

Re: Computer time -> Developer time -> User time?

Source: <http://coding.derkeiler.com/Archive/Assembler/alt.lang.asm/2005-01/0831.html>

From: Beth (*BethStone21_at_hotmail.NOSPICEDHAM.com*)

Date: 01/22/05

Date: Sat, 22 Jan 2005 17:34:32 GMT

Frankie say:

> *Randy wrote:*

> > *Beth wrote:*

> > > *...hence, tactics used to maximise programmer "productivity" came to*

> > > *the fore: "portable source code", "black box libraries", etc....this*

has

> > > *lead to "bloatware"...*

> >

> > *No, the thing that has lead to bloatware is the insistance on the*

user's

> > *part that their software have more features than the "Jones' software".*

But isn't that an industry "self-fulfilling prophecy" rather than a true "user demand"?

You know: Swamp the stores with "version 2"...brand new users can't buy anything else because they don't stock anything else...these move to "version 2"...the stores threaten others with "it will be obsolete! Obsolete, I tell you! Change now!" (in a grim-reaper-dressed "Ghost of Christmas Future" kind of way: "Buy version 2 for tiny Tim, Scrooge, or this grave you see before you shall have your name carved into it!" ;)...people "panic" and move to "version 2"...well, lo and behold! Now that everyone's been "herding" and "panicked" onto "version 2" (and the store isn't even stocking "version 1", so, even if you wanted it, how can you get it?), there's few sales on "version 1"...they discontinue it and make it "obsolete"...then a year or two later, repeat the cycle...

A demonstration of this is Microsoft's "5 year rule"...basically, Microsoft retract "support" for each of their OSes, 5 years after release...

Now, first, if this is a "reaction" to user "demands" then how on Earth can they declare when it'll be "obsolete" BEFORE they've even released it! The way the "5 year rule" works is simple...if Longhorn is released this year - 2005 - then it'll be officially "obsolete" (and Microsoft then refuse to "support" it in any way) in the year 2010...just add 5...as simple as that...

alt.lang.asm: Re: Computer time -> Developer time -> User time?

It's a "prediction"? Then it's not a terribly good one because, for Windows 98, they've had to break the "5 year rule" and extended the "support"...why? Because, completely contrary to this "users are in a mad rush to upgrade, upgrade, upgrade!" theory, far too many people WEREN'T upgrading from 98...a sufficiently large number weren't upgrading to be significant to Microsoft to "extend" the "support"...the retraction of "support", in truth, is a "threat" to force "constant forward motion"...if everyone refused to "upgrade", then they'd be forced – as in the Windows 98 example – to stay "supporting" it, otherwise they'd be cutting off their own nose despite their face...while 90% of users are still on Win98, they'd "retract" all "support"? No, they wouldn't...they'd be risking their "monopoly"...

Sir Bill has used this "poker bluff" repeatedly to make himself a billionaire...developing software for Apple (for their original Mac) ready for "launch date", Bill used another "threat": We will not provide you any of the software...Apple panicked at a "launch party" where they couldn't actually show the machine doing much of anything because there were no software applications and they didn't call his "bluff"...worse, part of the "extra terms and conditions" Sir Bill demanded? "Some" rights to Apple GUI technologies...

He "threatens" retracting his software and then people panic that it might disappear and leave them in a terrible situation so "surrender" to his demands...but, really, people should start calling that "bluff"...a software company that refuses to sell any of its software? You can't keep up that "spoilt brat" attitude for long...unfortunately, Bill will keep getting away with it because most people are not "wise" to his "poker bluff" tactics...and there's no way to get this message across and for everyone to believe it, in order to get the whole world to "call that bluff"...so, he'll get away with it...

Users DO want better features...but they don't want "random, completely arbitrary features, so long as there are numerically more features than before"...

The "feature bloat" is caused by one thing...and it's not users "demanding" it...the industry wants to make profits...if it did this in the more "honest" way, then they would release a good application and then the next upgrade may not happen for many years (and the "new version" isn't simply, unlike Windows, a case of "better graphics, USB support and that's it" – after all, why exactly do you need to buy a boxed copy of that? Microsoft can upload the "new bitmaps" onto "Windows Update" or a few DLLs for the "USB support"...it could all be done by "Windows Update" but that's free and they want to make money...so, they draw a "line in the sand" and it's at those points they change the graphics to make it look "brand new" and sell it under a new title with a new logo – but an actual fundamentally significant improvement of functionality)...

But that doesn't make too much money...so, they turn what should be like buying a house into what's more like buying your weekly

Re: Computer time -> Developer time -> User time?

groceries..."threaten" things going "obsolete" ...bring out "new versions" that aren't appreciable that different from what you've already bought (let's look at it...Windows 95...+ USB support + IE built-in...= Windows 98...umm, did ME have ANYTHING new to it at all from 98? All superficial, anyway...ME -> XP? An illusion, really...XP is simply 2K "in disguise" (though not disguised too well when 2K binaries are totally compatible and it can even be found identifying itself as the same as 2K)...hence, really, they are just "herding" the "home" users over to the "business" NT kernel and have just added "backwards compatibility" support...of course, this "support" isn't a "new feature" at all...it's a "must make sure there's no barriers to forcing people to upgrade to XP" feature ONLY to support their "forward constant motion" plans...if you didn't "upgrade" then you would need no "backwards compatibility feature"...in short, just like the ten years before 95 (where they remained "16-bit" far too long - on the "home" front (the fact OS/2 and NT weren't like that shows they could have it...because, in fact, THEY DID! But only sold it to "business" ;) - despite the hardware having been perfectly capable for a decade...this also sheds light on "commercial software develops faster"...not by Windows standards, it doesn't! It takes them 10 years to write a new OS...that easily sounds like less than "hobbyist" rates of production (accounted for by the "extra work" needed to create the in-between "false progress" versions ready for release)...the "versions" in between have all been superficial modifications to make it look "different" for the purposes of duping users into believing they aren't doing what they actually are doing: Paying for something they already own)...

