Things that Make us Stupid

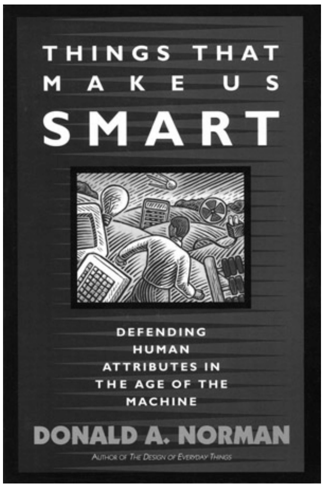
Peter Gutmann

University of Auckland

Things That Make Us Smart

Influential industrial design book by Donald Norman

Things That Make Us Smart (ctd)



THINGS THAT MAKE US SMART

DEFENDING HUMAN ATTRIBUTES IN THE AGE OF THE MACHINE

DONALD A. NORMAN

AUTHOR OF THE DESIGN OF EVERYDAY THINGS

Discusses the use of appropriately-designed technology to help humans accomplish tasks and achieve goals

Things That Make Us Smart (ctd)

Arguably the most usable piece of security technology ever

Source: photoeverywhere.co.uk

Ignition key tells the car when to start

- Security is a free extra

Things That Make Us Stupid

This works in reverse too…

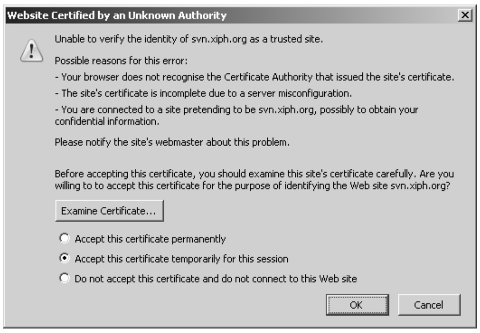
- Instead of acting as enablers/force multipliers, bad designs can reduce our effectiveness

Technology isn't always appropriately designed, and can have quite the opposite effect to the one intended

- This has proven particularly problematic in security user interfaces
- Designed purely by geeks for geeks

Things That Make Us Stupid (ctd)

Security UI often renders us incapable of acting usefully on the information presented to us



Website Certified by an Unknown Authority

Unable to verify the identity of svn.xiph.org as a trusted site.

Possible reasons for this error:

- Your browser does not recognise the Certificate Authority that issued the site's certificate.

- The site's certificate is incomplete due to a server misconfiguration.

- You are connected to a site pretending to be svn.xiph.org, possibly to obtain your confidential information.

Please notify the site's webmaster about this problem.

Before accepting this certificate, you should examine this site's certificate carefully. Are you willing to to accept this certificate for the purpose of identifying the Web site svn.xiph.org?

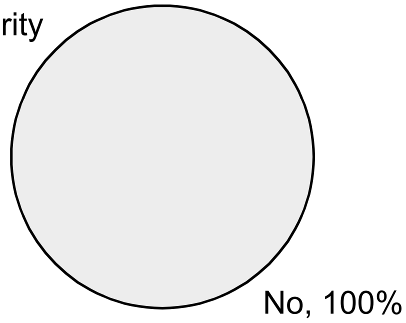
○ Accept this certificate permanently
● Accept this certificate temporarily for this session
○ Do not accept this certificate and do not connect to this Web site

- A lot of the time it doesn't even usefully present the information for us to use

Things That Make Us Stupid (ctd)

Pie chart of results of a 2005 usability study on the use of smart cards as authentication tokens

Would you want to use a smart card as an Internet security token?



Things That Make Us Stupid (ctd)

Cryptographic ignition keys have been around for 20+ years

- Failed in the market because they weren’t smart cards
- (The security industry has a lot in common with the fashion industry)

Things That Make Us Stupid (ctd)

How do you know it’s a bad design?

Effects of good design + stupid users is indistinguishable from bad design + smart users

Definition: “Smart”

- How geeks wish that users would behave

Definition : “Stupid”

- ¬ (How geeks wish that users would behave)

Users are “stupid” simply because they don’t behave in the manner arbitrarily tagged “smart” that’s defined as “How users should be using my software, dammit!”

Ding-ding, and Away!

In UK rail systems the guard signals the driver that the train can leave the station with two rings on a bell or buzzer

- “Ding-ding!”

