an “executive” (without the skills to do either well),
- making promises to customers that couldn’t be kept without extraordinary expenditures in time and money and resources – one way he did so was by always saying “yes” to any and *all* customer requests no matter what they were (the old adage that the customer is always right is a falsehood, particularly in IT),
- insisting on pet technical solutions that wasted millions of dollars, and
- refusing to stand behind his technical teams, throwing them under the bus at the drop of the hat (even to dropping the hat himself) thereby destroying what little morale remained.

Within six months he had annihilated the department’s reputation for excellence, one that had been years in the making. Six months or so after that he was gone. Leaving behind an overloaded nearly crashed IT team in a serious mood for “a lynching”.

You can successfully educate someone in the finer points of management who already has a history of success and competence in their field. But. It is very difficult to educate someone on how to be successful and competent unless they have a foundation of Ability and Passion in that field.

By the way, neither of these two success factors come from a college education. That is not education’s purpose. Educators can uncover Ability and heat the flames of existing or discovered Passion, but they do not create them.

A degree represents a certain number of courses completed (classes attended, books read, labs fulfilled and tests passed), i.e. data acquired. It does not, by any stretch of imagination, replace experience; not now, not ever. And yet, more and more, IT organizations are trying to turn that invalid assumption into fact, and failing because of it. Education can facilitate the gaining of experience, if the individual uses it as a foundation and not an end point, but it does not replace hard won experience.

Recently I was speaking with a professor from a well known School of Business. I made my case, much as I am with you here, and ... he agreed. Which is one of the reasons why they are re-gearing their MBA program to involve heavily experienced individuals in the teaching of their curricula and having the students invest a good deal of sweat equity into what they are learning – it is being recognized (slowly) that theory alone is insufficient.

If you’ve been around for a while you may recall that the vast majority of what we call IT was designed, built and constructed, i.e. “figgered out”, by folks who had no degrees, or if they did they had them in areas having nothing to do with IT. The same quality level of un-degreed talent is still out there. That same measure of passion still exists in at least some new members of each subsequent generation, whether or not they have or are able to afford degrees. What this means to IT HR and to its executives and managers is that if we cut that resource off at the knees, based on an arbitrary filter, we will find our level of success at finding and retaining innovative IT team members suffering.

Yes, one needs (in theory) a degree in electronics engineering before sitting down to design new chipsets – though that is certainly not absolute truth. I’m willing to bet there is more than one “practical engineer” out there who has studied the subject within an inch of its life and, not being boxed in by, what is too often collegiate Should’s and Must’s and Must-Not’s, may very well come up with the next big breakthrough. College educations *can* provide an excellent background of information with some context. But, what if the “answer” is completely outside of the collegiate information framework?

To the degree you have eliminated all non-collegiate experts from your team, you have reduced and eliminated your potential pool of alternative solutions and “next bright ideas”.

I would be willing to bet that the majority of the great inventors were educated folks but not degreed folks, and many if not most of the “big idea breakthroughs” came from same.



I once told a friend who was feeling a bit down that “feelings don’t get things done, getting things done gets things done”. Similarly, “degrees and pieces of paper don’t get systems built, knowing how to build and then building systems gets systems built”. And that takes an understanding of the subject’s fundamentals coupled with experience. As founded on Passion. As built atop Ability.

Training:

In many circumstances, most I would say, Training is senior to college degrees for one very good reason – if the material being used for the training is based on sound experience and solid fundamentals (versus untested ivory tower theories) it allows the student to drill down into a subject in a very focused manner. And you will recall one of the attributes of a successful engineer (and thus IT candidate) is: the ability to focus?

One project given to me to manage required an application development toolset that no one in my group knew anything about. Fortunately, we were a very experienced team, in another field of development, and my techies had their basics and their fundamentals down cold – we had successfully cranked out some seriously complex applications in short order – which is why we were given the project. So well had we done on previous systems that the executives decided we could handle this new one as well, even though, again, we knew next to nothing about the required platform or the required development environment.

After giving it some thought I decided to do two things:

1. train the team on the new development environment by sending them to certified courses on the subject – by the way, sending teams off site is better than in-house, if it can be done, as there are less distractions; then
2. once we laid down a common foundation of understanding, I brought in a “guru”, an experienced individual, to further train and mentor them on the new platform, environment and tool sets. She also helped to build the framework and application but that was not her primary focus. Very much to the point, her passion and expertise, and her confidence, communicated quickly to the team and the end result was that within a few months they were nearly as expert in the new technology as they had been in the “old”.

In hindsight, my solution was a blend of training and acquired experience laid on top of existing experience, competence and passion. If you don’t mind my patting myself on the back, it worked brilliantly. And it hadn’t been necessary to fire the entire team and bring in all new “techies”, or out-source or ship everything overseas. Those would have been horribly disruptive, wholly unnecessary “solutions”.

If you have someone who has a great passion for the subject, you can very quickly bring them up to speed by using selected training courses. They will eat it up, they will inhale the material and, once you’ve provided the basics, they will often go out and buy their own books and/or find their own resources ... and before you know it, they will be accelerating rapidly towards expert status in the subject (although there is always more to learn in IT, always).

By the way, there is an underground datum that is rarely mentioned but always true. It goes something like this. Once you’ve learned your second or third or fourth language, once you become educated on the in’s and out’s of your second or third or fourth database ... a funny thing happens. You realize that all languages and all databases must do the same things. They must. So the learning process accelerates, sharply. After the “light bulb” goes on, one realizes that it is simply a matter of learning the new syntax and any new or additional features. But.

The bottom line is that ALL languages must ultimately do the same basic things, they all must talk to the various pieces of hardware, they all must present information on various pieces of display equipment, they all must communicate across the web in the same basic manner, they all must use and rely on databases in the same basic way ... and all databases must do their “thing” with equally predictable results.