Users don't care for "technology for technology's sake" (well, "geeks" do but they are a minor fraction of the "audience" here)...users just want to get their work done...in fact, the less "nonsense" from the OS in doing so, the better (features that "enable" work to be done, great...features that are never used and "bloat" and slow it down and get in the way of doing work because it's all far, far too confusing: Nope, no-one wants that...absolutely no-one...even Microsoft would wish that upon themselves...they put it in because it "justifies" the next version so they can sell it to make profits...as simple as that)...

You're usually the one with the statistics, Randy...what are the statistics about USER PRODUCTIVITY since all this "feature bloat" has been increasingly introduced? Well, lo and behold, it's getting WORSE, despite the bold advertising claims, used to push people onto the new version...there's "memetic reality" - a constructed self-delusion of "how we would like things to be" - and then there's the real thing...a "connection" between the two, apparently, is not always necessary when "profits" are involved ;)...

It is an industry "self-fulfilling prophecy" to keep pushing this "constant forward motion" hard because you can double, triple your profits by selling some "technology" in "baby steps" over a decade with minor "version enhancements" rather than just sell a new version when there is a significant, fundamental change and improvement to the product...this isn't merely a personal opinion, it has been investigated as charges in law

alt.lang.asm: Re: Computer time -> Developer time -> User time?

courts against Microsoft (typically, the ones that ended up as "out of court settlement" that we never heard the court's verdict on this practice)...

Users "demand" it? Well, they "demand" getting something worth the money they hand over...they are EXPECTING actually better software...what they are typically supplied is an "illusion" of this...and all the "extra features" are a "blind them with science" tactic (because most users don't really know what 90% of these claims on the side of the box actually refer to), to "convince" them that they really are buying a "significantly improved" product...once bought, though, actually listen to them voice their opinions on what they bought...if you don't outrightly "hate" Microsoft, then you're certainly "disappointed" in what they do...I have met ONE person who defended them and was enthusiastic (who suspiciously has an incredibly detailed knowledge of NT kernels, that one wonders if he's an ex-employee or being "hired" by Microsoft to "be on the look-out" for the "next Linux" developing on the OS development groups ;)...and challenging that enthusiasm demonstrates no actual solid logical base for it, anyway...indeed, comes across more like some kind of S&M "fetish" or something...enjoying being pissed upon by contemptuous Microsoft policies)...

One has to be careful of "industry theories" from the perspective that, for instance, the industry is grossly unlikely to suddenly say: "you know what, we should make ourselves less money because, really, it isn't justified"...or a developer declare: "There's too many developers here...so you should fire me in order to do things properly"...money is a powerful motivator and it motivates as many (if not more) LIES and "self-delusion" than it does "competitive progress"...it has to work this way, of course...but, as always, the presumption of something that's 100% "advantages" and 0% "disadvantages"? Not anything that exists in this universe, sorry...money isn't the root of all evil – that IS an exaggeration – but it's certainly not the root of "only good" either...

The industry as a whole might not have quite the same levels of "contempt" that Microsoft specifically do...but we're all "motivated" in the same direction, so to speak...indeed, Microsoft are "king of the pile" almost certainly because they were the ones without "conscience" or "guilt" to push it all the way "to the max"...right up to 11...an unfortunate "strange attractor" in the system we build for ourselves, it does actually favour "bastards", so long as they are smart and devious enough to "get away with it" (because so long as they remain legal "bastards" – and even then, money can "oil the wheels" that various things are "overlooked" – they are allowed to be that way as much as they like)...

> *I don't doubt that this is true, but I don't think it tells
> the whole story.*

It doesn't...now that Randy has brought this "jigsaw piece" into the picture, it brings the topic around for me to introduce another "jigsaw piece" above that not all of this "progress" is actually "demanded" by

Re: Computer time -> Developer time -> User time?

alt.lang.asm: Re: Computer time -> Developer time -> User time?

users but is rather "persuaded" for "constant forward motion" because, simply, you only make money when you SELL the software, so you want to be selling as much software as possible...a simple "trick" is, therefore, to basically sell the same thing over and over with "minor modifications" and "more features" – pretending in the glossy adverts that it's some kind of "revolution", not "minor evolution", that you're selling – to keep generating that money...the "dealers" also make their money that way, so they support you on the other end of things in ceasing to stock the older software and to "push" the newer software for you in the stores...

Not that this is any "conspiracy" (well, looking at Microsoft's "Halloween" documents, perhaps it IS a "conspiracy" in their case ;)...it's just the natural "strange attractor" to this "chaotic system"...that is, dealers make money selling software, developers make money selling software...marketing agencies make money pushing the sale of software...it's all pushing hard to "sell software"...the user is "swamped" by various "constant forward motion" tricks (sometimes, of course, they are willingly going along...I mean, sometimes the software developers actually DO have something worth paying money to "upgrade" to...but, often, this is a "forced" thing...one needs to not solely "look at the figures" but also "sample" the user opinions voiced...you know, sure, they bought it but did they actually LIKE what they bought? It's not solely programmers and "technical" users who voice anger (and sometimes hatred) at Microsoft or joke / complain about the "constant push forward" tactics and that they DON'T feel they really are getting much of anything that useful out of it :))...