Drivers became conditioned into pulling away as soon as they heard the signal

- “Ding-ding, and away!”

Classical pavlovian conditioning, they even use bells!

- Conditioned stimulus = bell, conditioned response = dogs salivate/drivers leave the station

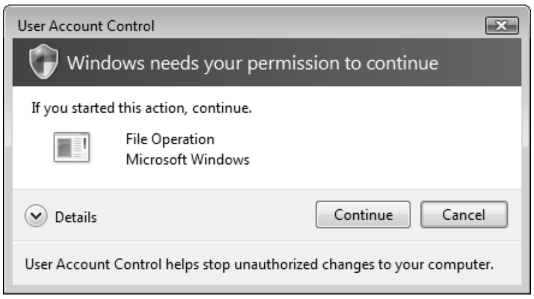
Ding-ding, and Away! (ctd)

Demarcation disputes over the role of guard and driver meant that this couldn't be fixed

- Finally fixed under media pressure in 1980 after a horrific accident that cost 7 lives

Design rule: If you have a usability problem that's so common it has its own name, it's something that needs to be fixed

Click-click, and OK



User Account Control

Windows needs your permission to continue

If you started this action, continue.

File Operation
Microsoft Windows

User Account Control helps stop unauthorized changes to your computer.

Design rule corollary: If your competitors are using your security UI in their advertising to make their own products look good, it's something that needs to be fixed

User Conditioning

Psychologists distinguish between two types of actions taken in response to a situation

Controlled processes

- Slow and costly in terms of mental effort
- Provide a great deal of flexibility in handling unexpected situations

Automatic processes

- Quick, little mental overhead
- Acting on autopilot, little control or flexibility

User Conditioning (ctd)

Example: Novice vs. experienced drivers

- Novice driver has to manually and consciously check mirrors, change gears, …
- Experienced driver performs these as an automatic process
- Novice drivers deal with this by load-shedding
  - Sacrifice driving speed for steering control

User Conditioning (ctd)

Automatic processes are people acting on autopilot

- Once the correct stimulus is presented, it’s very hard to stop

People click away warning dialogs without thinking

- This is an automatic process, performed without conscious awareness

The action is not only automatic, but people aren’t even aware afterwards that they’ve done it

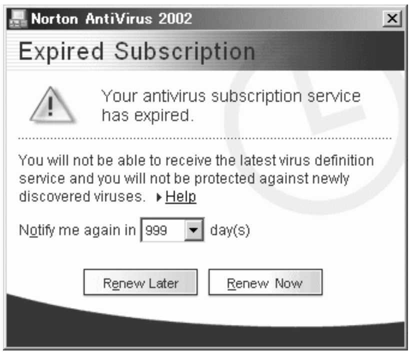
- “Did I lock the door/leave the iron on/…?”

User Conditioning (ctd)

Microsoft has encountered this in its automatic update system

- Users clicked away update dialogs without even knowing that they’d done it
- Many Windows systems are so riddled with adware and popups that this would be a natural action for users

Windows XP SP2 changed the update process to nagware to get around this



Norton AntiVirus 2002

Expired Subscription

Your antivirus subscription service has expired.

You will not be able to receive the latest virus definition service and you will not be protected against newly discovered viruses. ▸Help

Notify me again in 999 day(s)

Confirmation Bias

Humans are bad at generating testable hypotheses

- Phenomenon is called confirmation bias
- Try and prove, rather than disprove, a theory

Humans will look for (or cook) the facts in order to support the conclusions that they want to reach

- Dissonance-motivated selectivity, look for material that avoids cognitive dissonance (challenging your opinions)

Confirmation Bias (ctd)

How do you check whether a web site is valid?

- Enter your name and password
- If the site accepts the password, it's valid

If the security geeks had actually designed the mechanisms properly, this would be a valid site test

- TLS-PSK mechanism provides mutual authentication of client (browser) and (web) server without revealing the password or other shared secret
- (Little-used because it renders PKI superfluous)

Confirmation Bias (ctd)

US Navy addressed the problem of confirmation bias in tactical decision making after the Vincennes shootdown of an Iranian airliner in 1988

Introduced the STEP cycle for decision-making

- Create a Story (hypothesis)
- Test the hypothesis
- Evaluate the results

Makes creating a testable hypothesis an explicit part of the decision-making cycle

- Unfortunately a person in front of a security dialog hasn’t had US Navy training and constant drilling to assist them

Other Biases

Disconfirmation bias

- People are more likely to accept an invalid but plausible conclusion than a valid but implausible one
- “This site looks and acts like my bank site, even if it’s in eastern Europe. The browser must have got the URL wrong or something”

Blind-spot bias

- We can’t see our own cognitive biases

You just can’t win!