Example: I have designed, built and supported hundreds of databases using: Sybase, MS SQL, DB2, MySQL and two or three other database engines I've long since forgotten about. And. They ALL must, in the end, do the exact same things, must serve up the exact same functions, etc. to be of any value to the business and scientific communities. At most it would take me a few days to come up to generally useful speed on a new one. And a couple to a few additional weeks to become highly proficient.

So. Just because someone doesn't know a specific language, or your company's specific database, if they have mastered two or three or four or more comparable toolsets you might consider looking at their overall level of experience as the primary hiring factor.

The bottom line: don't limit your company's resource pool by too restrictive filtering.

Degrees:

If you have a degree, or two or three, it is entirely possible that you are swearing up and down at me, possibly ready to throw things at me. I mean, after all, you spent four or more years and likely at least \$100,000.00 on your higher education and "... Darn it, it should be worth something! ... Or why bother!!!"

So. Let me be very clear. Do not take what I've said the wrong way. I don't hate folks with IT degrees, I don't think they should be drawn, quartered and then made to walk the plank. On the contrary.

There *is* a place in IT for folks with degrees in engineering and Information Technology! What is learned in college is valuable. The experience of studying a wide range of subjects with others can be (is not necessarily but can be) uplifting. Degrees in IT are not anathema to IT, they do, as I have said, have a place. ... Just *not* to the exclusion of the *first four* hiring success factors.

Potential Upsides Of A Degree In IT:

A plus is that graduates have shown the persistence to slog their way through four years, or more, of classes, and they may even have paid for it themselves. That is good. If one ignores the beer parties and so forth.

The opportunities to learn and explore new directions are phenomenal. If one takes, has taken, advantage of them.

Even though many of the required courses to obtain a Computer Science degree are irrelevant to IT itself (or nearly so), nevertheless college intentionally provides some breadth to a student's education ... in an effort to ensure they don't have a mono-vision view of the World (history, foreign languages, arts, etc.). Such an expanded view is good, particularly if true "critical thinking" is also taught; although a similar breadth can be acquired else wise. As a minor, they may also have (should have) studied basic business, business accounting, and so forth, which is also a plus.

So, do not in any way interpret what I've had to say here as a rejection of a college degree. If you have the opportunity by all means go. (Just remember one of your primary engineering characteristics: the ability to stay focused.)

Potential Downsides Of A Degree In IT:

On the downside, however, the core IT curriculums at colleges and universities are almost always behind the technical curve, sometimes significantly. Years after the IT industry has dispensed with a platform and language many Computer Science programs still teach them, sometimes exclusively (ex: in select cases FORTRAN still has *some* value, but close to 98% or so of IT is dedicated to and focused on Business, one way or another, not Science). As a result, that a new employee has an IT degree isn't necessarily an advantage, as you often have to take the new college graduate and train and/or re-train them before they can be useful.

If you've been around any length of time, you know this is true: that a graduate fresh out of college/university does not have the skill-set to just "drop in" and "plug in" to your IT group. There are exceptions (those with passion who were doing some form of IT for years before they got around to getting a degree) but they are rare.



The rule of thumb we had in engineering “way back when” was that an engineering graduate was pretty close to worthless for the first FIVE years (at least, which is a pretty significant investment for a company). Assuming the student went right to work after graduation, it takes them that long to get their feet on the ground and figure out which end is up, to figure out what was most important out of everything about engineering they did learn and what wasn’t, and to figure out what they haven’t but need to learn. Applied engineering is almost always wildly different than what is taught in a classroom.

Further, too many students take the *framework* of available information provided by colleges as the extent of known knowledge rather than the first panel of a road map from which to proceed and explore further ... too many take the fundamentals they are taught, when taught them, and treat them as absolute limits never to be looked beyond. A few do not, of course, but too many do.

Then factor in the following:

1. once passed how many courses are forgotten? But, more importantly:
2. five or ten years after the courses were taken, how relevant are they? Remember, this is IT we are talking about, where technologies take quantum leaps every five years or less.
3. IT is too broad for one umbrella. There are too many tools out there, too many technologies, too many approaches, too many platforms, too many development environments, too many database engines, too many protocols, too many infrastructure architectures, too many “many’s”, too much change for a CS degree to adequately cover them all and have more than a brief passing relevance.

Two to five years after graduation, assuming the student was trained on cutting edge to begin with (a premium assumption), their original knowledge base is out of date. After that you must depend on: their Ability, their Passion, their Experience and their (subsequent) Training. So why not judge your applicants by these four primary criteria to begin with?

So, please, do not limit yourself to just those applicants with degrees, for you may be hiring yourself into a box from within which the innovation, originality and “occasional spurt of genius” you constantly need to compete may not appear.

Again. Don’t get me wrong. Having a degree is not a bad thing. It *can* be a good thing. But. Except in select fields such as electrical engineering, chemical engineering and so on. Is it the end all? No, not hardly. Having a degree in an IT related field is *a factor* that you *may* want to consider. But. It is *not* senior to. It is *not* superior to: Ability, Passion, Experience and Training.

You would be better served to modify your job application processing system to *not* cutoff the heads of potentially incredibly valuable and successful candidates for lack of a degree that, to be honest, may be meaningless when it comes to your IT group and your company.

Summary:

The increasing emphasis on possession of a degree, over all other factors, is a very weird, contagious, recruiting and promotion psychosis that has been sweeping through the IT industry over the last few years (much like off-shoring as a solution to anything). Degrees are nice, they can be of value, but, again, they are not more important than the other, more senior, attributes.

Focus first (and when appropriate exclusively) on the first four hiring fundamentals. And your IT group will prosper and grow to that degree.

Hope this helps,

DP Harshman

PDF Link