You know, if I set up a "system" where you get \$50 every time you jump in the air, it's not unusual to find big crowds jumping up and down...not an actual indication that jumping up and down is a "good" or "right" thing, it's just a way to make a living under this kind of "system"...it's a "strange attractor", so to speak, to borrow terminology from "chaos theory"...systems inevitably have their "natural attractors"...things tend to "settle down" in certain "patterns" (oh, and the word "strange" only refers to the fact that in "chaotic" systems, these attractors aren't necessarily "simple" or "obvious"...put them into a graph and the shapes can look rather "strange"...hence the name...with a "simple" non-chaotic system, the "attractor" might be obvious...some "all the 'virtual ants' following these simple rules, all end up gathered together in the top-left corner of the screen" conclusion...you can even work that out pretty easily from looking at the "rules"...but with "chaotic systems", the same thing inevitably happens, it's just not "simple" and the "shape" of this "attractor" not "obvious" at all...but it's the same thing...indeed, the underlying point in "chaos theory" is that these "systems" LOOK "random" and "unpredictable" but they aren't really...they just become "far too complex" to defy attempts at "prediction" and come out looking "random"...they actually ain't...there is no inherent difference between "order" and "chaos" (this is the "chaos theory revelation", so to speak...don't go by "appearances" because they are the same thing, just poor measurement, complexity, interaction, etc. all conspire to make it "go wild" sometimes :))...

Hence, these things aren't "conspiracies" and you DON'T need any "conspiracy" (not to say these don't actually happen from time to time...the Holocaust is a clear "conspiracy"...the neo-con agenda of "control through fear to 'unite' the American people against the 'chaos' of the liberal society" is also another "conspiracy", though being "on-going", some people refuse to see it)...but when a number of "personal agendas" all match up, they create a "thrust" in a certain direction...everyone is making their money through "selling software", so the "thrust" becomes "get these users to buy things! Buy! Buy! I need to feed my kids, darn it! BUY SOMETHING!!"...and, on the "making a living", then who can blame them? BUT, this doesn't mean that it's a "reality" that we actually really do "need" to buy these things...very often, we don't need to do any such thing at all...but the media and advertising agencies create a climate of "consumerism" to constantly convince of this false "need" to "buy" all the time...

They all make their money that way...you can't really "blame" them...but this does not translate that users "demand" this or "need" it...they get it, like it or not...

> *I don't think that the reasons Beth gives
> provide a complete explanation, either.*

Well, it wasn't meant to be some "Hitchhiker's Guide to the Entire Universe" or anything in one post...even if *ahem* the size of my posts sometimes make it look that way ;)...

> *Sure, "portability"
> is going to increase the code size, but to the extent that
> it provides "functionality", and isn't just a "mantra",
> that isn't what I'd call "bloat".*

Neither is it what I'd call "bloat" either...the distinction hinges on that "isn't just a mantra" clause...

After all, with LuxAsm, I'm proposing "modular architecture" (that's NOT going to be "flat", "specific" and "simple" but, for the "integration" of the project (and as "foundations"), I do believe it's NEEDED to make sure the project can reach its objectives :)...we also have the "Unified Model" – which really annoys the "master of specific", Rene...so it's not meeting his "specific" approval – which is inherently placing a highly generic abstraction tool (capable of being used – especially with macros – to create very "HLL" HLLisms ;) NATIVELY into the syntax of the tool itself...again, I consider this NEEDED (that's why I put it in) in order to reach the objective of simultaneously pleasing "real assembly" coders as much as HLA users (that is HLA as in "high-level assembler" as a category, not solely Randy's tool...hence, yes, HLA but also MASM, TASM, etc. :)...without – this is what I tried to "delicately balance" – this "compromising" one for the other...make it all 100% "user-defined" then the programmer writes their code in the "style" they like and it should Hopefully result in both being able to use it in their preferred "style"

without this ever "getting in the way" of other people using the "other style"...not just that you aren't being "forced" to do any of it...but the "extension" I've also added is that you're also not "forced" to do it in any particular way, either...it's all "user-defined"...

Hence, in practice, we can see (as well as the fact that I have said this many times too...nearly every time I criticise "portability for the sake of portability", I pop in the "condition" that I'm not criticising the practice itself but that I'm criticising the _ABUSE_ of that practice)...that I don't have a problem with "abstraction" or other such things...indeed, that it requires "abstract thinking" to even notice these things...

- > *"Black box libraries"*
- > **can* be bloated, but not necessarily so. There *is* a*
- > *difference between libraries, and a "concise" library would*
- > *be just as effective in reducing developer time as a*
- > *"bloated" one.*

Ah, but the point is not so much the "library" part, as the "black box" part...and it's not that it's "inherently bad" (whatever Rene says ;)...

It's all to do with what Randy says in his radio interview: The difference comes when the programmer _CARES_...

And the problem can exist that the "black box library" is employed exactly because the programmer _DOESN'T CARE_...but, true, it is a _symptom_, not a "cause"...when you see the smoke, it's usually coming from a fire...

Plus, there is something to note about "black box"...in order to "modularise" and "encapsulate", this typically demands "indirection"...every "indirection" is working through a "proxy"...going "via", not direct...and, simply, you know the equation, yes? "speed = distance / time"...well, works for "logic" distances, as well as physical ones...the computer runs at a set "speed", it has a certain "distance" (measured in "instructions to complete" rather than miles or kilometers but it can still be thought of as a "distance to travel")...thus, it's going to take such-and-such a time...if you "indirect", then you increase the "distance" but the "speed" does not change (that is, the 3GHz machine keeps ticking away at 3 billion (American) ticks a second...indeed, the only speed changes are _downward_ - waiting for I/O devices - never upward)...hence, the "time" takes longer...

It's a minor thing in a single instance (usually) but it's an _accumulative_ thing...indirection to indirection to indirection to indirection, then loop back round and do it all again...and again...and again...you need the "indirection" to keep the "black box" "decoupled" from the other code...you know, you can't jump straight into the other module...you call via a "jump table" to an entry-point that makes a "transition" to a "layer" which communicates to a "library" (with another call via a jump table to an entry-point that makes its own "transition"

alt.lang.asm: Re: Computer time -> Developer time -> User time?

which returns) which returns to call another "library" (which calls via a jump table to an entry-point which makes a "transition" and returns) and then returns...

Of course, there is nothing "inherent" about "black box" or "libraries" that demands things work this way...BUT, inevitably, that's actually how it's working...

