

## NotesOn: IT Fundamentals – Simple Defined

### ***Introduction (v1.1):***

In various articles for this website I have used, and will continue to use, the word “simple”. But using it without clarifying what it defines is a fundamental error, as “simple” must be understood within the context of its subject. It must be appreciated and accepted (or not) within the framework of a discussion area.

There is an almost universal acceptance of what “simple” means, e.g. easy enough for an eighth grader to understand, straight-forward enough for anyone to assemble (hah!), and so forth. But. What is considered “simple” in the arena of nuclear physics does not equate to “simple” in the zone of, say, “rocket science” or level set with “simple” in terms of the management of, and working in, an IT organization. However. That statement still doesn’t define “simple” sufficiently well so one can recognize it when one has gotten there. Or, more importantly, why the word “simple” is at all important.

|                                    |          |
|------------------------------------|----------|
| <b>Introduction (V1.1):</b>        | <b>1</b> |
| <b>Background:</b>                 | <b>1</b> |
| <b>The Fundamentals Of Simple:</b> | <b>3</b> |
| Simple 101:                        | 3        |
| An IT Example:                     | 5        |
| <b>Summary:</b>                    | <b>8</b> |

### ***Background:***

What triggered this post (in the middle of working on the upcoming “IT Organization Models” post) was an article from “Herding Cats” entitled “[The Malformed Criticism of Power Point](#)”. Odd title. Interesting post. Especially if you’ve been following the others written by Glen B. Alleman (who, in my humble opinion is one of the top-drawer project management gurus around). And. I agree with the post. ... With one exception.

A repeated theme in a number of Glenn’s posts (apparently one of his pet peeves) is the use, overuse, of the word “simple”:

“... Simple is a term tossed around without context or domains.”

That I agree with. But. He then refers the reader to the following quote as part of making his case:

*“For every complex problem there is an answer that is clear, simple, and wrong.” - H. L. Mencken*

The quote I do *not* agree with. Definitely. Absolutely. Not. (In fact, his use of the quote is my pet peeve.)

True, it doesn’t take too terribly much in the way of study to come up with real life examples of when and where the word “simple” is overused, abused and generally misapplied. However.

By my way of thinking, Mencken's quote is far too cynical and far too absolute to be of practical use. It is a highly opinionated statement, at best, that may only occasionally represent a demonstrable fact, (a case of "all generalities are false, including this one").

One of the reasons why Mencken's quote irritates me is because superficial, uninspected, use of it provides the "perfect excuse" for *not* doing "due diligence", i.e. not taking the time to, not making every effort to break a problem down into, what I call, "consumable components" (or elements).

Glen is right, "things" can be oversimplified to the point where all information of value is lost. But. Such abuse should not swing the pendulum of understanding so far over to the other side that "simple" is obliterated from the dictionary. The Mencken quote requires both inspection and challenge for this reason.

Effort is required to break a complex subject down into elemental statements and concepts. The upside is that when done right and when they are strung back together they will represent a coherent picture of, and provide answers and solutions to, the original problem. What Mencken's quote seems to ignore is that those elemental components are, and must be, simple. If not they should be worked over until they become so.

When done right, the process is called: analysis. When done wrong the process is called: a waste of time.

[The difficulties folks have with this subject represents, in my humble opinion, one of the flaws in our current educational system: the lack of training sufficient to do even an elementary analysis of a complex problem. True understanding at least requires the ability to shred a topic down to its most fundamental, most consumable, components – pros and cons, benefits and risks, pluses and minuses, etc. – with the goal of reaching rational defensible decisions. Without that skill (though it need not be at a level sufficient for an individual to function well in IT), an individual is doomed to a life of mediocrity ... and manipulation by others.]

Let's move from slightly esoteric theory to the more concrete zone of "get it done".

There is close to a 100% chance that a new IT project manager's first "full bore" project seems, to them, both daunting and horribly complex. In sharp contrast to an "old hand" (an experienced pro), when a "newbie PM" attempts to analyze the whole of a project against an as yet unknown set of steps that "hopefully" will lead to a yet to be defined "successful" conclusion at some yet to be determined time in the future, they naturally have serious second thoughts and doubts about their decision to become a PM.