Other Biases (ctd)

CIA has published a special manual on dealing with biases

- Agency was particularly concerned with projection bias, a.k.a. “everyone thinks like us” bias
- “Psychology of Intelligence Analysis”, available online from `www.cia.gov`

Well worth reading, since some of the techniques are also useful for performing security analyses

- The other side has authenticated themselves, from now on we can trust anything that they send us

Other Biases (ctd)

Has hit numerous SSH implementations (client and server)

- Only check data validity before the user-auth phase
- The peer would never dream of authenticating itself and only *then* sending a malformed packet

Widespread in other implementations as well

- Unix access control: Only check security on the first access
- Signed ActiveX controls: It’s signed, it’s gotta be OK
  - Signed anything: All it means is that someone paid a CA for a magic token to turn off the warning dialogs
- Firewalls and the firewall mentality
- …

Geeks vs. Humans

The geeks who build computer software don’t think like normal humans

Example: Brand recognition

- Consumer-electronics store has two DVD players, a Philips and a Kamakuza
- Normal humans look at the Philips brand and buy it
  - Simple heuristics (RPD)
- Geeks will see that the Kamakuza player supports DivX, XviD, WMA, and Ogg Vorbis, has an external USB input and SD card slot for playing alternative media, and buy it
  - Economic/Bayesian model

Geeks vs. Humans (ctd)

For both sides this is a perfectly natural, sensible way to make a decision

- Both have come to completely opposite conclusions

Geeks vs. Humans (ctd)

Geek vs. human MBTI traits

- MBTI is a widely-used psychometric for personality traits
  - (Classifies personalities by Jungian personality types, not necessarily a personality test)

Geeks tend to be *TJ types

- Computer security people have a preponderance of INTJ’s
  - (Geek → *TJ doesn’t mean *TJ → Geek)

Why does this make geeks weird?

- Only 7% of the population has the *TJ profile

93% of users that geeks build software for think entirely differently from them

Geeks vs. Humans (ctd)

Final example of the difference between geeks and normal humans

All of Anne’s children are blond

Does it follow that some of Anne’s children are blond?

- (For logicians, assume that Anne has a nonzero number of children)

Geeks vs. Humans (ctd)

Most geeks would agree that the inference (a subalternation in Aristotlean logic) from “all A are B” to “some A are B” is valid

70% of normal humans consider this invalid

- This result is consistent across different cultures and re-phrasings of the problem (in the jargon, it is robust)

The people creating the security software just don’t think like the majority of the people using it

Security and Rationality

Our brains evolved for survival and reproduction, not to automatically seek the truth

- Quick and dirty techniques serve evolution better than purely rational ones

We can rationalise away almost anything

Security and Rationality (ctd)

Example: Subjects were given a canned biography on someone along with a random snippet of information like “He joined the Navy” or “He committed suicide”

- In every case they could explain the snippet via some item in the short bio
  - Sometimes the same item was used to explain away diametrically opposite facts
- When subjects were told that the information was fictitious, they still maintained their beliefs
  - c.f. earlier suicide-note analysis experiment

Security and Rationality (ctd)

People will concoct plausible explanations for something and continue to believe it even if they’re shown that the evidence for their conclusion is wrong

- This plays straight into the hands of con artists and phishers

Security and Rationality (ctd)

Example: Researchers created “inexplicable” situations by giving subjects two sentences covering totally unrelated events

Kenneth made his way to a shop that sold TV sets. Celia had recently had her ears pierced

- Subjects had ten seconds to come up with an explanation
- 71% of them could

Sentences were changed to contain a common referent

Celia made her way to a shop that sold TV sets. She had recently had her ears pierced

- 86% of subjects were able to come up with an explanation

Security and Rationality (ctd)

Example: Humans going to a phishing site (part of a phishing study)

www.ssl-yahoo.com must be a “subdirectory” of Yahoo!

sign.travelocity.com.zaga-zaga.us is probably an outsourcing site for travelocity.com

The company running the site probably had to register a different name from its brand because the name was already in use by someone else

Other sites use IP addresses instead of domain names so this IP-address-only site must be OK

Sites use redirection to a different site so this one must be OK

…

Reasoning this way is normal human behaviour!