Basically, let's "anthropomorphise" this...you're working in a "team" of people...each "team member" does some work...BUT the other team members actually DON'T know what they are doing...indeed, it's a "secret" ("information hiding")...perhaps you need a hammer ("printf") to do your work and so do they...but you don't talk to each other, so you both might go and buy two hammers when only one is really needed..."team members" are only permitted to know what the others are doing at certain "key points" of interaction between them...

In such a situation, how do make sure they are working to maximum efficiency? CAN you ensure this when you have this "information hiding" secret that no-one knows what the other is doing?

Well, it is at least "slightly possible" from the perspective of one person having an "overview" of the whole process and co-ordinating the "team" to work efficiently (they don't necessarily need to know everything the other people are doing, so long as someone has sat down and worked this out :)...indeed, you could think of this as one of those "top secret" military projects where they get commercial companies to build the various "parts" but as each has no idea what the others are doing and are under "secret orders" then only the military themselves know what all the bits are destined to be used for (some new "fighter plane" design or something...like the radar-invisible Stealth bomber / fighter...or, if you believe those "flying saucer" fanatics (the Earth-bound ones, not "space aliens" ones...though, the "Roswell" folks would insist that "space aliens" told them how to build these UFOs ;), the "flying saucer" craft they are building at Area 51...oh, and boiling an egg? 3 minutes, of course ;)...)

But if the programmer doesn't care – doesn't take that "overview" – then these things can't be expected to necessarily work well together...

It's not the "cause", true enough...but it's often a "symptom"...and, of course, there's "theory" (where it's "not so bad") and then there's practice (where it can often get bad, as libraries are used to access libraries which use libraries to call libraries and the whole thing "bloats"...not because any particular "library" itself is necessarily bad but because the programmer has stopped taking that "overview" and caring about what's going on...

- > *It seems to me that to account for the size of "modern*
- > *apps", I have to assume that needed/desired features are*
- > *being implemented in a "bloated" fashion, and thoughtlessly*
- > *pasted onto already "bloated" code.*

Re: Computer time -> Developer time -> User time?

alt.lang.asm: Re: Computer time -> Developer time -> User time?

Yes; Because you have to think about how modern instruction sets can do more (more "addressing modes", atomic single instruction "exchange", etc.) in a single instruction than those older 8-bit instruction sets could do...the machine has more things "offloaded" from the CPU too...and `_code_` is actually a typically `_very compact_` thing...

It just doesn't seem right that `_code_` (as in `_instructions_`) for some not terribly complicated "feature" is bloated the size by 1MB or something...that's a `_lot_` in the physical code terms...but these applications seem to be throwing away MBs and MBs on "extra features"...

It's not, of course, possible to "generalise" because of all the different applications with different "features"...other than to say that it "intuitively" doesn't seem right...doesn't seem right at all...

> *This *may* be*
> *intentional – the "average software consumer" may think that*
> *a program that takes 3 CDs to install is a "better value"*
> *than one that fits on a floppy, even if the functionality*
> *and "feature set" is identical.*

Mind you, though they may want to make their software seem "better", this can only extend so far...what I mean is, if it's 1 CD rather than 3 CD then the `_price_` can be lower...and I think a user will find a lower price "better value"...

After all, even if it fits on a floppy, they could put it on a CD `_anyway_`...just waste most of the space on it...why not? The CD typically doesn't have anything else on it...and, to be honest, it's probably cheaper to buy CDs than floppies these days, anyway...less materials, easier construction...not that this is any great cost, anyway...as think how cheap blank floppies and CDs are to consumers and we ain't buying "in bulk" of a few thousand they will be buying in (which reduces it further: "economy of scale" :)...plus, they are getting it "wholesale" rather than "retail" too, I guess...no, this `_ISN'T_` a big cost at all...

So, sure, it fits on a floppy but stick it on a CD, anyway...you can put a better "picture" on the CD...they, to be honest, are more "convenient" too...you don't have to fill up the CD, in order to give the user the "impression" that it's "better value" than a floppy disk...

And, you can "change strategies" too, anyway...if it only takes up a floppy, then make a "compilation" of small programs and put them all on one CD...now, if the users sees "DOOM, DOOM 2, DOOM 3 plus 27 mission packs (and bonus Space Invaders, Asteroids and 100 card games)" all on `_ONE_` CD? Well, I think no-one's going to then think 3 CDs for just the one game is "better value", eh?

Plus there are other simple tricks...put on the cover: "New Supersqueeze(tm) compression technology! Exclusive to this CD! Squeezes 3 CDs' worth of data on 1 CD!!"...wow! It's got 3 CDs' worth of data on just

Re: Computer time -> Developer time -> User time?

alt.lang.asm: Re: Computer time -> Developer time -> User time?

the 1 CD because of "better technology"? That means I'm getting the "better value" of 3 CDs AND it's all on just 1 CD (much more convenient :) AND they must be "geniuses" to be able to come up with this really clever new "technology" that can do all this! (It's just programmed properly – no "bloat" – and ZIPped up with ZLib or something...that's all...but your users will think you're the greatest geniuses of "compression technology" there ever was! ;)...

