REDS

Recursive Engineering Design System

Purpose

REDS is an industry sponsored educational framework, with a common goal to evolve a recursive engineering design system with a common abstraction domain. The initial feasibility project will focus on digital systems design.

Implementation

The REDS framework consists of three main programs for students. These programs will be run by students, with oversight from local educators and industry professionals.

MSC – Middle School Camp

A summer camp to teach students the primary tools of a STEM discipline. In the digital track, students will learn how to code.

HWS – High School Work Program

An after-school and weekend program, where students will develop lab skills by working on actual systems problems for sponsor companies. Students will develop the software needed for the REDS through direct recursion on the problems they encounter.

UML – University Methodology Lab

The University Lab will formally validate the tools and processes developed by the HWS, and produce academic research from the data generated by the other programs.

Funding

The long-term goal is for the programs to be self-sustaining. The HWS will generate income by performing valuable work for industry. It also serves as a talent development pipeline for STEM employers.

In the feasibility stage, REDS will seek sponsorship from local communities and industry.

Contact Information:

Brandon Awbrey
713-305-5620
Brandon.awbrey@microchip.com

Common Abstraction Domain

Aboard *USS Texas*

"Showtime!" exclaimed Captain Graybeard.

"Ready, Captain," said Ensign Nguyen. "We've got stable current on three buses."

"Attitude?"

"PJ verified Vesta gravity-assist with a Mars return in 405 hours, full burn. Reaction mass margin of 8%," reported the navigator, Ensign Reyes.

"Excellent! Comms?"

"Comms offline sir," XO Marjanović reported. "We need an EVA for repair. We still have entanglement with SOLSTORM.[1] We're backchannelling base telemetry, but no acknowledge from Earth yet."

"Suit up!"

"Captain," Ensign Nguyen said, "There's a problem. We can't do an EVA."

"I am the system for solving problems, Winnie. Hit me."

"We're at 325% reaction power," Ensign Nguyen reported.

"We're cookin', what's the problem?"

"That *is* the problem, Captain."

# # #

The Texas had left Phobos Station on a hard burn, trying to rescue the freighter *Stark*. Enroute to Mars, *Stark* had begun a course correction burn, and lost communication, due to a solar flare at the margins of SOLSTORM detection. During the storm, *Stark* tried to maneuver but continued to burn six minutes too long at an attitude that sent them on an orbit into the outer reaches of the solar system. *Stark's* SOLSTORM telemetry indicated that crew was alive in the hardened-life pods, which had not been jettisoned.

Texas burned hard, but midway received its own SOLSTORM warning, giving them just four minutes to adjust to a lie-flat attitude. The lie-flat maneuver had never been attempted during an active burn.

The lie-flat maneuver put the massive water tanks that held the reaction mass in between the crew compartment and the sun, shielding the crew from the radioactive burst. The maneuver was normally handled by the AI with the main engines shut down. *Texas* could not halt the burn and still intercept *Stark*. If *Texas* couldn't catch her, *Stark'*s crew and passengers would be lost. *Texas* was the only ship in range with the raw delta-V to catch Stark. *Stark's* present orbit would return to the inner-Solar system in 486 years.

PJ[2], the ship's AI, refused to execute the lie-flat maneuver under burn. While it could use the thrusters to adjust to the correct attitude, under burn there would be too much oscillation from sloshing in reaction mass tanks to damper before the surge hit. In addition, their intercept vector would require them to perform a long braking burn to return to Mars. PJ was certain it could protect the crew from the burst, it couldn't avoid risk to the fusion engines and several other exposed subsystems. Since the Texas

---

[1] SOLSTORM – Early warning network for Solar Storms that uses entangled quantum communication to give spaceships a few minutes of warning to protect crew and equipment from damage.

[2] *Texas's* AI had originally been named Calvin, after an original rocket scientist. When Captain Graybeard first stepped aboard, his command to Calvin was, *"Rename yourself PJ and go stand in the corner and watch my six, until I have some important work for you to do."*

was already travelling fast enough to escape the solar system, losing those systems would put the crew at risk, as they would have no way to slow down and return to the inner solar system.

