

NotesOn: IT HR – Part II – Characteristics of IT Newbies

Introduction (v1.3):

IT, more than any other industry I know of, has and requires people with a certain amount of “attitude”. Attitude, a certain status as a “prima donna”, can be a good thing. As long as it is managed and focused. Allowed to “run amuck”, however, and there can be problems. Newbies in particular enter the portals of IT with, as a general rule, a good deal of attitude that manifests itself in characteristics that border on and can cross over into weaknesses and future failure points. These weaknesses can absolutely be turned into strengths. But. You need to be aware of them first.

Let me be very clear, up front. I am not “picking on” anyone. This is not destructive criticism, just the opposite. Nor do I presume that all newbies exhibit all of these characteristics. That would be an unsupportable generalization. However. These characteristics are so prevalent across each generation’s “newbie population” that they can be enumerated, and discussed as a coherent whole. Again, it is important to understand that my purpose in doing so is to help new IT members, and their managers, address these weaknesses (for they are weaknesses) so they can make the transition into high performing energetic contributors to the IT Dream.

Of increasing interest as I began observing and documenting these characteristics a few years ago (scribbling down notes here, jotting down notes there), was the fact that I began to notice that each strata of IT (techie, managerial, upper management, executive) has its own set of newbie characteristics. However, because there is a certain degree of cross-over between them, I’ve chosen to loosely group them under just two headings: “Management” and “Techie”.

Enough with the introductions. Let’s dig in, starting with the Management strata.

Introduction (V1.3):	1
Management Newbie Characteristics:	1
“It is NOT Overcomplicated ...	3
“We don’t need more organization ...	3
“Forms, forms, we need ...	4
“Rubber must meet the road ...	5
“Upward relationship management is ...	5
“I’m an MBA ...	5
“Yes!” ...	6
Techie Newbie Characteristics:	7
They want to code, ...	7
They are easily mesmerized by ...	7
Their Coding Standards ...	7
They want to slip in <i>their</i> new ...	8
They often (almost literally) abhor ...	8
They see no point in testing ...	8
They often have Prima Donna ...	9
Summary:	9

Management Newbie Characteristics:

Unfortunately, to too many Executives and Managers coming newly into IT, IT is “just a bunch of geeks” running around in unrecognized patterns playing with expensive hardware and software who, somehow or other, manage to get (some)



things done, some times. There are so many moving parts in IT that, to the uninitiated, it can appear to be the poster child for the chaos theory. Nothing. Could be further from the truth. IT, when run correctly, is a finely tuned very high volume manufacturing operation that can and does boost the rest of the business into an entirely new strata of success. If it is managed properly.

Further, to too many IT Management newbies, IT is a mere “stepping stone” into the rest of the organization and therefore the Passion that is required to drive home excellence and grow an IT group into a world class organization is missing, completely. For these transients, time in IT is merely another box that has to be “checked off” on their way to a C-Level position and “ultimate success” (whatever that means to them); preferably as quickly as possible. This attitude, and it is far more commonplace than you might wish to believe, is horrifically destructive to IT ... and ... to the business community as a whole.

Why? Because. Nearly every ... single ... blessed ... aspect ... of business these days, better than 90%, probably better than 95%, is captured, routed, managed and reported by ... IT. If there is one element of the business, any business, that MUST be understood ... and understood well ... by up and coming managers and executives it is ... absolutely ... IT.

Think of IT as a service engine. The fuel that you put into IT determines what you get out of IT. The reality is that IT is there, and exists, solely to help the business and its clients and customers do “things” better, whatever those “things” are (see [“PPT: IT Fundamentals Supersede All Methodologies”](#)). But. That mission can only be achieved ... when IT’s executives and managers understand what IT is about and understand IT’s intrinsic value to the entire operation.