I don't care how much class-room training and book-learning they've had, for the "newbie PM" there is no such thing as "simple". And. Sadly. If they ever read Mencken's quote, above, they most likely absorbed it whole heartedly as absolute truth. When it is no such thing.

To the newbie PM (or Manager or Exec): It ALL seems confusing! Every last bit of the project. Such that a certain level of panic sets in. Until they are sure it will only get worse before it ever gets better; if it does. [This state of affairs is one of the triggers for, root causes behind, the "Forms, forms, we need ..." characteristic describe and discussed in ["NotesOn: IT HR – Part II – Characteristics of IT Newbies"](#).]

Unless!

The newbie is being mentored by an experienced PM, or Manager; or they have been educated at least well enough to know the basics of how to break the “whole” down into its minimum necessary “consumable components”; or both. Then. It gets “simple(r)” and the beads of sweat on the forehead and under the arms begins to reduce. Gradually, experience and success increase the new PM’s confidence and competence level until “suddenly” it all seems “natural and easy”. But, how to get there is the question at hand. Here is how.

### ***The Fundamentals Of Simple:***

Glen is correct in that when speaking of “simple” one cannot, must not, discount the frame of reference, what he refers to as “context and domain”. The frame of reference is the subject’s environment, the knowledge required to understand the subject’s environment, the skills required to utilize and apply the understanding of the subject’s environment (once obtained) and the required experience to know when and how to apply it.

The word experience assumes that a gradient, step by step, learning process has, via one method or another, occurred; beginning at the beginning (i.e. starting with “the basics”, the fundamentals) and then building on it in order to scale higher onto the tower of acquired knowledge. Not just knowledge but knowledge that can be applied. For knowledge without (eventual) applicability is a waste of time and effort – other than, perhaps, for amusing and amazing ones friends and acquaintances at a party or trivia contests.

One of the skills required in IT is the ability to take complex problems and situations and break them down into “consumable components”. Not everyone can do this well, some not at all. This skill can be improved upon with practice, but for best results the native ability to “see” the whole picture and mentally break it down has to be there. Then, the essence of good project management (and IT Mgt as far as that goes) is the honing of this skill, for the ability to easily take apparently disparate, sometimes confusing, data, processes and requirements and come up with a coherent plan ... that leads to a product with a True ROI ... is *the* singular skill that separates the pro’s from the newbies.

It may help to demonstrate the meaning and usage of “simple”, within a technical frame of reference, if we step outside the “IT World” for a moment and walk through an example from another field, thus setting aside any “baggage” associated with IT.

### **Simple 101:**

Take the case of building a bridge, over a river – that’s pretty far removed from IT. And the question is: what would you as a civil engineer need to know, what would you have to do to build it? As a “newbie” engineer such a project might seem daunting. No, make that: would be daunting. Everything you ever learned in school about building bridges would start swirling around inside your skull to an almost overwhelming degree; certainly enough to keep you awake at night. Not only is a great deal of someone else’s money at stake, but other people’s lives as well ... if ... you don’t do a better than “good enough” job. Of course, this is where experienced engineers would step in to help out, but you needn’t tear your hair out while waiting for them.



The engineer's definition of "simple" is: each element of the problem broken down into ... bite-sized "consumable" pieces of useful, usable, understandable information ... that helps resolve the whole.

You can't tackle everything at once. The answer, complete with color images, blue prints and spec sheets on how to "build the bridge over the river", won't leap into your (or anyone's) skull in one giant "whoosh!"

So. What DO you need to know? What DO you need to do? Here's a starter list of what you must find out:

1. Why the bridge needs to be built = the business driver
2. Hoped for, anticipated, Budget
3. Desired life span of the bridge (usually in decades)
4. Potential locations for the bridge – high level
  - a. survey of population centers, etc.
  - b. shortest distances versus potential optimum building locations
5. Initial on-site survey work (for each potential location):
  - a. The average width of the river
  - b. The average depth of the river
  - c. The average speed of the river's current(s)
  - d. The approximate volume of water per hour (for force calculations, etc.)
  - e. The seasonal variations in the river during:
    - i. Summer
    - ii. Winter = does it freeze? Are there ice flows?
    - iii. Spring = floods, detritus (logs, brush, rocks, sediment) carried by the river, etc.
    - iv. Fall = lowest levels?
    - v. Silt loading for each season (the amount of dirt carried by the currents)
  - f. Type(s) of water
    - i. Fresh
    - ii. Salt
    - iii. Both
  - g. River traffic
    - i. Commercial = if so, what size boats / ships and how many?
    - ii. Recreational = ditto
  - h. Soil / core samples from both sides of the river to determine composition of and depth to the type of bedrock (if any)
  - i. Ecological impact study – what affects will the bridge and the traffic have on the surrounding environment
  - j. Utilities -- what if any utilities will need to be included in the bridge design = electricity, gas, potable (drinkable) water, etc.
6. Regional survey (for each potential location):
  - a. Existing infrastructure
    - i. what utilities are immediately available for the construction and operations phases
    - ii. what roadways exist, if any
    - iii. what housing (hotels, apartments, etc.), restaurants, fuel supplies, etc., etc. exist
  - b. What potential construction materials are in the region
    - i. Steel mills, suppliers
    - ii. Wire cable manufacturers / suppliers





- iii. Pipes
- iv. Cement
- v. Etc., etc.
- c. What skilled labor pools exist in the region
- d. Weather
  - i. Historical weather patterns
  - ii. Projected weather patterns = 50 year, 100 year storms, etc.
  - iii. Construction season (typically late spring to early fall in most regions)
    - 1. Minimum
    - 2. Average
    - 3. Maximum
- 7. Anticipated traffic (the “should be”)
  - a. Volume – current and projected
  - b. Types – car, trucks, motorcycles, foot
  - c. Type / weight limitations
  - d. Daily traffic patterns
- 8. Possible traffic (the “could be”)
  - a. The “Fudge Factor” elements, i.e. what might have to cross the bridge despite “restrictions”, and how often
- 9. Etc.

By breaking the “problem” down into distinct “consumable”, i.e. understandable, elements that can be accurately described you *begin* to gather a picture of what is required and what is possible within the project’s “triple” constraints: scope, time (schedule), cost (resources), quality, risk, customer satisfaction. Until you have broken the bridge problem down, until you have done your homework on each element (and its interactions with each of the other elements) you won’t have a clue how to proceed and your confusion, uncertainty and lack of confidence will continue to exhibit itself to one and all. Once you’ve done your initial homework, then, and only then, can you (and should you) *begin* to consider the actual design of the bridge.

**Rule: the essence of good design is taking into account what is and what should be while allowing for what could be.**

Does that help? ... Good. ... Then let’s get back to IT.

### **An IT Example:**

Let’s say a business owner comes to you and says: “I need to automate my distribution business, we have too much “paper” around, the inventory is never accurate, our current processes are too slow ... we need to become more efficient and cost effective if we’re to stay competitive.”

You might be saying, under your breath, after reading the above statement of need: “Heck, everyone is on computers these days, everything is “automated” so that’s a pretty lousy example Mr. DP Harshman!”



Perhaps. ... Then again. ... Perhaps not.

First off, it isn't factually true that "everyone" is on computers. However, if I grant you that one it still isn't true that everyone is using a computer system(s) that is doing what *they need* it to do for *them*; systems that help them do "it" better whatever "it" is [ref: "[PPT: IT Fundamentals Supersede All Methodologies](#)"]. You might be surprised at how many businesses run on makeshift "systems" that in some way fail to meet the definition of an automated set of fully integrated software applications providing a True ROI. Too many, businesses "survive" on Excel spreadsheets (maybe attached to Access databases), and websites cobbled together by someone. This is also true for business units of some well known "ginormous" corporations. [To a large degree true ERP and CRM systems are still a myth and where they exist at all the business most often had to drastically change the way THEY successfully did business in order to use the system.]

So. Please. When given such a task as above, don't fall back on "everyone \_\_\_\_\_" (fill in the blank). You have to, you must look. You have to look at that situation, you have to walk the walk of that company. You need to reach a point where you know their business (or that portion of their business) nearly as well as they do. Remember the Executive Mantra from my earlier posts? "Look, Ask, Listen, Verify"? This mantra applies to *all* aspects of management, including project management.

What you need to do is break the problem down, much as we did with the bridge building example above. Each of those elements was, by itself, simple. When combined they will again result in a complex system; bridges are fairly complex organisms. But. Each element should stand on its own as a precise, simple, entity. If an element is "too complex" you haven't broken it down sufficiently, you haven't named the problem sufficiently well and done your homework to the degree necessary. If even one element is too complex, and you ignore it, you have opened the door to a future failed system.

