Alex Groce (agroce@gmail.com), Northern Arizona University

Richard W. Hamming's *The Art of Doing Science and Engineering: Learning to Learn* is one of the true essentials, a book to be read by every scientist or engineer (and certainly software engineer) interested in the big picture; it may be safely skipped by those who aim only at a life and career where one has "merely survived and amused" oneself, as Hamming himself puts it in his final, summative, chapter.

The essence of this book is a combination of anecdote and aphorism drawn from a mature, active mind, reflecting on a great career spent at great places (Los Alamos, and then Bell Labs in its heyday, to be precise). There is also a great deal of mathematics, but the aphorism, backed by well-chosen anecdote, is fundamental; you do not need to know or care much about differential equations (and there are many here) to benefit from this book. It is hard to convey the value of the anecdotes, in a short review. The aphorisms, however, can be more than mentioned; what follows is, in reverse order, the entire set of Hamming's bolded, inset, aphorisms, by chapter (combining multiple chapters on the same topic, e.g., simulation). I have truncated one long item, which offers up a description of pure mathematics; you will have to read the book to get that one. The focus on aphorism is appropriate; I think Hamming himself would have agreed that this entire wonderful book is a kind of elaboration and proof of Pasteur's "Luck favors the prepared mind", which Hamming quotes not once, not twice, but *seven times*.

The unexamined life is not worth living. Change does not mean progress, but progress requires change.

You get what you measure. The rating system in its earlier stages may tend to remove exactly those you want at a later stage.

The closer you meet specifications, the worse the performance will be when overloaded. Part of systems engineering design is to prepare for changes so they can be gracefully made and still not degrade the other parts. If you optimize the components, you will probably ruin the system performance.

90% of the time, the next independent measurement will fall outside the previous 90% confidence limits! There is never time to do the job right, but there is always time to fix it later.

What you did to become successful is likely to be counterproductive when applied at a later date. If an expert says something can be done, he is probably correct, but if he says it is impossible, then consider getting another opinion. All impossibility proofs must rest on a number of assumptions which may or may not apply in the particular situation. An expert is one who knows everything about nothing; a generalist knows nothing about everything.

Man is not a rational animal, he is a rationalizing animal. Pure mathematics consists entirely of… Mathematics is the language of clear thinking. Mathematics is nothing but clear thinking. Mathematics is what is done by mathematicians, and mathematicians are those who do mathematics.

What you learn from others you can use to follow; What you learn for yourself you can use to lead.

Why should anyone believe the simulation is relevant? A simulation is the answer to the question, "What if…?"

The purpose of computing numbers is not yet in sight. The purpose of computing is insight, not numbers.

It has rarely proved practical to produce exactly the same produce by machine as we produced by hand.

The unexamined life is not worth living. No vision, not much of a future. Unforeseen technological inventions can completely upset the most careful predictions. The past was once the future and the future will become the past. In science, if you know what you are doing, you should not be doing it. In engineering, if you do not know what you are doing, you should not be doing it. Education is what, when, and why to do things. Training is how to do it.

This is not to suggest (and some of these hint otherwise) that this is largely an "airport" business inspiration book for scientists and engineers, dressed up with some math because the author is a notable mathematician. Those books are usually glib, but only sham-profound. Reading Hamming again, I struck by how many times the passing thoughts of Hamming connected to ideas mentioned in Gilson and Langan's *Modern Philosophy: From Descartes to Kant*, which I happened to be reading at the same time. Hamming openly aims to focus on what is foundational in each field (after preparing a concise and compelling back-of-the-envelope argument that everything else has a short half-life, so should not occupy all your mental effort), and succeeds in *thinking hard* to such an extent that he echoes and expands on the problems that faced Descarte, Malebranche, Montaigne, Bacon, and Pascal.

There is, in addition to the deep philosophy and the anecdotal background for the above pithy sayings, a great deal of technical meat here: Hamming's insights from the 90s on the fundamental limits and promises of computing, on AI, on understanding and dealing with high-dimensional space (as required in all complex designs, including software designs) are remarkably up-to-date, and a bit in advance of the date. Hamming on LLMs is technically not here, but the chapters on AI read, at times, as if Hamming were musing on the most recent advances. For the price of admission and attention, in addition to an examination of how to do great things, you get a brief history of computing, hardware and software, a practical guide to

understanding signals and noise, a first-hand account of information theory and coding theory, and even a quick summary of why quantum mechanics matters. Hamming's writing is friendly and intimate, perhaps because the chapters are taken from classroom lectures at the Naval Postgraduate School; the result is something like the Feynman Lectures on Physics, but focused on computing, and short enough to read in a few evenings. Go! Read it if you haven't, and if you have perhaps it is time for a re-read; have you done great things? If not, why not, and it's not too late.