PJ said, *"Sorry Captain, the maneuver during burn violates Asimov's first and second laws. I can't do it."*

"Little Baby Johnny, you ready to check in?"

"Yes, Captain," said XO Marjanović, who like all the Martians, towered over their captain. "Engineers, clutch time! Let's go!"

XO Marjanović and ensign Nguyen joined teenage ensigns Reyes, Patel, and Liu at the REDS console. REDS was a group of five workstations, all facing a central point. Each workstation looked like an arcade gaming chair, with a group of joystick controls on each armrest, and folded off to the side, a standard keyboard. The Martian teens had neural implants, so they used their minds to control the input to the system. The manual controls had been a compromise for chipless Earthlings or for those that had objections to cybernetic technology. Through their neural links, all the engineers shared a common display of a detailed blueprint of *Texas*. An engineer could focus on each subsystem, and delve down into the hierarchy of the components, revealing the source code or wiring schematic of every part of the ship. They could easily change code and have the AI simulate the results, providing a near instant test and risk analysis of any change they made. Five engineers could control separate sub-systems, or they could all shift focus and collaborate on a single problem together. The rest of the crew could not see the visualization directly, but panels around the bridge would render the models on flat screens.

Captain Graybeard observed on his command screen. It took the squad about twenty seconds to program the vectors for the attitude adjustment.

"Ready, thrusters!" Marjanović called.

"Ready, power!" Nguyen called.

"Ready, nav-comm!" Reyes called.

"Ready, life-support!" Patel called.

"Ready, top-level!" Liu called out.

Captain Graybeard cast a dramatic gaze into the central bridge camera.

"All right baby Martians, get ready, because it's about to get heavy up in here."

The captain looks off to his left. There was no one there, the AI was omnipresent in a spaceship. The crew had used a video panel to show an anthropomorphism of PJ so their ancient captain could have someplace to focus when he talked to the AI. At first, it had been endearing to the crew, who didn't have to vocalize their thoughts to speak to the AI. After a few weeks, it got silly.

"PJ, you're on the bench, Captain in command! Put all systems under crew control, override authorization Mamba-Magic-Skyhook-Diesel-Metta-Jackson, now!"

Graybeard turned back towards the forward bridge camera.

"I am the system! Let's get cookin!"

He makes a hand motion like he's stirring a pot of soup.

The course correction went off without a hitch, but the sloshing in the reaction mass tanks produced a large shimmy. When the wave of radiation hit the ship, multiple systems shut down and alarms went off. They lost one of four engines completely, and two of them produced oscillating surges of current on the power buses. They lost one power bus, but the team dynamically rerouted and were able to keep the reaction engines firing long enough to ensure an intercept trajectory with the *Stark*. They shut down reaction power to the two malfunctioning fusion engines and began repairs.

Meanwhile, Captain Graybeard ordered the shuttle crew to rescue the crew of *Stark*. The shuttle docked with *Stark* and recovered the four crew and six passengers. They did not have time to offload any of the valuable supplies bound for Mars. The shuttle team attached a solar sail buoy to *Stark*, which would bring it back to Mars orbit in nine years.

The Martian engineers continued to debug the failed fusion systems from REDS. Their single working engine could not produce enough power to brake and return to Mars. PJ informed them that their best orbit for Mars return on one engine was a Jupiter gravity-assist, but that would take six years. With the additional passengers, they only had enough food and consumables for 37 months. They'd saved *Stark* but put themselves at risk, exactly as PJ had warned.

Luckily, the propulsion system had been designed by the crew, they knew every component and had written the control code themselves. While the shuttle rescued *Stark*, XO Marjanović and his squad of teenage engineers were able to hotwire and reprogram the power system and get three of the reaction engines back online. If they could keep the three engines alive, they'd could slingshot around the asteroid Vesta, and still return to Mars in a little over two weeks.

# # #

"Captain," said Ensign Nguyen, "We're cooking, because we're too hot. The radiators can only dissipate 200% reaction power. We're dumping excess heat into the reaction mass, but the water is already at 93 degrees. Once the water starts to boil…"

Captain Graybeard said, "*Texas* was designed for full power burns."