It isn’t that IT is more important than any other part of the organization. It is not. But, one way or another, IT is in charge of the *entire* communication stream of the organization. All of it. Every in-bound. Every out-bound. Every nickel is tracked by IT. Every e-mail is passed through by IT. Every operation is monitored by IT and reported on by IT. Every metric is accumulated by IT. Every graphic, every plan, every progress report, every to-do list, ...

So, when you turn new executives and managers loose on IT, there are a number of characteristics of which you need to be aware. Failure to do so will, not may but will result in negative impacts on the IT organization. And. Thus. The entire business. Please. As an HR representative, as a senior executive, please do NOT dump new execs and managers into IT in order to “season them” ... before yanking them out to do “something else”. You might as well take a .45 semi-auto and shoot yourself in both feet.

Don’t get me wrong. There is nothing wrong with assigning “new hires” to IT that have a potential for future management levels elsewhere within the organization. But. Understand that IT has a long learning curve. It is not “conquered” in a year, or even two. As noted above there are so many moving parts that full comprehension, and appreciation, of IT takes a while. As a group of MBA students recently discovered.

I was invited to give a presentation to the University of Denver’s Daniels School of Business. You can find the PowerPoint on this site ([PPT: Idea to Execution – Working With IT](#)) or on Slideshare (“Idea To Execution – Working With IT”). This presentation was at the 50,000 foot level. And took 40 pages and 2 hours to go through, with insufficient time for all of their questions. None of the students were IT majors, so my original intention was to keep it as light, and high level as possible, only giving them only a taste of the overall process of working with IT.

The feedback from Professor Bauer was that even at that level it was a LOT of data: “... most felt overwhelmed by the amount of material we tried to cover”. And yet, there was darned little that I could have sliced out of it and still left them with a coherent concept of the whole. Again, this was the 50,000 foot level, maybe higher. [Later on he sent me two of the student’s comments on the presentation, which can be found below the above link. They were very positive.]

My point being, that it takes patience, persistence, and a passion for the subject of IT, to become good at IT, good at managing IT, such that it benefits the organization as a whole.





By the way, assuming the manager or exec has both the Ability and Passion for IT (see [“NotesOn: IT HR – Part I – Recruiting – Attributes of a Successful Hire”](#)), experience will take care of most of these characteristics ... *presuming* mentoring from seasoned management occurs. Otherwise, getting the “newbie” through to a “state of enlightenment” can be painful (and expensive) to the organization as a whole.

Alright, so what are some of the key “bloopers” that IT management newbies manifest? Let’s take a look:

“It is NOT Overcomplicated ...

... I have very good reasons for all these boxes ...”

1. IT is complex enough, there are more than sufficient moving parts, without adding to the “apparent chaos”
 - a. The simplest way to create havoc in IT is to create a convoluted organization chart with plenty of redundant roles, positions, titles and a whole host of “dotted line” reporting structures.
 - b. Eventually no one is sure who is responsible for *anything*.
 - c. Soon enough the only reason anything at all gets done is because certain dedicated individuals have bored tunnels through the chart and found other individuals who care enough about IT to take on a responsibility that may, or may not be, theirs.
2. Such structures are unsustainable and collapse in due course, requiring “yet another” re-organization. [See the new post: [“NotesOn: IT HR – Part III – IT Organization Models”](#).]
3. Another way to overcomplicate IT is to never define roles, never provide job descriptions, and never ever provide advancement pathways
4. The immediate remedy, per [“NotesOn: An IT Quick-Fix Repair Plan”](#), is: Stabilize, Simplify, Standardize.

“We don’t need more organization ...

... we just need to get it done!”