Security and Rationality (ctd)

Extreme example: Patients whose brain hemispheres have been separated in order to treat severe epileptic attacks

- Split-brain/corpus callosotomy
- Left brain was able to rationalise away what the right brain was doing even though it literally had no idea why it was doing it

An example of a phenomenon called illusory correlation

- People see connections where there aren’t any

Security and Rationality (ctd)

Example: Subjects were shown drawings of humans supposedly done by people who had been matched to random psychiatric disorders

- Reported various signs in the drawings that were indicative of the disorders
- (Like the Rorschach test, the Draw-a-Person diagnosis method used to be common in psychiatry until experimental psychologists showed that the “results” obtained were meaningless)
  - They show something about the person who set the test, but not the subject

(Experimental psychologists *really* like messing with people :-)

Security and Rationality (ctd)

Self-deception isn’t a bug but a psychological defence mechanism

Depressed people have a *better* grasp of reality than non-depressed people, not the other way around

- Phenomenon is called depressive realism

Depressives suffer from a deficit in self-deception

Security and Rationality (ctd)

High levels of self-deception are strongly correlated with conventional notions of good mental health

- If the self-deception is removed, various mental disorders may emerge

Some level of irrationality is a fundamental aspect of human nature

The “Simon Says” Problem

Users are required to change their behaviour in the *absence* of a stimulus

Problem is well-known to social psychologists

- Experts can do this in some cases because they’ll notice the absence of a particular cue
- Novices don’t know what’s supposed to happen and so won’t notice when it doesn’t happen

The “Simon Says” Problem (ctd)

Example: Subjects are shown sets of trigrams with a special feature

- After (on average) 34 sets of trigrams, they figured out that the special feature was the presence of the letter ‘T’
- No-one was able to detect the absence of the letter ‘T’, no matter how many trigrams they saw

This is exactly what browser UI designers expect us to be able to do!

- We have to detect the *absence* of a stimulus like a padlock

The “Simon Says” Problem (ctd)

People find negative information far more difficult to process than positive information

- Educational psychologists advise educators to present information as positively-worded truths, not negatively-worded non-facts

The “Simon Says” Problem (ctd)

Example: Propositional calculus problems used by psychologists

If today is not Wednesday, then it is not a public holiday.

Today is not a public holiday.

- Is today not Wednesday?
  - People find these problems far harder to evaluate than positive-information ones

Example: Browser security indicators

If the padlock is not showing then the security is not present.

You couldn’t make this any worse if you deliberately designed it this way!

Inattentional Blindness

People don’t register objects unless they’re consciously paying attention to them

Best-known example is “Gorillas in our Midst”

- Subjects were asked to watch a basketball game with players dressed in black and white
- Told to count the number of times that each team bounced the ball
- In the middle of the game, a person in a gorilla suite pranced across the court
- Only 54% of users noticed
  - This amazing demonstration is often shown in pop-psychology programs on TV

Inattentional Blindness (ctd)

Commonly encountered on the road

- Drivers are looking for cars (and in some cases pedestrians), but not non-cars
- Bike riders are non-cars and therefore practically invisible to motorists

It’s possible to change your bike’s profile from “not-a-car” to “car” by mounting two driving lights far apart on a frame

- (Then your bike looks really ugly)

Inattentional Blindness and Security

The padlock and other security indicators are a perfect match for inattentional blindness

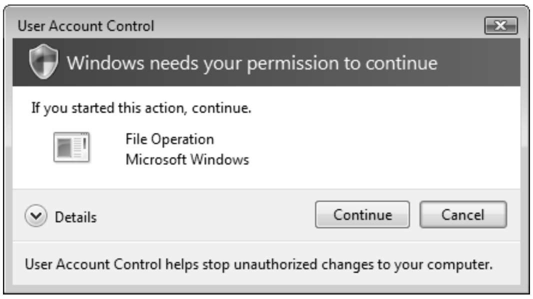
- Researchers have found up to 100% failure rates for these indicators

IE6 SP2 added a security bar to warn users of security issues

- One usability test found that not one user had noticed its presence

Inattentional Blindness and Security (ctd)

Windows Vista added UAC dialogs to warn users of (potential) security issues



User Account Control

Windows needs your permission to continue

If you started this action, continue.