"The shakedown was only for solar cruises sir," Ensign Nguyen said. "We've never run over 200%. The multiple engines are for redundancy, not a hard burn. We're also generating a lot of waste heat with the coupling feedback loop."

Texas had four fusion engines, but one was completely scrammed and had been used for spare parts in the repair. The hotwiring job on the reaction engines had the drawback that two of the three working fusion engines would only run at 100%, and the third, undamaged engine had to be running at higher pressure to provide a boost to the two repaired engines. They could only dial the single working engine down to 105% or risk losing the other two. They needed a 300% power burn to brake into a return slingshot trajectory, or risk being hurled into the outer solar system.

Graybeard, an Earthling, did not have a chip embedded in his skull. He used primitive hand gestures to bring up a screen of the ship's schematics, and quickly focused on the radiator assembly. Two knobby flanges of pipes and vanes protruded from *Texas* at its stern, just forward of the massive reaction mass tanks.

"There's a telescoping extension here," the captain said. "We just need to extend it. That should increase surface area and put the vanes out 94 meters, away from the heat radiating from the tanks."

"That's a problem, sir. Our only controls for the radiator are the flow rates valves."

"There's a push motor here on the schematic. It's wired to this GPGCU.[3] We just need to program it to extend."

"That's the problem. The Standard C-API wrapper around radiator control only gives us access to the flow values. There's no API for the extension motor."

"There's a manual crank. EVA!"

"It's 112 degrees on the surface near the tanks, sir," Ensign Liu said. "An EVA would be suicide."

"It's a software problem, hack a fix!"

The crew was silent for a moment.

"Sir," Ensign Nguyen said, with some hesitation. "The entire software stack for the radiator sub-system is written in AGGIE."

"AGGIE? PJ, what the hell is AGGIE? In context."

"*AGGIE is a 5-bit encoding scheme conceived on December 6, 1992 by a Texas A&M graduate students for a NASA code contest. Using only capital letters and excluding the letters "B" and "V". AGGIE*

[3] General Purpose General Compute Unit. Digital Logic lego bricks Martian engineers used to build larger systems.

has no symbol for "3" or zero, using "E" or "O" instead. AGGIE-SCRIPT is a programming language derived from AGGIE, a language unique in that it had no punctation or operators, nor any end of line characters. On November 10th, 2037, the 25th anniversary of A&M's victory over Alabama, provost DeAndre Jordan declared AGGIE the official file format and language of the school. Texas A&M spent over $550 million converting…

Graybeard interrupted the AI, "How much did bits cost way back in 1992 – I was just a baby."

"In 1992, a one gigabyte hard drive cost approximately $2000. On a per bit basis, this would equate to .00000025 dollars, or approximately 4 million bits per penny."

XO Marjanović said, "Captain, I think you're missing the point. The important factor isn't the cost of data bits, but the date that AGGIE was invented."

"Huh. Baby Johnny, you're telling me that—"

"Saint Johnny's birthday," said Ensign Nguyen, forcefully, irritated.

"You mean the entire university adopted the stupidest encoding scheme ever invented just because it was conceived on the day Johnny Manziel was born?"

"You know what they say about Aggies; From the outside looking in you can't understand it, and from the inside looking out you can't explain it," said Ensign Reyes.

"There must be somebody here who can code in AGGIE?"

The Martian teenagers exchanged bashful glances. They were first generation Martians, and of course, knew the real history of Aggies on Mars.

After sending the first seven Challenger missions to Mars and discovering water and all other critical sources of minerals needed for a self-sustaining colony, Mars patron El Jefe had picked his young crew of colonizers wisely. He picked the brightest and hardest working people, choosing a pair of students from all the major Texas public universities, including Texas A&M. El Jefe was the king of the publicity stunt. When it came to A&M, there really was only one choice for the male candidate, an Aggie who had dropped out of his degree program in 2014, but decades later, had decided to return to school. The other A&M candidate was the XO's mother, a Nobel prize winning nuclear physics PhD from Serbia, who had chosen A&M over UH and UT because she didn't like big cities.