[Did you ever download that "the product" demo thing? Basically, it's a 64KB file but it uncompresses itself, then it "dynamically creates" all the graphics and sound...and manages to compress what's effectively 9GB of data into *ahem* a 64KB file...30,000:1 compression! 15 minutes of high quality music on just one of the demos alone...now, if anyone asks me what is "better value" then these guys win hands down without any hesitation...and it's not just being "technically" impressed by it...I let some "non-technical" friends know about the demos and could stick tons of them on a single floppy disk (about 23 of them if all just 64KB big :)...yet, it provides HOURS of demo "entertainment" of a very good quality...they were jaw-droppingly impressed at all of this...and, ever since, I've never had to explain why I keep complaining about "bloody Microsoft!" thereafter...they now get the point perfectly well...they've seen what could be possible...that the problem is not their hardware but the frackin' bloated software they are running on top of it...remember, we all know this but this is "revelation" to many an "average user", who always thinks it's that they didn't buy the more expensive processor – the hardware – not that the software ain't using it properly...well, it makes sense from everything else they encounter...if the picture is crap on your TV? Well, that's "bad reception"...a "hardware" problem...if the car won't start on a cold morning? Well, that's an ignition problem...a "hardware" problem...when they see that the window is just sitting there hanging, unpainted, unresponsive while the disk is whirring? Ah, must be a "hardware problem"...the disk can't be fast enough or something...need to buy a better disk...and, of course, if they phone the "support line", what do they say: "hmm, yes, you need to buy more RAM and a better disk drive"...they "confirm" it all...it's a cruel illusion because the user is basically being deceived into thinking that their "hardware is no good" when it's perfectly good...but just isn't being used correctly...indeed, perhaps Rene is at least correct to start bringing up words like "moral" and "ethical"...I mean, if we willingly indulge ourselves in supporting and continuing this deception, do we deserve to sleep snugly at night? Think of those parents working hard in order to only just be able to pay for a reasonable PC for their kid's "education"...then they are scammed to buy this thing (which isn't needed) and then convinced that their "hardware is not good enough"...and they both work overtime and stuff like that, in order to "upgrade"...because they can see their kid likes and has an "aptitude" for computers (the computers teacher at school says so too :), so they want to "encourage" their kid to get on in life...to do what they like doing...starting to feel like shit yet? To screw them out of every last penny unnecessarily, just to afford what for yourself? A bigger "plasma TV" or something you don't really need and you know you don't really

Re: Computer time -> Developer time -> User time?

need...but the truth is that this probably IS a "true story" many times out there...but, on the other hand, if you do the best job possible and are selling software on the fact that it factually is "good" then you've nothing to be guilty for because you've not let anyone down...certainly not yourself either (yeah, go on, feel smug...you've earned it, so why not?)...then, in that case, get richer than Captain Kirk...honest day's work...you've raised yourself and others, not let anyone down at all...ever see those Spiderman movies? This is the "hero" in everyone: "with great power comes great responsibility"...people who take that on – without fear, without "cheating", without "avoidance" – and triumph are "little heroes" :)]

And all this doesn't quite add up with "reviews", anyway...what I'm thinking is, if Half-life 2 came on a CD rather than a DVD (they squeeze it into a CD)...then I don't think anyone would think "this is not good value"...but they are going to be judging it by magazine reviews, "preview movies" on the website, friend's "word of mouth" and so forth...I can't see any user picking up a CD of, say, DOOM 3 and then throwing it away: "Huh! Must be total crap because it's not on a 27 DVD boxset format!"...

On the contrary, people with only CD drives and not a DVD player (okay, very few these days probably :), which will be saying "Great! I can still play this without upgrading to a DVD player!"...

I mean, at the end of the day, it's quite simple...all these "illusions" might be "clever" but they can't beat something that actually IS "better value", "superior work", "high performance", "small footprint"...rather than just "pretending" to be so...in the words of Vic Reeves, "you can't compete with concrete!"...

> *I recall reading a post –*
> *long ago – from the author of a "password recovery" program,*
> *intended (nominally) to recover encrypted files to which the*
> *owner had forgotten the password (for some specific word*
> *processor, IIRC). He stated that his algorithm was so fast*
> *that he'd added a delay, so that his customers would think*
> *the program was "doing something".*

Missing a trick, really...people ought to learn a touch of "marketing"...simply market it as "the FASTEST recovery program on Earth! No program does it faster!"...then leave the "delay" out...

Study "P.T. Barnum"...when his show was crap, he actually SOLD IT ON THAT FACT: "Come see this terrible show! See how fake the illusions are! Witness how hilariously bad the magician is!"...and so forth...you know, if it's bad, then sell it as a "parody"...if it's short, then sell it as "minimalist art"...if it's incomprehensible, sell it as a "mystery"! A "riddle to be solved" ;)

The customers will see that it's "doing something" from the fact that the recovered files have re-appeared, for Pete's sake...sell it as being "super

alt.lang.asm: Re: Computer time -> Developer time -> User time?

fast!" and then they won't be in the slightest way
"disappointed"...basically, the Barnum "trick" is simply to "prepare"
people for what to expect...when your "show" meets that expectation (or
beats it), they are impressed by it...

Anyway, the most obvious "trick" of them all is to put up a
prompt...something like "file XYZ has been found, shall we recover
(Y/N)?"...then the "delay" is the customer's own reaction...

Nope, this is just stupid to make your program seem _WORSE_ by adding a
"delay" into it...I mean, if the files weren't there and then you run the
program and the files are recovered (and presumes that you're "recovering"
the files because you, you know, want to _USE_ them or something ;)...then
that's all you need to tell the customer it's "doing something"...

Indeed, in a sense, this is just how _BAD_ things have gotten...when
someone does a _GOOD JOB_ then they are "frightened" that because it's so
much better than everyone else's program, they should _RUIN_ it and _MAKE_
it _WORSE_, of all things, so that it doesn't "stand out" as being, you
know, good and fast...?!? People have become so "brain-washed" to expect
slow programs that they are "disorientated" by programs that don't take
forever? Scared that customers will reject programs that _WORK PROPERLY_?!?

This is a demonstration of why this industry problem has gotten so, so
serious that it can't really be "overlooked" anymore...that's just severely
twisted...

