DOI:10.1145/3342707

Vinton G. Cerf

Undo, Redo, and Regrets

THIS MONTH I have been mulling over a number of conversations and experiences that I would like to share with you. They all revolve around the notions that some actions or decisions are irreversible and thus deserve considerably more attention than choices or actions that can be undone. I am a regular Apple MacBook Pro user and one of the features I like the most is the Command-Z key that, most of the time, can undo something I just did and wish I hadn't. Unfortunately, not all things in life have an attached Command-Z button.

Ismail Serageldin, the founding director of the new Library of Alexandria^a explained it this way. "There are some solutions I call 'tooth-paste solutions' because once you put them into operation, like toothpaste, you can't put them back in the tube." This reminds me of the story of the magic lamp and its resident genie. Once you rub the lamp, the genie escapes and you can't put him back again. Perhaps Pandora's Box^b has similar features—once opened, the ills of the world escape and resist recapture into the box.

This suggests to me that our increasingly digital, software-driven world has many opportunities to exhibit these features. There are software updates that, once done, cannot be undone. As much as I wish all actions were reversible, they aren't. I suppose there is something about entropy that drives this one-way direc-

tion. Unscrambling an egg provides a concrete example. For sufficiently powerful cryptographic methods, the loss of a cryptographic key spells doom for any attempt to decrypt the content. This should be a major concern for users who rely on cryptography to support confidentiality. Key management becomes a crucial matter thanks to the consequences of losing the key(s).

Judy Estrin draws another example of things you might not be able to undo. "Sorry, we broke democracy." I think of artificial intelligence and machine learning: "Sorry, the AI-based self-flying airplane did not work and we can't revive the fatalities."

In our increasingly complex digital world, these features (hazards, phenomena) deserve our utmost attention to minimize the irretrievable consequences our software systems might introduce. More than once, I have experienced the "software update" that has run amuck and proves resistant to being put back into a known and safe state.

There are software updates that, once done, cannot be undone. As much as I wish all actions were reversible, they aren't.

Software backup practices can easily fall into this category. Suppose you religiously back up your primary disk to a backup disk—but the backup fails and you have partially overwritten the previous backup. Then the primary disk fails. You have no complete backup. Ready. Aim at big toe. Fire.

It occurs to me that we would do ourselves a great favor if we were to design our digital systems to the maximum extent possible to avoid irreversible traps to fall into. Software upgrades offer examples for good practices: Backup the system disk *before* doing an upgrade so you can recover if the upgrade fails or produces some unacceptable consequence. Of course, this raises a fundamental question about backups. Are they complete? Do you ever test the backup system before it is actually needed to ensure the backup is complete? I have heard many horror stories of people who have religiously performed the backup incantation only to discover that—in reality—the backup system was not doing a thing, or was only saving some but not all of everything that might be needed to recover from a serious disk failure. Testing the backup system is just as important as carrying out the procedure itself.

Ronald Reagan's famous quote seems appropriate here: "доверяй, но проверяй."^c Reagan said this with regard to strategic arms limitations and reductions, but it seems to apply equally well to software systems.

^c Translation: "Trust but verify."

Vinton G. Cerf is vice president and Chief Internet Evangelist at Google. He served as ACM president from 2012–2014.

Copyright held by author.

^a <http://bibalex.org/en/default>

^b In classical mythology, Zeus gave Pandora, the first woman, a box with strict instructions that she not open it. Pandora's curiosity soon got the better of her, and she opened the box. All the evils and miseries of the world flew out to afflict mankind.