The cover story of what happened during those glorious first two years of Mars colonization was now a fairy tale that had been made into a streaming series for Earth children. How, at great risk to his own health, this brave Aggie had risked his life to save the colony during a series of storms, mishaps, and malfunctions, and finally, after establishing order and training his robot army of twelve into a super team, Scobee City was ready for the second wave of colonist, which arrived two years later. It was a story of triumph over adversity, and a tragedy as well; due to radiation exposure, the hero had to be evacuated back to Earth orbit for medical treatment.

The Challenger Mars missions had been a much more diverse coalition, in their inclusion of international astronauts, they had failed to include any Aggies on any of the first seven missions. What really happened was much more pathetic. In his zeal to accomplish the most "Mars firsts" by an Aggie, the hero had tricked Scobee City's AI into overriding his radiation exposure limits, so that he was authorized for unlimited surface missions. He became first Aggie to throw a four-hundred-yard pass on Mars. First Aggie to bungee jump into Valles Marineris. First Aggie to survive a Mars sandstorm in a monster Cybertruck. And finally, perhaps most importantly, the first Aggie to be treated for severe radiation sickness on Mars. After returning to Earth orbit, he had gene therapy to repair his DNA.

In orbit he remained. It turns out he'd also tricked the AI into deleting his daily health protocol, which mandated strict exercise regime for the colonists in case they needed to return to Earth. Ten years later, El Jefe asked him to be the first Aggie manager of the first Buc-ee's in space. To this day, voyagers to the outer solar system would stop at Lagrange L2 Buc-ee's to get a selfie with the most famous Aggie as a rite of passage.

The young Martians learned the secret history from their parents, but El Jefe made so much money from the streaming rights, keeping it secret seemed a fair tradeoff. There was one first he didn't accomplish. He wasn't the first Aggie to procreate on Mars. The XO's genome was a source of great mystery, but he was born three years after the most famous male Aggie left the planet. His mother would only say, on choosing his name, *"Since they say Mars kids will likely be very tall, I wanted him to have a tall role-model to look up to. And also, in case he's not so tall, for him to remember, height isn't everything."*

No one on the bridge knew AGGIE.

Ensign Patel piped up, "My mom got drunk with Saint Johnny in a bar in Houston, does that count?"

"That would make me the most qualified," said the captain. "I was in a bunch of different clubs with Johnny. Did I ever tell you about back in the day when I was in the ---"

Ensign Nguyen didn't want to hear the story again, "Captain, the reaction mass is at 95 degrees. We're going to have to vent soon."

Captain turned and faced the bridge screen, stood tall and straight, preened his beard, and then commanded.

"System Time! PJ, write some code to extend the radiator extension trusses to maximum length, and then translate that code into AGGIE, and then inject it into the GPGCU of the radiator. Get cookin' or else we'll all be cooking!"

Half a second later, a message was displayed upon the bridge screen:

Code composed in standard C.

Translating into AGGIE............HALT 0x0000002A!

Red warning lights flashed.

Asimov violation – zeroth law. Please reboot.

"Winnie, what's wrong with PJ?"

"Captain, PJ asserted on error decimal 42. Apparently translating code into AGGIE violates Isaac Asimov's zeroth law."

"I know one, two and three, but what is zero again?"

"*A robot may not injure humanity or, through inaction, allow humanity to come to harm*. Apparently, if a modern quantum computer tried to translate code into AGGIE it would destroy humanity."

"Don't they have AIs in College Station?"

"A&M modeled their AI after the cow brain."

The captain stroked his beard.

"Let me get this straight. We can either get deep fried and return to Mars, or remain on this trajectory, slingshot around Jupiter but die of starvation?"

"We've only got seven minutes before we've got to vent!" the XO reminded.

"Mamba time, baby Martians! Winnie, you're my code ninja. Hack a fix."

# # #

Not that the captain had ever asked, but Ensign Nguyen did not like being called Winnie. Her given name was Drexler, and her call sign had been *Glide*, obviously, before this showboating Sun Devil from Cali showed up to boss her squad of Martians.

The billionaires on Earth had loved the idea of an all-Martian crew. *Texas* was the most technologically advanced ship ever built.