Now, I understand something like a "dealer" in "luxury items" like sports
cars or perfumes or whatever, deliberately keeping the price high (so it
doesn't come across as "cheap" = "tacky"...you know, it's the standard
"Lottery win" dream: Go out and buy a Ferrari or Porsche with your
millions...so, these things should always actually _deliberately_ remain
"out of reach" so that they stay as "dreams" for people :)...but it's just
TWISTED to ruin the _quality_ of a product to make it "the same as all
the other crap out there"...you know, the customer wants a plasma
TV...quick! Crack the screen and cover it in dirt and pull the power chord
until there's a "loose connection"! "Could I take a look at your hifi
players?"...quick! Pour coffee all over it! Smash up the radio component
with a hammer so it doesn't work!! We can't sell the customer a good
product because it might "look out of place" against all the other crap
around here...

Totally and utterly _TWISTED_...if this is what the industry is coming to,
then it has a _mental disease_...it needs _treatment_ fast...that's not
normal...that's not natural thinking at all...if this is what the industry
is coming to - deliberately creating _crap_ products, just because they are
"frightened" of "looking out of place" amongst the "rogue's gallery" of
utter crap - then these points aren't minor things to make...the industry
must be diseased and rotting to think in these terms...to be creating
people who think this is "normal" thinking...something that they "should"
do...it needs a "cure" and it needs one _FAST_...and, no, this time, I

Re: Computer time -> Developer time -> User time?

alt.lang.asm: Re: Computer time -> Developer time -> User time?

think we shouldn't be adding any "delay" for that "cure" to come ;)...

> > *Money talks.*

>

> *Yes, and what the "money spenders" like (apparently) might*

> *not match what someone who actually *knows* something about*

> *software would like.*

No, sorry...I just don't buy this crap at all...and "crap" it is...

It's all "self-fulfilling prophecy"...developers, as noted, are deliberately putting in a "delay" to make a program WORSE and then offering it to customers...I bet if you ASKED the customer: "Should I unnecessarily delay the program?", then they'd say "huh?!? Are you mad? You were seriously going to delay it for no good reason? Why on Earth did you think I wanted to be slowed down? Look, I can tell it's working from the fact that my recovered files have returned...that's what it's supposed to do, the fact that it's clearly happened tells me that it worked...the 'delay' tells me nothing"...

The industry wants to sell software (that's where money is made)...it's NOT about what the "customer wants"...indeed, if it was, then we'd simply be ASKING THEM what they want...I mean, they aren't going to object to being listened to for making better software...

But, no, instead, as the industry wants to hear "add more features! add more features! Doesn't matter what they are! Just add more features so there's a new version I can spend more money on!"...so it convinces itself that this is what users want...

But they don't...

And there's a simple way to prove this to your own satisfaction...find "average users" then ASK THEM...of course, do it properly with "unbiased" questions...

I did try this out before and the suggestion of "faster operation" / "no 'please wait' messages" / "faster loading of programs" was UNIVERSALLY APPROVED (not a single objector to it at all...logical enough, though, why would anyone want a program to run more slowly than it needs to? Only "twisted" people who get "frightened" that their programs are doing a good job (!?!)) ;)...contravertially for this discussion, there was NOT any significant call for "more features" at all in any of those I asked...or, at least, the point was that "more features" ONLY rated when it was suggested that they were "useful features"...

On the contrary, the industry has been trying to sell people the "universal device" which has all the MP3 player, TV, video phone, text messaging, personal organiser, calculator, tea-maker and so forth...only "geeky" people ever buy them...because people prefer the SIMPLE concept of "one device, one purpose"...

Re: Computer time -> Developer time -> User time?

alt.lang.asm: Re: Computer time -> Developer time -> User time?

The "3G" phones are having a HARD TIME being shifted...and many were trying to blame "technical problems put people off buying it"...ummm, NO! Because the people who haven't bought "3G phones" would HAVE NO IDEA that the "network went down" last Tuesday...you only find out such things once you've already bought...

It's a MYTH...it's an "urban legend" that the industry "self-deludes" itself with...

It works like this: One makes money by selling software...someone gets the clever idea of: "hey! If I just add on two extra features, change the bitmaps then I can sell it all over again and market it as 'new and improved'...most users will fall for this"...so they do it and they, indeed, "hype" well and sell more copies of something that isn't really substantially improved at all from the last version...this person KNOWS that it's an "illusion" and a "scam"...they've deliberately chosen to do it...but when they are successful, everyone asks "what is the secret of your success?"...well, the developer ain't going to reply: "By contemptuously deceiving my users with an illusion that things are better and new when, really, they ain't"...not least, the developer can't say that publicly...their future customer might be listening in on the interview, after all...they want to be able to "get away" with the trick once more...that won't work if they admit openly that it's a "trick"...others thus COPY what they see...why, look, if Gill Bates is successful with this strategy then it must be "what customers want"...we should do this too...

And then it becomes this grand "myth": Customers want random, arbitrarily "more features" so long as it is numerically more than the last version and the bitmaps are different...

Now – STOP! – apply logic...WHY on Earth would a customer want that? You know, they would want it for a REASON, yes? What is the logical REASON that they want "more features" for no useful purpose? Why do they want to be confused with too many buttons and options?

Ever helped out a "clueless newbie" with their computer? Do you know what causes all of their troubles? TOO MANY buttons...TOO MANY "options"...TOO MANY "features"...it confuses the hell out of them...I don't blame them...it often confuses the hell out of me too and I'm supposed to "know" about this stuff...

And, as I say, the matter CAN easily be settled...construct a proper unbiased set of simple questions to ask users about "what they want" and then see what the results are...one thing that might surprise is that the "wants" DON'T differ that vastly from what "technical" users want...the only really difference is technical people say "I don't want automatic" while the "newbies" say "automatic would be nice" (but, even then, they don't want things so "automatic" that they have no idea what's going on)...

Re: Computer time -> Developer time -> User time?

alt.lang.asm: Re: Computer time -> Developer time -> User time?

And this "abstraction helps them"? That's Sandra Bullocks too...I often go and help "newbies" with their computer problems...and it's amazing how the things that supposedly "help" do the _complete opposite_...for instance, they put in the CD and suddenly everything is running automatically and they are like "no! Stop! Why are doing that? I didn't want you to do that! How do I stop it?"...

