**Margaret Hamilton**

****Margaret Hamilton has a long list of accomplishments. She is a computer scientist, a systems engineer, and even a business owner. She's also the woman who got man to the Moon.

**Early Life**

Born on the 17th of August 1936 in Paoli, Indiana, to Kenneth Heafield and Ruth Esther Heafield, she attended the University of Michigan and then earned a BS in mathematics with a minor in philosophy from Earlham College in 1958. Although a strange combination, Margaret’s interest in philosophy stemmed from father who was a philosopher and a poet and her grandfather who was a writer, a schoolmaster and a Quaker minister[[1]](#footnote-1). When Margaret graduated university, software engineering wasn’t an industry, never mind a college course, so this mix of philosophy and maths helped Margaret to decide and integrate what was important for the path that she would take in the fields of computer science and systems design and software engineering.

It was 1959, not a time when women were encouraged to seek out high-powered technical work. Hamilton, a 24-year-old with an undergrad degree in mathematics, secured a job as a programmer at MIT and took her first steps to becoming one of the founding fathers/mothers of software engineering, more by chance than by design.

**SAGE**

Hamilton’s software engineering career took its fledgling steps in 1959 when she moved to Cambridge Massachusetts with her husband as he studied at Harvard Law. Here she took an interim position in MIT's meteorology department with Professor Edward N Lorenz, known for the Lorenz effect, chaos theory and the butterfly effect. This position was intended to be tempory, merely a job to support both her and her husband until he graduated and she could pursue an advanced degree herself in abstract mathematics. At the time, the field of computing was still relatively new. Computers were large, often taking up entire rooms, if not an entire warehouse, and the work of programming was still developing.

“When I first got into it, nobody knew what it was that we were doing,” Hamilton told Wired. “It was like the Wild West. There was no course in it. They didn’t teach it.”

The task Hamilton was given was to develop software for predicting the weather using LGP-30 and the PDP-1 computers. In an interview for Futurism, Jolene Creighton asked her about this early programming experience and Hamilton explained that under the guidance of Dr. Edward N. Lorenz Hamilton began programming in hexadecimal and binary and began the process of designing and building software. With Lorenz’s guidance, she learned how to build software in hexadecimal and how to take advantage of the LGP30 computer hardware to most benefit the software’s performance. Lorenz encouraged her to design and build what would today be called a ‘mini operating system’ for her applications. One of her early assignments was working on SAGE, a weather prediction system which later became an early missile defense computer system.

Then from 1961 to 1963, Hamilton worked on SAGE (Semi-Automatic Ground Environment). The SAGE Project was an extension of Project Whirlwind, started by MIT, to create a computer system that could predict weather systems and track their movements through simulators. SAGE built on this foundation to create a system for anti-aircraft air defense from potential Soviet attacks during the Cold War. The system at Lincoln Labs, hidden deep inside caverns, was huge and was the first to have real-time computers. Hamilton's role here was to write software for the XD-1, the first AN/FSQ-7 computer. [[2]](#footnote-2)

One of the most important things that came out of Hamilton’s time spent working at SAGE was that it set Hamilton on her career-long mission to discover what causes errors and how to avoid them, as she revealed in the Futurism interview:

*SAGE was one of the first jumping off points where I became interested in the subject of software reliability. When the computer crashed during the execution of your program, there was no hiding. Lights would be flashing, bells would be ringing and everyone, the developers and computer operators, would come running to find out whose program was doing something bad to the system.*

*The only information the computer provided to the developer for debugging his program was to light up a very large register on the console of the computer, showing the address where the program halted.*

*When my program was running, it always sounded like the sound of waves on a beautiful seashore, so we all referred to it as the ‘seashore program.’ Until one night at 4 am in the morning when I got a call from one of the computer operators who said something terribly wrong has happened with your program. When I asked him how he knew, he said ‘it no longer sounds like a seashore!’ Now, we had a new way to debug, using sound!*[[3]](#footnote-3)

**NASA and the Apollo Missions**

Hamilton’s contribution to SAGE made her a prime candidate for the position at NASA as the lead developer for Apollo flight software. At the time, software and programming weren’t high on NASA’s list of priorities. Software wasn’t even included in the original budget which outlined the engineering requirements for the missions. The software for Apollo was outsourced and when Hamilton heard that MIT had received the contract to develop the software for sending man to the moon she was quick to respond:

*I immediately called MIT to see if I could be involved in what sounded like the opportunity of a life time, and within hours I set up interviews with the two project managers at MIT. Both of them offered me a position on the same day as the interviews. I did not want to hurt anyone’s feelings, so I told them to flip a coin as to which group was going to hire me.[[4]](#footnote-4)*

It was Dan Lickly, who had worked on the Polaris Guidance Computer and had been appointed to lead group which would develop the Guidance and Control methods for the Apollo Reentry vehicle, who won the toss and this determined how Hamilton's career would proceed. As Hamilton explains on her NASA Home Page:

*Computer science and software engineering were not yet courses to be taught (or disciplines to be named). These were pioneering times. Learning was by "being" and "doing" on the job; as more people came on board, the more I became an "expert" and rose up through the ranks. A real world system developed in uncharted territory, one could not have asked for a better research environment to set the stage for what would come next and remain thereafter (a life long career centered around developing more advanced methods and tools for designing systems and developing software).*

Programming was a very different activity at the time of the lunar landings as the photo of Margaret Hamilton standing beside the printout of the source code for the Apollo Guidance Computer (AGC) indicates at the start of this biography indicates.