Years ago I designed a full bore stock options system with a third party package at the very core of it doing the "grunt work". While in the design process I came up with a "potential" Use Case that my boss said had, at best, a million-to-one chance of ever occurring. Nevertheless, I added code to account for this condition and darned if three years later it didn't pop up. Had I not recognized that case, that consumable element, we would have spent days, if not weeks, tracking down the cause of the incorrect options calculations.

**Rule (general):** If something doesn't make sense either (a) it is insane [true insanity is, by definition, neither sensible nor logical nor understandable], or (b) data is missing [which may give the subject the appearance of insanity], or (c) all the data is present but something about it is not, yet, understood.

**Rule (IT):** If a requirements and/or design element doesn't make sense, there is still something(s) you don't know or don't understand, or both. The correct response is NOT to ignore it and go on (maybe coming back to it later) but, rather, to research and clarify it. When you clarify it, it will simplify.

At the signature instant a complete clarification has been obtained, when a proper simplification has occurred one often hears an: "Ahhhh-hahhhhh!" After which one can work with, and think with, the element.

The next thing to know about "simple", is that you may need to break a problem down by layers. In many cases you cannot assimilate the entire concept in one pass, or even two. Most often you have to lay it out at a

high level, break it down further at a mid-level, then drill down to a base (functional) level, in such a way that all of the levels cleanly align – vertically and horizontally. In some cases it may require more drill down levels.

Remember the rule from my PowerPoint on “[IT Fundamentals Supersede All Methodologies](#)”?

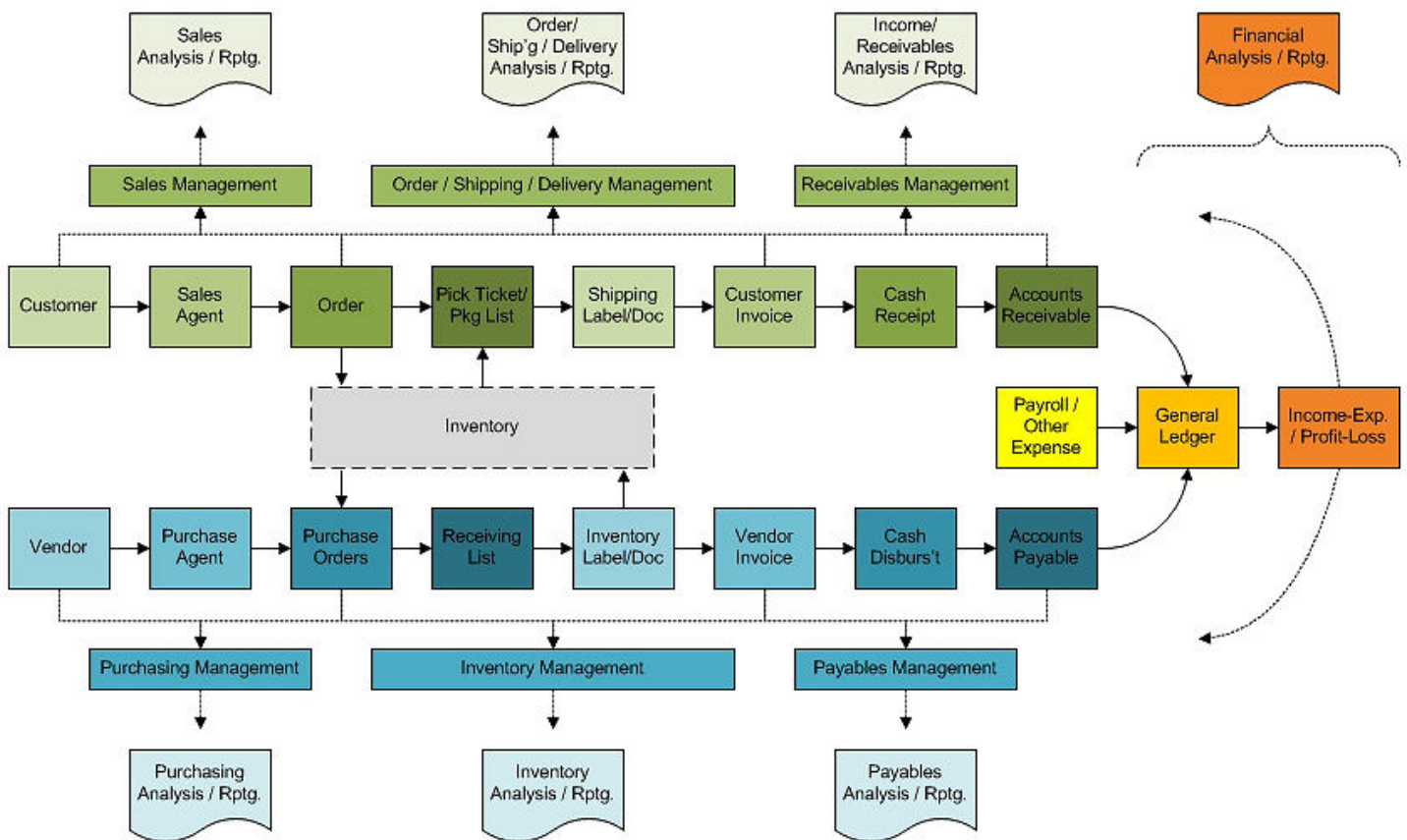
“If you can’t draw it in two dimensions you can’t build it in three.”