Despite what Microsoft think is "useful" and "helpful", I found that if I made a series of "shortcuts" to each individual drive (labelling them with "hard drive", "floppy disk", "DVD drive" and then one "icon" to open up their internet connection and so forth :) then turned off "autoplay", it was MUCH MORE USEFUL to the "newbies"...exactly like it's much more useful to me – it's what I have too – because it's much more logical...

If you want to play a DVD then put the DVD into the DVD player...then double-click on the DVD icon I've created on the desktop...there, that's it...

And they are perfectly happy with this...users are NOT stupid...they are simply CONFUSED by all this "autoplay abstraction" nonsense...you know, they put a DVD in and it starts playing...no, no...I don't want you playing now...they hit the "close" button...right, okay, now I want to play you...ummm, there's no "icon" for the DVD player...how do I start this? All this has done is play things when they didn't want it to play...and then, when they do, the means to be able to do that has been "abstracted" away...

Look, it's on the front of the box...there, see? A DVD player...it has "DVD" written on it...in order to play a DVD, the user has GOT TO KNOW which slot is the DVD player and how to put a DVD into the DVD player...

So, as the user HAS TO KNOW what slot is the DVD player in order to use it...and it's a PHYSICAL, VISIBLE thing right there in front of their nose...how on Earth is it "useful" to "hide" this in an "abstraction"? It isn't...this is not only "nannying" and highly patronising...it is so much so that it defies all logic...the user does not find it "helpful", they find it vastly confusing...they fracking KNOW what a DVD is and which slot is the DVD player! If they didn't know this, then there's no fracking way they could ever actually play their DVD movies, right? You can't wave the disc around in front of the monitor and pray it'll work...you've GOT TO KNOW which is the DVD player and you've GOT TO KNOW how to put it in because that's a PHYSICAL thing...once they know which PHYSICAL thing – that they can plain see right there in front of them on the front of the box – is the DVD player, what is the point of "hiding" this from them on the desktop? That just confuses them because they can see the PHYSICAL player right there but they can't see anything on the screen that corresponds to the player to click on...

ASK the users...go HELP the users with their problems...the truth is, what they want does NOT that vastly differ from what you'd logically expect (IF you are in possession of any logic at all ;)...)

alt.lang.asm: Re: Computer time -> Developer time -> User time?

- > *Money talks in another way, too:*
- > *advertising. That has to be a factor in what people buy...*
- > *or use. If Windows and Linux were compared on the basis of*
- > *"users per advertising dollar spent", I'll bet Linux would*
- > *look pretty good!*

Agreed; I did, a while back, suggest that perhaps something like Linux might actually try selling itself at a "modest" price (perhaps as "shareware"...those who really can't afford it could *ahem* just kind of "overlook"...those that can...well, come on, pay up! ;)...and this isn't for "profit" or "wages" (although, if any developer were to "fork out" on something resources for the project, then they could be "compensated" back :)...but is put into a "pot"...if you like, it's not the developer's money...it's the "project's" money...this could be used to take out adverts in magazines or TV or whatever...to fund "support"...that kind of thing...

Now, bringing "money" into "open source" would alter the "dynamic"...you know, money has a strange way of changing people's behaviour (just check out how families suddenly start ripping each other to shreds over an "inheritance" from a dead relative...and, bizarrely, they do this even when the amount is next-to-nothing...all that "I was closer to them than you were!" / "You call yourself a relative! Where were you?" / "I was with my own family...we need the money, unlike you greedy lot, because I have a family to support...that's why I didn't get in contact all those years, I was working to make enough money for my family" / "A likely story! You're just trying to steal it all when everyone knows I was the 'favourite'" / "So? None of you came to visit her in her final days at the home! I was there every day" / "No, you weren't! You only turned up on weekends! I did visit and the nurse told me so!"...blah-blah-blah...they fight and fight...then, in the end, the "inheritance" of only \$1000 is shared 10 ways between them...the family is destroyed forever...well worth \$100, eh? ;)...

But it's a "thought", anyway...you know, kind of like how Red Hat commercially sell Linux...but, instead, the Linux developers themselves did that kind of thing...then the money is used for "support" and "advertising"...probably is, what happens if the money generated becomes more than needed for such things? Perhaps an agreement that anything deemed "excess" is given to, ooh, the "Tsunami Relief fund" or an AIDS charity or something?

Ah, it's a possibility...but getting something like that to work in practice, however "nice" it sounds in "theory"...well, I'm not deluded that it would be "easy" to get this to work, especially when the developers are on 4 continents or whatever ;)...

- > ...
- >> *"Sure Word is big and bloated. But that's what people*
- >> *want. They won't pay for apps without a lot of features..."*
- >
- > *I don't know how big Word actually is,*

Re: Computer time -> Developer time -> User time?

alt.lang.asm: Re: Computer time -> Developer time -> User time?

[Massive; Especially when you consider it's only a word processor...heck, it's not even a DTP program, where you'd kind of expect it might be bigger...there's a *_reason_* why it is the "proverbial bloatware" in everyone's examples ;)]

> *or exactly what*
> *features it has (I understand that, unless told otherwise,*
> *it'll execute any macro it encounters, possibly executing*
> *malicious code – not a feature I'd put on my "must have"*
> *list, but that's a different argument...)*

You know the "proverbial" toolbar button that "makes tea"? If any application will eventually include the "make tea" toolbar button then it'll be Word that gets there first...indeed, I haven't seen the latest version, perhaps it's already got there...put it this way, Word had so many "features" (90% of which are of "dubious" value...or make you think: "Oh, for frack sake, if you want to be a DTP package then become a DTP package and stop trying to this weird "highly extended word-processor" thing ;) that if someone did say "it now has a 'make tea' button" then this wouldn't raise an eyebrow or anything...not at all surprising...