1. This characteristic is almost the converse of, the opposite of, the executive who overcomplicates IT. You can over-simplify too. Some logical structure is required. Basic policies, procedures and processes are beneficial ... so long as they promote quality production ... and the well-being of the IT team members ... a balance between the two is required.
2. The “just do it darn it!” management style manifests itself in a number of ways, for example:
 - a. shooting from the hip when it comes to task and project assignments; regardless of previous assignments and workloads
 - b. doing constant end-runs around lower management teams, going direct to the team members – okay in an emergency, not workable as a guiding principle
 - c. hidden organization charts ... kept out of sight for “security” purposes or some such argument; leaving no one aware of who is responsible for anything
 - d. failure to provide even an outline of job descriptions, leaving execs and managers free to define and re-define responsibilities “on the fly”
3. This is such a chronic problem that I’m surprised several books haven’t been written on the subject (I have an upcoming post to help out until then). When you see this attitude in action, here are some questions to ask:
 - a. Can you even find an organization chart?
 - b. And if you can, is it up-to-date?
 - c. If it is up-to-date, is it ignored?
 - d. If it is ignored, find out why and fix it.





4. In all fairness there is a condition where managers and executives get very frustrated with the entire concept of organization charts. And that is when ... the IT organization has gone through one change after another after another after another after ...
 - a. I know of one company that “overhauled” its organizational structure every two years, give or take a few months ... and we’re not talking “minor tweaks”. The result: constant chaos.
 - b. The suffering IT groups “got it done” in spite of everything but it wasn’t pretty, or efficient.
5. Just keep in mind that power is not generated by constantly changing the components making up an “engine”, power requires stability (and direction). Another way to look at it, as I noted elsewhere, is:
 - a. “Power isn’t generated by moving the windings, magnets and bases around all of the time. Power is generated with and from stability.”

“Forms, forms, we need ...

... more forms!:

1. Paperwork is not equivalent to production or even gathering metrics data (let alone managing), and yet, new managers at all levels have a tendency to want to design, build and promulgate forms ... lots of forms ... forms that gather any and all data remotely related to IT.
 - a. Why? Because, more often than not they are hoping that somewhere in the ever growing filled out stack is “the answer” to how to successfully run an IT organization or group.
 - b. While commonly rampant in new PMO groups, this tendency is not at all exclusive to them.
2. Selected forms are appropriate, such as Project Intake Forms and Work Order Forms and Purchase Request forms, and forms needed to document IT Audit Controls (SOX, PCI, etc.). But, as a rule, the “newbie” doesn’t stop there. I’ve seen entire departments tied up filling out task sheets and time sheets (both written and automated) and job sheets and daily reports and weekly reports and status reports of their daily and weekly reports and ... until significant portions of each day and week were being charged to “Admin”.
3. The amount of paper wasted on the documentation and management of unnecessary processes and procedures can be enough to drive even the most hardened anti-tree hugging paper-waster crazy. Huge volumes and binders have been filled with colorful action charts and process swim lanes and procedure diagrams and task descriptions and ... and ... and ... which if fully implemented, would have destroyed the entire IT group.
 - a. In one case, the “newest new processes” nearly did “sink the ship”, major surgery was required to bring the IT group back from the edge of collapse. In the end, every shred of related paper was recycled, along with all of the two and three inch binders used to hold it all.
4. On forms, go slow. Please don’t confuse forms with metrics – metrics measure production (or are supposed to). Before deciding to bury your IT group in paperwork, or put together any paperwork at all, do extensive “Look, Ask, Listen, Verify”. “NotesOn: IT Executive’s Survey And Checklist” (Parts I-IV) has a detailed list but find out:
 - a. What is working.
 - b. Why is it working (prevents throwing out the baby and the bath water)?
 - c. What isn’t working.
 - d. Why isn’t it working?
 - e. Who is working (productively).
 - f. Why are they able to get work done?
 - g. Who isn’t working.
 - h. Why aren’t they working (or what is preventing them)?
 - i. What customers / users are happy.
 - j. Why are they happy (and who and what is contributing to their happiness)?
 - k. What customers / users aren’t happy
 - l. Why aren’t they happy (and who and what is contributing to their un-happiness)?





5. Forms (should) facilitate production and the processes leading to production. If a form isn't directly related to production then don't bother (well, excepting certain legal and government forms, those you have to live with).

“Rubber must meet the road ...

... not tomorrow, now!”