When evaluating and analyzing a business, or portion of a business, you should be able to (when done) clearly draw out the flows of that business or business process ... including the exceptions. These days everyone does multi-color Use Case diagrams with stick figures and images of networks, clouds, desktops, servers and so on. Earlier, black and white diagrams with highly specialized symbols were (sometimes) used—I still have my old plastic IBM style mainframe process template around somewhere. But, the truth is, you can do just as well with a series of boxes that show the flow, detailed progress, of a process from beginning to end. For example:

Years ago I developed a high level chart for distribution companies. The goal was to show how each of the modules of their business and the proposed system interrelated. That chart looked something like this:

### High Level Distributor Business Workflow Diagram

© DP Harshman – www.fromtheranks.com



Does this process flow make sense to you? Can you easily, simply, identify each of the business process areas and understand how one feeds into (is input for) another and then others?

If you are thinking: “That is *too* simple, no one could *build* anything from that!” ... you would be correct.





Because.

You aren't done yet.

Once you've completed such a high level analysis you next need to take each of the boxes and break out the individual processes within them; doing a more detailed analysis of both their "business as usual" and their "exception" functions. ... Never, ever forget the exceptions, they will "kill" your project if you do. Then:

Repeat until down to the "consumable components". Once there, work all the process flow diagrams up and down (between the various levels), front to back, side to side, corner to corner, and so on, to ensure not a single "snag", "oops", or "uh-oh" is left behind. When complete, you will have a, still flexible, road map you can rely on. [Note: you *could* stuff all of the "boxes" into one gigantic extraordinarily difficult to comprehend diagram, but there wouldn't, normally, be any sense in doing so ... unless your goal is to *create* confusion.]

A complete set of process flows allows you to sit down with your users and intelligently walk their data inputs, data transformations, and data outputs from one corner of the business to the other. And that! Not chrome or iterative iterations. My friends. Is what builds confidence in the hearts of your users. And. Builds great systems.

Special Note: Any process – business, personal, governmental, military, etc. – can be broken down in this manner. *Any*. But. To be fair. There may be components that you can't seem to fully nail down, that you can't, or can't easily, define to an "Ahhhh-hahhhh!" point. When dealing with humans and human situations there are always elements of unpredictability which don't fit into neat process steps –as exemplified at certain points in the diagram in the above referenced Herding Cats post. When these occur, and they will, you need to bring to bear your Risk Management tools. ... (Did I hear an "Ahhhh-hahhhh!?)

## ***Summary:***

Again. As much as I admire the Herding Casts site, and most of the author's hard hitting truths about project management, I have to singularly disagree with the quote he has offered on several occasions:

*"For every complex problem there is an answer that is clear, simple, and wrong." - H. L. Mencken*

Again. This, by my way of thinking, is far too cynical. Not to mention more incorrect than not. Plus. It provides an excuse for not doing "due diligence", not taking the time to, not making every effort to break a problem down into its "consumable", or "risk management required", elements.

Hope this helps,

DP Harshman

PDF Link