Indeed, I'd even say: Microsoft are getting *_DESPERATE_* in thinking up "new features" to constantly "justify" why they need "new versions"...because they don't, of course...it's just that "constant forward motion" thing...indeed, they change the file format every version...do they do this because of "new features"? Nope, they do it just to introduce "incompatibility" to force people to "upgrade" all the time...

Oh, remember, "Clippy" was one of the "features" of "Office" and "Word"...have you ever met little old "Clippy" the paperclip? I think this annoying little pest makes the case all alone about the dubious quality of how Word features "help" the user, eh? Enough said ;)

> *So perhaps Word*
> *isn't a good one for me to use as an example... but I'll*
> *take the chance. I don't think anyone is going to write a*
> *"feature for feature clone" of Word in "carefully*
> *hand-crafted assembly".*

Perhaps not in assembly language but Star Office and Open Office – both free...while "Office XP" was some £400 on release here...£400? Microsoft software is the only software that *_DEFIES_* the "economy of scale"...being the most popular software in the entire world (other than Windows in order to run the darn thing), they should be generating this thing in *_massive_* "bulk" to get "bulk discount", "amortised costs" and so forth...on the contrary, though, Microsoft's "off the shelf" stuff with the largest sales of any software on the entire planet is almost costing more than "custom software" (okay, not quite...but it's close when it should, by almost rights, be "no contest" and a massive division by all "mass-production" / "economies of scale" arguments) – might not manage "feature for feature" (but, to be honest, with the amount of useless bloated "features" on Word,

Re: Computer time -> Developer time -> User time?

alt.lang.asm: Re: Computer time -> Developer time -> User time?

that's a GOOD THING) but if had these then you wouldn't in any way miss "Office"...it can read all their files, do the job just as well...

BUT – here's the point – if there are no "developer time costs", then, sure, it's going to take longer to develop at a "lower-level" to get the better quality program...but the whole point is: Still costs next-to-nothing...this is all "hobby time"...

It's a change of paradigm here...that's kind of the point I was making...instead of "bloatware" every year or two, there's one "kick arse" version every 4 or 5 years that has taken that "extra time" to do a far, far better job...

Yes, it takes more time...but, in "open source", time isn't an expensive cost anymore...hence, that's the "paradigm shift" itself...release "versions" less frequently because you are putting MORE EFFORT into each version...remember, the "commercial" needs the semi-frequent release to MAKE MONEY to pay all those "developer wages"...again, not applicable to "open source"...

Okay, not the best example maybe (though, it's more "design" we disagree with than "implementation") but Rene has put together RosAsm IN ASSEMBLY LANGUAGE in 5 years (and was doing it single-handedly for a while too...he has "contributors" now, though :)...this isn't a good enough "rate" for "competing" but it's a "hint"...and, as noted, I'm not talking "assembly language" but just more "low-level programming", "careful programming", "optimising code" and so forth...you can do it in C...it's more the "attitude", not the programming language...

If you like, the comparison is: The commercial releases every two years and adds on "some new features" each time...bloated and tends to get more "bloated" all the time...they need to release these frequently to make the money to afford the "developer time" and make a profit...

"Open source" has different costs and priorities...so should "open source" be otherwise following the exact same commercial assumptions and practices? Why "release frequently"? Why care about "rushing" because of "time"? Releasing frequently is needed in commercial development to make money to afford it all...of course, with "open source", it's kind of "constantly released" because anyone can dive in to download the source as it is being developed...but you know what I mean?

The "paradigm shift", in a sense, is to realise that Rene could work on RosAsm for the next 20 years, if he liked...and they'd still be no "developer time costs" for him to really worry about...so, how about – because of the changed "priorities" in "open source" – traded some of that "rush" for "quality" instead? The idea is instead of "bloatware with some extra features" every year or two, we have "leanware with fundamentally improved performance and features" every 4 to 5 years...you see what I mean? As there is no "developer time costs" to consider then there's the possibility of thinking more LONG-TERM...taking that "extra time" to the

Re: Computer time -> Developer time -> User time?

better job...

- > *Wouldn't have to be all that*
- > *"hand-crafted" – such a thing might be done using the*
- > *high-level features of HLA, for example – not the smallest*
- > *possible code, but not that bad (it's "not so humble" of me*
- > *to have *any* opinion of Randy's code, perhaps... but I do)*

Ah, yes...the point is that things have got so "bloated" everywhere else that even with every possible "bloat" you could try to incur using HLA's most "HLL" features, it'll still kick the "bloatware" size and performance...I mean, getting that confession out of Rene might be impossible but, as "bad" as HLA is in his eyes, it's still not as bad as using Java or VisualBASIC or whatever instead...doing our `_WORST_` with proper "low-level programming" is still typically going to weigh in better than the majority of HLLs (C / C++ is an "exception" here because it may be a "HLL" but it's one with a distinctly "low-level attitude"...indeed, some call it "medium-level language" to point out this little "exceptional" place it holds...it is a HLL but it's almost as if it's trying its best NOT to be as far as it can – except for that "portability" clause it needs to keep to)...

Indeed, I sometimes tempted to "re-classify" to consider C as being more on "our side" than the `_TRUE_HLL` side...the interpreted Java running "scripts" on top of "virtual machines" on top of "scripts" type of nonsense...

- > *Even using C – in the manner that Herbert suggests: *learn**
- > *assembly so you understand what you're asking the processor*
- > *to do, *know* your compiler well enough to understand what*
- > *code it will emit (Herbert sometimes neglects to mention*
- > *this part), and then write "thoughtful" C based on that*
- > *knowledge – would do the trick.*

I completely agree; I did note that my points were more about "low-level programming"...in which I would include "thoughtful C"...indeed, C without the "standard libraries" is just a kind of "machine-independent ASM" in some regards..."what the compiler does" is somewhat trivial from C to ASM...indeed, when MASM and HLA introdu