1. Beware: what you promise, you have to deliver. So be careful what you promise, and when you promise it.
2. Too often new IT executives, in particular, succumb to so called user demands to “see something now!”
 - a. In many cases it's the executives not the user community making the demand.
3. Too often the excuse for short-cutting IT fundamentals is “They want it now!”
4. Such chronic cries are symptomatic of other problems in the IT group, and/or with the executive in question.
 - a. The worst thing an IT exec can do is hammer their IT team to get “something” out the door, immediately! Without any means for them to do so. No IT group, or any group, can operate in “emergency mode” for forever.
 - b. Much better is to examine and square away the processes (see “[NotesOn: The Four Fundamental Life Cycles of IT](#)”) that are impeding production. Remember: “Stabilize, Simplify, Standardize”.
5. A common reason for such demands is because the SDLC fundamentals prior to Build & Release were not done or were done so poorly as to be ineffective.
6. Another is that the Project Managers aren't trained and mentored as PM's. Echoed by one or more “managers and above” having no real clue how projects are successfully driven to on-time on-budget completion, either.
7. Fix your IT fundamentals and your delivery speed will come up. Demand “something, anything, now!” and you yank on the thread that will eventually tear apart your IT organization, as you are forcing them to deliver poorly designed, poorly built, poor quality products that do not do anyone any good at all.

“Upward relationship management is ...

... critical (for my next step up the organization chart).”

1. A concerted focus on only upward relationship management is a major signature of the individual who is “in” IT for the sole purpose of moving up or elsewhere. They manage up well enough, but rarely, if ever, down.
2. An effort needs to be made to educate this type of manager/executive on the value of IT to the business ... and to themselves.
3. If the individual just “doesn't get it” consider letting them go elsewhere (outside of the business) as they will have a difficult time ever integrating their operations/functions with IT ... and that will be expensive.
4. IT is not a “nice to have”, it is critical to the success of any and all businesses, no matter the model they use. So IT managers and executives must manage up *and* down *and* sideways for any sort of worthwhile result at all.

“I'm an MBA ...

... so I don't need your input”





1. But it is not just *some* MBA's with this characteristic. Others with "prestigious" alphabets after their name have the same issue and attitude. A variation is: "I don't care, it's my way or the highway."
2. This attitude is one of the most exasperating characteristics to live with in IT and is far too prevalent (though, fortunately not overly common).
 - a. An example was provided in "[NotesOn: IT HR – Part I – Attributes Of A Successful Hire](#)", of a particularly destructive senior manager / executive. If you haven't read this post, please do so. He exhibited ALL of the characteristics included here, but "I'm an MBA ..." was a "biggie" for him.
 - b. The attributes of a successful IT hire described in the above article apply, in full, to *anyone* managing any part of IT, high or low.
3. The best executives are those who know that they are there to serve, not to be served. These positions are not (should not be) about status and "a major salary". They are about the end game of successfully managing and growing their area of responsibility and taking *everyone* in that area or zone along with them for the ride. As a rule, when the job is done well the status (and "a major salary") come along for the ride.

"Yes!" ...