File Operation
Microsoft Windows

User Account Control helps stop unauthorized changes to your computer.

- Informal tests revealed that no-one had noticed that it had different colours in different situations
- Now try and find out what the colours actually signify…

User Education

Security is a nebulous concept that’s difficult for users to relate to

- Removing red-eye in a photo program is a tangible goal with tangible actions
- “Being secure” is, uhhh, …

In normal computer usage, users can rely on satisficing

- Click on something that looks appropriate
- If it doesn’t work, go back and try again

User Education (ctd)

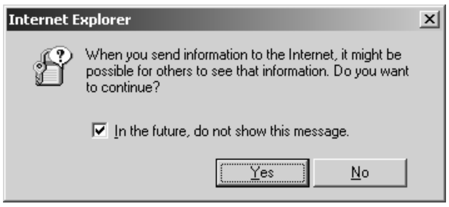
Users don’t seem to mind how often they have to click (up to a point), as long as each click is an unambiguous, mindless choice

- Le systéme D, to muddle through

People don’t make optimal choices, they muddle through, and only read the instructions when all else fails

User Education (ctd)

This interacts *really* badly with the way that most applications handle security



When you send information to the Internet, it might be possible for others to see that information. Do you want to continue?

☑ In the future, do not show this message.

HCI researchers call these warn-and-continue (WC) dialogs acknowledging that users will click straight past them

- “Yes, yes, yes, yes, yes, oh dear…”

Dropping problems into a WC is easy for developers but doesn’t help keep the user safe

User Education (ctd)

User studies have found that users expect the application to make security decisions for them

- Application developers expect users to make the security decisions

No-one takes responsibility, because it’s the other side’s problem

- Psychology professor James Reason calls these design flaws “latent pathogens”
  - Ding-ding, and away!

User Education (ctd)

Mozilla developer reported WC dialogs as “a chronicle of indecision within the walls of Netscape”

> Every confirmation window and question to the user marks a case where two internal camps couldn’t agree on the most secure way to proceed and instead deferred to the user

Firefox developers discovered via feedback from users that they saw through this deception

> [Warning dialogs are] intentionally obfuscated warnings that companies can point to later and say ‘Look, we warned you!’

User Education (ctd)

Conventional human-based error mitigation techniques

- Pre-selection screening (academic grades, psychological profiles)
- Selection screening (entrance exams, interviews)
- Work training
- On-the-job evaluation

Computer human-based error mitigation techniques

- (None)

No standard mitigation techniques can be applied for any but a very restricted set of uses (the military, SCADA use, etc)

User Education (ctd)

Even experimental attempts to train users have run into problems

In one study, researchers found that training had no effect on users’ ability to detect phishing email

- All it did was scare them into rejecting more email of all types (legitimate and phishing)

User Education (ctd)

In EV certificate evaluation tests, users actually performed *worse* after the “education”

- The education trained them to rely on the wrong security indicators

  > The EV approach is to do more of what we have already discovered doesn’t work
  >
  > — Ian Grigg

US banks have a proud tradition of training their users to rely on inappropriate security indicators

- (See the phishing talk slides for more on this)

User Education (ctd)

If user education was going to work, it would have worked by now
— Anti-virus researcher Vesselin Bontchev

The Bystander Effect

New York city, 1964: Bar manager Kitty Genovese is murdered outside her apartment

- Attacked at 3am, attacker got scared and ran off after stabbing her several times
- Returned 10 minutes later, searched the area for her, repeatedly stabbed her, and raped her
- After half an hour, someone finally called the police

Received widespread coverage in the US, although common reports exaggerated the details

The Bystander Effect (ctd)

Prompted psychologists to research the phenomenon in depth

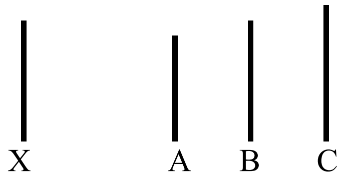
Effect is so noticeable that experimental psychologists have quantified the chances of a bystander becoming involved

- One bystander = 85% response rate
- Two bystanders = 62% response rate
- Five bystanders = 32% response rate
  – (Figures apply to for this particular experiment, not every case of the bystander effect)

The Bystander Effect (ctd)

This effect exists in many variations

Example: Does sample X match lines A, B, or C?