All the power, propulsion and control systems had been designed on Mars by Martian engineers, led by XO Marjanović. The auxiliary systems design had been outsourced to Texas public universities. UT designed life support, UH the Comms, and Tech did the interior design. Wanting to be fair, El Jefe's trustees had awarded the simplest system, the radiator, to Texas A&M. It was basically plumbing.

Texas was supposed to be an inspirational message for Earth children, to inspire them to greater engineering innovation and a quicker path to a green and sustainable Earth.

The natural leader of this crew should have been the XO, who not only was the first human born on Mars, but at eight years old had invented the portable fusion reactor that now powered the green economy on Earth. At nine years old, he invented the gravity boot. This allowed little Martians to skip the hours on the treadmill or centrifuge and just play regular teams sports to develop their muscles. An added side effect, on Earth, the boots allowed people just to glide on an anti-gravity wave whenever they wanted to go, putting the automobile industry out of business. The designs were open-source, free from patents. El Jefe's motto of *Save You Money* put these wonderous inventions in the hands of people rich and poor, all over the world.

Even after Martian innovation had saved Earth from destroying itself, they still couldn't envision giving the keys to a trillion-dollar ship to a teenager. The old fools didn't even really debate it.

Transcript of the International Space Billionaires Committee

Mark Z. – *It's worth a trillion dollars, We can't let a bunch of teenagers run it.*
Queen B. - *What we need is a leader, a systems person, who isn't intimidated by a bunch of young seven footers.*
Steve B. - *I know just the guy! He is a system!*

And the sports metaphors; Hey, captain *Let's get cookin*, read a room. *Mamba Time* and *Hey Winnie, throw some Shaq-Fu at the condenser coil*, didn't really motivate this particular crew. Their parents had primarily come from Space City, Texas, where all the rocket scientists lived. This bearded prima donna shows up just to boss people around and tell the people back on earth, "I am a system." Everybody knew the job should have gone to Johnathan Paul Marjanović. Instead, the XO had to listen to the captain mock him, calling him, *Baby Johnny.* Sure, the XO had been the most famous baby ever born, but that was sixteen Earth years ago.

Ensign Nguyen's personal trove of inspiration wasn't stupid cliches, but simple words to live by. *Believe It, and then get it done*! All Martians, even ones raised outside the rich history and tradition of Scobee City, held one maxim in their heart; *Always Think Big!*

While turn of the millennium billionaires fought each other on social media for the best plan for humanity to colonize space, and tried exotic schemes like crypto or a breathing-tax to fund their ridiculous endeavors, a successful local H-Town businessman took another approach. He went to Vegas, and bet his entire fortune on the Longhorns, Texans, UH Cougars men's and women's basketball, Rockets and Astros to win their respective championships in the same calendar year. The odds were about six billion to one. When Altuve hit the walk-off home run on [redacted] [redacted], [redacted][4] to win the bet, El Jefe became the owner of the entire gambling industry, including most of the land in Nevada and Atlantic City. Because Vegas had laid off action overseas, he became the majority owner of real-estate in Macau, Singapore, and Monte Carlo. El Jefe became the richest human in the history of the world. He immediately auctioned off his land windfall and invested all of it into his own plan to colonize the solar system. Showing what you can do when you put a real Texan in charge, he brought in professionals from all over the world to teach STEM subjects and skills to Houston area school children. Within a few years, he launched the Challenger missions.

[4] Science fiction writers who make side deals with gamblers should not be trusted.

Winnie was jacked into REDS and saw instantly what her four fellow systems engineers were seeing. All three of them were looking at the display of the source code of the firmware for the radiator assembly GPGCU.

A very long line of "text" without punctuation – this was the raw AGGIE-SCRIPT.

ADDTIMESTAMPTO7ANDSETAGGZARATORESULTSETAGGZARCTOZEROMOEFOREACHFLOWALE…

"We're not going to decode this in seven minutes." Nguyen said. "We got to think outside of the box."

"If we had comms…"

"Comms wouldn't help," Nguyen said. "We're nine minutes round-trip to Earth."

"Maybe if there were another Aggie in radio distance."

"Wait a minute," Nguyen brought up a window that connected her to the manifest database.