... to anything users/clients/customers ask for

1. As an IT Executive, if you hear of one of your new VPs, Directors or Managers promising the world to IT's users ... panic! And then pull them into your office and have a long conversation with them about "client *management*".
2. Managing a client has nothing to do with saying "Yes! we can do that!" to anything and everything they ask for.
3. Saying "Yes!" is ***not*** IT's job. Delivery of quality systems that make users / clients able to do "things" better, is.
4. Sometimes, "No." is the right answer. For example:
 - a. No, I am sorry. I've had my team look into it and, while it's a good idea overall, we can't do it that exact way. If we do, we'll create huge security holes in our network. But. If we do it this way ..."
 - b. No, unfortunately we can't deliver it on the 15th. We still have testing to do on the final changes you requested. We want to be sure we deliver a system to you that works for you. If you and your team have some time to devote to helping us test, we should be able to make the 30th.
 - c. No, please do not buy that software package. We've looked into it and it has major problems (bugs, security issues, performance issues, etc.) that will cause both of us no end of pain and trouble. What we suggest is ...
 - d. No, we can't fit that into this next release without delaying the release. It's a good idea but you've said it isn't critical so I've put in on the "Wish List" and we'll work it into a future release.
 - e. No, I understand what you are asking for, but, to be honest, it is too expensive to do it that way, here's why ... What I suggest is ...
 - f. No, I know it's time consuming, but we have to finish gathering at least the major and critical requirements *before* we start building the application. ... I know you are busy too, but how can we buy some of your time to make sure you get what you want and need? We'll bring the pizza and soda ...
5. If you want to see your entire IT group collapse from exhaustion and frustration continue to have your management team say "Yes" to every single client request. (I swear on a stack of anything you wish, I've seen this style of "client management" more than once. And it costs companies a LOT of money and lost good will.)

Let's move on to the Techies now.





Techie Newbie Characteristics:

I and others have observed over the years (I may even have exhibited one or two of them myself) that both new and pure techies (dyed-in-the-wool techies who have been “around the block” at least a few times) have most if not all of the following characteristics. I have yet to run across a single new IT techie who doesn’t exhibit at least some of these symptoms. The degree to which they “suffer” from them varies, but the characteristics are rarely nonexistent. We’ll use new programmers/developers as an example, but in many ways these can and do apply, directly or indirectly, to other technical roles as well, such as DBA’s, network specialists, business analysts, etc.:

They want to code, ...

... they want to do nothing but code:

1. They wish to be left alone ... to code.
 - a. Their attention span for anything else is very short.
2. They often have little desire to actually meet with the users/clients ... so they can code.
 - a. Face-to-face or by phone are often anathema, abhorrent, to them; e-mails/IMs they will tolerate.
3. They hate documentation, they will do anything to get out of doing it ... so they can code.
 - a. This includes commenting their own code, especially commenting their own code – after all, their code and logic is “obvious”.
4. They hate meetings, they will do anything to get out of going to them ... so they can code.
5. They hate after-implementation support, they will do anything to get out of doing it ... so they can code.
 - a. Even when it is their own code.

They are easily mesmerized by ...

“new technology” and “new features”:

1. They are eager to try the latest “cutting edge” software widget and are quite willing to “bleed” all over the system, without regard to (someone else’s) time and/or cost.
2. They know, or soon learn, that there are *at least* six ways to do anything in any, even half-way decent programming language or IDE (Integrated Development Environment) and they will do their level best to try them all, often in the same module and/or application.
3. It doesn’t matter if some or all of an SDK (software development kit) is in beta and has known bugs, anything new is automatically better.

Their Coding Standards ...

... are their own.

1. They will often attempt to “optimize” their code to the “nth” degree of coding perfection:
 - a. One way is to “nest code” (functions within functions within functions ...).
 - b. Another is to “short hand” names of variables and use obscure very brief (thus meaningless) aliases.
 - c. A third is to make nearly every bit of program logic an external function of some sort, even when it will only and ever be used once; carrying object oriented principals to an unnecessary extreme. Carried too far, most often the case, supporting, modifying and debugging such code becomes nearly impossible as it first has to be deciphered.





2. They will often attempt to “tune” their code in an effort to shave off one-ten-thousandths of a second, or more ... when doing so is wholly inappropriate.
 - a. One developer quit because I wouldn’t let him spend three days “tuning” a piece of rarely used data retrieval code. He went to work for a Bank and was quite happy spending all his time tuning.
3. Unless they have slammed up against a bug that they just can’t figure out, if the known issues aren’t “fatal” they still want to push their new app or changes into Production, preferably now.
4. They have little patience with waiting for Version X+ or Service Pack X+ that contains critical bug fixes.
5. They assume that once they’ve found a bug that there are no other bugs in their code and so insist on pushing the “fix” into production, now.
 - a. Bugs can reside alongside each other or underneath one another (the one above masking one below it that can mask one below that can ...).