Hamilton together with and her team at MIT coded all of the code by hand. This involved writing all of the AGC source code on AGC source coding paper, and then they would give it to the "keypunchers" to copy. This involved punching the code from the coding paper onto cards would be processed overnight on a mainframe computer to output the source code as listings (printouts) which were then put into binders. The code is written in assembly language and this is why it prints to such a large volume of paper - each line printed is a single machine instruction! Once the code was considered solid, expert seamstresses, referred to as "Little Old Ladies", threaded copper wires through magnetic rings (a wire going through a core was a 1; a wire going around the core was a 0) to create rope memory that was built into the onboard navigation computer.

One thing that hasn't changed is that task was demanding and stressful. Once, after a late-night party, Hamilton rushed back to the computer lab to correct a piece of code she’d suddenly realised was flawed:

*“I was always imagining headlines in the newspapers, and they would point back to how it happened, and it would point back to me.”*

This is just one of the many examples of Hamilton’s acute awareness errors and the devastating effects that they could have. It was also during this period that Hamilton coined the term "software engineering to gain recognition for the work she and her team were doing:

*Software during the early days of this project was treated like a stepchild and not taken as seriously as other engineering disciplines, such as hardware engineering; and it was regarded as an art and as magic, not a science. I had always believed that both art and science were involved in its creation, but at that time most thought otherwise. Knowing this, I fought to bring the software legitimacy so that it (and those building it) would be given its due respect and thus I began to use the term “software engineering” to distinguish it from hardware and other kinds of engineering; yet, treat each type of engineering as part of the overall systems engineering process. When I first started using this phrase, it was considered to be quite amusing. It was an ongoing joke for a long time. They liked to kid me about my radical ideas. Software eventually and necessarily gained the same respect as any other discipline.*

Programming, especially for Hamilton and her team, was not a normal nine to five job. Many forget that Hamilton was a young mother and she was conscious of the time she spent away from her daughter, so on evenings and at the weekend, it would not be unusual to see the mother and daughter duo in the lab. This time spent in the lab together led to a very important discovery and contribution to Apollo 8 by Hamilton’s daughter, Lauren. Hamilton recounted in an interview for Wired with Robert Macmillan, one day Lauren, then aged 4, was playing with the MIT command module simulator’s display-and-keyboard unit, nicknamed the DSKY (dis-key). As she toyed with the keyboard, an error message popped up. Lauren had crashed the simulator by somehow launching a prelaunch program called P01 while the simulator was in midflight. As a result of this incident, Hamilton wanted to add code to prevent the crash. Hamilton requested time to add error handling to prevent this error occurring but NASA denied her request so she created a program note that would be available to NASA engineers and the astronauts, saying: Do not select P01 during flight.

She recounted this concern in 2001 at the First Conference of the Apollo Guidance Computer History Project:

*Many of the things I was intrigued by had to do with how to make the mission software safe and reliable. And one of the things I remember trying very hard to do was to get permission to be able to put more error detection and recovery into the software. So that if the astronaut made a mistake, the software would come back and say "You can't do that." But we were forbidden to put that software in because it was more software to debug, to work with. So one of the things that we were really worried about is what if the astronaut made a mistake -- We were also told that the astronauts would never make any mistakes, because they were trained never to make mistakes.*

*So we were very worried that what if the astronaut, during mid-course, would select pre-launch, for example? Never would happen, they said. Never would happen. It happened.*

In December 1968, five days into the historic Apollo 8 flight, the first manned orbit of the moon, astronaut Jim Lovell inadvertently replicated Lauren action and selected P01 during flight. Launching this routine had the effect of wiping out all the navigation data Lovell had been collecting and without that data, the Apollo navigation computer wouldn’t be able to figure out how to get the astronauts home.

The only positive was that the cause of the problem was clear and Hamilton and the MIT coders started working on a fix. Nine hours later they had a plan, Houston uploaded new navigational data and Apollo 8 and its crew returned to earth. This led Hamilton further into her war with errors.

Hamilton’s true talent in software engineering and error handling became apparent on July 20, 1969, during the Apollo 11 mission. While most of the world remembers Neil Armstrong's "one small step for man, one giant leap for mankind" announcement, three minutes prior the software overrode a command to switch the flight computer’s priority system to a radar system. The override was announced by a “1202 alarm” which let everyone know that the guidance computer was shedding less important tasks (like rendezvous radar) to focus on steering the descent engine and providing landing information to the crew. Armstrong and Aldrin landed on the Moon, rather than aborting the approach due to computer problems. It transpired that due to a "checklist error" the crew had prematurely initiated the radar system needed for leaving the moon. This produced an "executive overflow condition" of the 72K on-board computer. Fortunately, due to the way in which Hamilton's team had designed the system, the software prioritized the task of landing the lunar module above all competing tasks and effectively saved the mission.