- Subjects were placed in a room with people who reported that A or C matched X rather than B
- One bystander = 3% agreement
- Two bystanders = 14% agreement
- Three bystanders = 32% agreement

The Bystander Effect (ctd)

Why did you agree with the (incorrect) assessment?

- I wanted to fit in
- I thought there was something wrong with my eyesight since the others couldn't *all* be wrong

This effect is particularly pernicious on the Internet

- Effect increases with the number of bystanders
- On the Internet, the *entire world* is a bystander

This is the worst possible situation into which to deploy something that falls prey to the bystander effect

The Bystander Effect (ctd)

In 1994, a security bug was discovered in the PGP 2.61 random number generator (xorbytes bug)

- This had been present in publicly-available source code since PGP 1.0

In 2002 a similar bug was discovered in GPG (xorbytes memorial bug)

- Had been in the code undiscovered for *four years*
- Discovered by accident by someone who was interested in seeing what security code was like (i.e. not by a code audit)

The Bystander Effect (ctd)

Linux /dev/random code had a bug in the entropy addition code

- Lay undiscovered for 1½ years
  - (This code has been repeatedly hacked-over and updated, and the mechanisms used are undocumented. The presence of this and other bugs isn't too surprising)

Debian OpenSSL RNG bug was present for two years and made all OpenSSL/OpenVPN/OpenSSH keys guessable

- (Still trying to determine how it was discovered)

"Many eyes make bugs shallow" only applies if people are interested in finding the bugs

The Bystander Effect (ctd)

Phishers use the bystander effect to build confidence in the phishing sites

- Provide a hotline number for users to call to check the site authenticity
- No-one ever calls it, but they trust the site more because of it

Conclusion

Humans' minds work very differently from geeks' minds

- Many applications are written by geeks for geeks
- (Even supposedly user-friendly ones)

The mind works in very counterintuitive ways

- There are good reasons for the behaviour, but they're not at all obvious

Geeks are weird

- (No, really)

More Information

Extended-length writeup of this talk, including assorted approaches to solving these problems, at

```
http://www.cs.auckland.ac.nz/
   ~pgut001/pubs/usability.pdf
```

The 10 Inescapable Truths of Security UI

Truth #1 (Security UI Prime Directive): If the user can’t understand your security interface, it doesn’t exist (— Clare-Marie Karat, Carolyn Brodie, and John Karat)

Truth #2: Any security message/dialog that can be rephrased as “Do you want to continue to perform your job/intended task?” simplifies to a boolean value of TRUE

Truth #3: Nil utilitatis sine probatione: Any security UI that isn’t tested in the real world reduces to Truth #1

- Truth #3 Corollary: Any security UI that isn’t tested for failure conditions as well as success conditions reduces to Truth #1

The 10 Inescapable Truths of Security UI (ct)

Truth #4: If your security UI is more complex than username + password → “accepted”/“declined”, you've lost 80% of your user base and your UI reduces to Truth #1

- Truth #4 Corollary: If your UI involves concepts like certificates, key fingerprints, CAs, and webs of trust, this increases to ~100%

Truth #5: If user education was going to work, it would have worked by now (— Vesselin Bontchev).

Truth #6: A security handshake can have only two possible outcomes, “connected with both sides mutually authenticated” or “not connected”. Anything else is just building substrate for phishing attacks

Truth #7: If a validation check fails then the data or session should be treated as if no security was present, not worse than if no security was present (— Phil Hallam-Baker)

The 10 Inescapable Truths of Security UI (ct)

Truth #8: Security features that are off by default will stay off

- Truth #7 Corollary 1: Security features that turn themselves off (for example anti-virus subscriptions) will stay off.
- Truth #7 Corollary 2: Security features that annoy the user and can be turned off will be turned off and stay off

Truth #9: Security UI design is the hardest of all types of UI design. While the typical user can muddle their way through a field sown with cowpats, they can't muddle their way through a field sown with antipersonnel mines

Truth #10: The best is the enemy of the good: Any effective but less than theoretically perfect security technology will be panned by experts, self-appointed or otherwise