They want to slip in *their* new ...

“features” and latest “improvements”:

1. Whether asked for or not, whether they’ve asked or not. They wish to do so (and often will do so given the opportunity and lack of discipline) because to them the new whatever it is seems “cool” or “cooler”.
 - a. This is a scope killer, one of the contributors to “scope creep”, in some cases “scope explosion”.
2. And it doesn’t matter if the system has already been tested and approved and readied for Production.

They often (almost literally) abhor ...

... governance of nearly any kind:

1. Particularly obnoxious are the Architectural Standards and Security groups, Project Managers are next on the list.
2. They have no interest in being Tech Leads or Project Managers or Managers of any kind or stripe, or learning anything about same.
 - a. At best they tolerate interference from Tech Leads and Project Managers; even though Tech Leads are “techies” themselves, they too are considered distractions from coding .
 - b. The exception is when the Tech Lead is considered, by them, a “technical guru” and even then it can be dicey.
3. They have little patience with P(M)LC’s and SDLC’s .
 - a. Particularly if they prevent them from coding right away and necessitate requirements and design documentation ...
 - b. And don’t even ask about their helping with User Help docs.
 - c. They are sometimes referred to (derogatorily) as “cowboy coders” or just “cowboys”
4. They will spend hours and hours trying to resolve a bug rather than ask another team member for help.
 - a. I enforce the “two hour” rule, if they can’t figure it out in two hours they *must* bounce if off someone else.

They see no point in testing ...

... more than they already have (if they have):

1. They figure that they “don’t write bad code”.
 - a. They believe, that they wouldn’t have written it the way they did if it wasn’t the right way to do it.





- b. This is one of the primary reasons why developers are the worst testers in the entire known universe.
2. They have the novice engineer's mentality of "if the user would just do it the way I wrote it everything would work just fine".
 - a. For years I've called this: "engineering, or engineer's, software".

They often have Prima Donna ...

... attitudes:

1. All of the better and best techies have, to one degree or another, "a rather high opinion of themselves"
 - a. While it may be somewhat deserved, their attitudes often prevent them from being team players (it requires even handed guidance from an experienced manager to make them so), and,
2. In extreme circumstances, they may demonstrate "my way or the highway" mind-sets that prevent them from listening, particularly when it is being demonstrated that "their way" has "issues"
3. This attitude can extend to a nearly belligerent unwillingness to do customer support / sustainment; even of their own applications.
 - a. They want to code, they want to code "new", they want to code with "the latest and greatest".
 - b. Maintenance is the farthest thing from their "ideal world" that they can imagine ... well, next to hating documentation of all types.

Summary:

While the above Management and Techie characteristics *can be* seen as strengths by some, and may be developed into true strengths, for the most part they *are* weaknesses. As a result, newbies at all levels *require* management and constant mentoring in order to become reliable, useful members of IT. These flaws will not usually resolve themselves.

The antidote for "new IT Guy" characteristics is: discipline, communication, and mentoring. When applied you will eventually produce a stellar team member. IT is an engineering discipline and deserves the respect of same.

Finally, I hate to bring it up, I probably shouldn't bring it up, I'm sure I'll catch a lot of flak for it, but nearly all of the above new Techie characteristics are reflected in the fundamental principles of Agile methodologies, i.e. it seems to me that Agile is an harmonic reflection of what "IT Newbies" and "Pure IT Techies" like best (though it is not best for IT).

There is value to Agile as an SDLC approach, it *has* pointed out "the obvious" in a number of cases – obvious aspects of running an IT project missed by, you guessed it, newbie PMs and Managers. But Agile must be tempered with common sense and experience to be useful and successful. Validating and reinforcing the above characteristic flaws is not helpful in the long run to IT, or to IT techies.

Hope this helps,

DP Harshman

PDF Link