**Post Apollo**

After Apollo, Hamilton worked on Skylab, the United States’ first space station, and some of the preliminary system software requirements for the Space Shuttle’s flight software. This was a logical continuation of the path she had been following and her area of specialization, which she explained in 2001:

*As time went on, I got very interested, even more, and more interested in error detection and recovery, because of the errors that took place and how we could avoid them in the first place. We were doing simulation, much simulation, but of course, we couldn't test the flight in real time. We had to simulate it, and I got very interested in static analysis.*

*We began to analyze all of the errors that had taken place on the flight software when we were in actually Validation & Verification mode. When each of many of the error reports came in asking for "reason for error," the engineers would fill in a response and they would just say "bug" and that wasn't enough.*

*So we got very interested in how we wrote errors up, so that if we understood the error, then we could maybe prevent it on the next mission. We did a thorough analysis of the on board flight software, including the errors themselves, and began to categorized those errors[[5]](#footnote-5).*

By the time Hamilton left NASA in 1976 she had evolved a theory with six axioms, that have to do with defining software in such a way as to avoid interface errors. This was initially called Higher Order Software and she co-founded a company named to devise a complete tool suite environment that could eliminate such errors. They created a product called USE.IT, based on the HOS methodology developed at MIT.[[6]](#footnote-6) It was successfully used in numerous government projects. One notable project was to formalize and implement the first computable IDEF, C-IDEF for the Air Force, based on HOS as its formal foundation.

Hamilton left the company, HOS, in 1985. In March 1986, she became the founder and CEO of Hamilton Technologies, Inc. in Cambridge, Massachusetts. The company was developed around the Universal Systems Language (USL) and its associated automated environment, the 001 Tool Suite, based on her paradigm of Development Before The Fact (DBTF) for systems design and software development.

**Awards**

In 1986, she received the Augusta Ada Lovelace Award by the Association for Women in Computing. This award is given to individuals who have excelled in either (or both) of two areas:

1. Outstanding scientific and technical achievement

2. Extraordinary service to the computing community through their accomplishments and contributions on behalf of women in computing.

In fact, the Apollo guidance software was so robust that no software bugs were found on any crewed Apollo missions, and it was adapted for use in Skylab, the Space Shuttle, and the first digital fly-by-wire systems in aircraft. Hamilton was honored by NASA in 2003 when she was presented a special award recognizing the value of her innovations in the Apollo software development. The award included the largest financial award ($37,200) that NASA had ever presented to an individual up to that point.

In 2009, she received the Outstanding Alumni Award from Earlham College.

In 2016, Margaret Hamilton was honored again – this time at the White House. President Obama has selected her as a recipient of the Presidential Medal of Freedom. The highest civilian award of the United States, it is awarded to those who have made an especially meritorious contribution to the security or national interests of the United States, to world peace, or to cultural or other significant public or private endeavors.

On April 28, 2017, she received the "Computer History Museum Fellow Award" that honors exceptional men and women whose ideas have changed the world.[[7]](#footnote-7)

**Conclusion**

Women like Margaret Hamilton, who, beginning in 1961, helped NASA “develop the Apollo program’s guidance system” that took U.S. astronauts to the moon, as Maia Weinstock reports at MIT News. “For her work during this period, Hamilton has been credited with popularizing the concept of software engineering." Robert McMillan put it best in a 2015 profile of Hamilton:

*It might surprise today’s software makers that one of the founding fathers of their boys’ club was, in fact, a mother—and that should give them pause as they consider why the gender inequality of the Mad Men era persists to this day.*

Hamilton was one of the most influential software engineer of the 20th century and her legacy is the 400 billion dollar software engineering industry that exists today. As Obama said at the Presidential Medal of Freedom ceremony in November 2016, “[Hamilton’s] example speaks of the American spirit of discovery that exists in every little girl and little boy who know that somehow to look beyond the heavens is to look deep within ourselves and to figure out just what is possible.”

1. https://futurism.com/margaret-hamilton-the-untold-story-of-the-woman-who-took-us-to-the-moon/ [↑](#footnote-ref-1)
2. http://www.i-programmer.info/history/people/10030-margaret-hamilton-apollo-and-beyond.html [↑](#footnote-ref-2)
3. https://futurism.com/margaret-hamilton-the-untold-story-of-the-woman-who-took-us-to-the-moon/ [↑](#footnote-ref-3)
4. https://futurism.com/margaret-hamilton-the-untold-story-of-the-woman-who-took-us-to-the-moon/ [↑](#footnote-ref-4)
5. https://authors.library.caltech.edu/5456/1/hrst.mit.edu/hrs/apollo/public/conference1/hamilton-intro.htm [↑](#footnote-ref-5)
6. M. Hamilton, S. Zeldin (1976) "Higher order software—A methodology for defining software" IEEE Transactions on Software Engineering, vol. SE-2, no. 1, Mar. 1976. [↑](#footnote-ref-6)
7. https://en.wikipedia.org/wiki/Margaret\_Hamilton\_(scientist) [↑](#footnote-ref-7)