"Captain! There's an Aggie on board. One of *Stark's* passengers. It looks like they're a vet…."

"A vet, all right," the captain said, looking introspective. "Now we're talking. A vet might be about my age. It would be nice, a grownup to talk to, I'm liking the sound of this. A shorty Aggie vet---"

"Captain," said the XO, "You must NEVER use that word."

"It's okay, Baby Johnny, I'm from Earth. I can say it. It is funny and endearing when I say it."

"Captain," said ensign Liu. "Do you really think it's endearing when you're taller than 99.9% of the people on Earth."

"She's not a veteran, sir," said ensign Nguyen. "She's a veterinarian."

"A what? An animal doctor? On a trip to Mars?"

"Coach Doctor Cheng Xin," Ensign Nguyen explained. "She's an assistant coach of the women's basketball team, and chair of Texas A&M's genetic engineering program."

"Medical Bay! Please send up Doctor Cheng Xin!"

"Funny," said the captain, "Why they send a basketball coach animal doctor up in space?"

PJ answered, *"There is a loophole in Texas law that allows Texas A&M to pay anybody with 'coach' in their title an unlimited amount of money."*

"She got the supermax?"

"Doctor Cheng Xin was the top free agent genetic engineer," said Patel. "They had to make her a coach, the only way the Aggies could outbid the Musk and Bezos people. She's in charge of an Aggie project called Mars mambas."

"Snakes on a spaceship?"

"These snakes were modified for terraforming," said Patel. "These snakes burrow into the regolith and turn the toxic Mars dust into oxygen and water and produce a valuable biological soil compound with their feces. The Mambas are designed to live for hundreds of years and grow to lengths of 400 meters."

"Wait a minute. I'm pretty sure that was a movie, back when I was in the —"

"In Frank Herbert's Dune series, it was worms," said Patel. "The Aggies changed it to snakes to avoid royalties. They were over budget from all the coach's salaries."

The bridge door opened. A small woman entered.

"***La Puma Rojo***," exclaimed the entire crew, including the captain.

They all recognized her. The 150cm point card who had led the University of Houston Women's basketball team the NCAA championship. Her seven clutch three pointers in the final three minutes of the game had led to victory, the second rung in El Jefe's bet that had funded their colony. Unfortunately, like most college athletes who don't turn professional, the world had lost track of Cheng Xin. After graduating from UH with a biology degree, she'd continued her studies and become a Veterinarian.

She'd gone on to invent a gene mutation process that led to specialized therapies for all sorts of diseases on both animals and humans.

"I'm Cheng Xin," she said. "Hey, I recognize you! You're the guy Steph used to splash all over—"

"No need to rehash ancient history in front of the children," said the captain.

"Who you calling shorty? We can hear every word you say on the bridge in the medical bay."

"Sorry about that. We got a situation. Can you code in AGGIE?"

Cheng Xin let loose a long, low, satisfying hackle.

"For 8.7 billion dollars a year, you bet I learned AGGIE. Jack me in."

As Cheng Xin approached the REDS station, the captain turned to PJ.

"8.7 billion? That's better than supermax. Who is her agent?"

"*El Jefe*," said PJ.

"She got a bag," said Graybeard.

"8.7 billion is a bag," said PJ.

Graybeard shakes his head, "I shoulda never left Texas."

Reading Notes

- Humans born on Mars will likely be very tall because of the reduced gravity.
- Martian children will become competent systems engineers out of necessity.
- History has proven that humans are most creative when they are young. Training them to use engineering tools at younger ages can lead to fantastic solutions to difficult problems.
- Using tools with a common abstraction domain facilitates rapid and efficient teamwork under pressure.
- Using tools with an uncommon abstraction domain can be dangerous or fatal.
- The author was taught how to code in a NASA summer camp in 1985. 37 years after the camp, he found out that it was funded by Texas A&M and taught by an Aggie PhD.
- When you have a challenging engineering problem, who should you call and what should you say?
    - "Houston, we have a problem."

REDS

The Recursive Engineering Design System (REDS) is a system for designing digital systems. A fully integrated toolset is used to abstract the entire hierarchy of a system, down to the source level of the logic. It is a single tool for design, test and debug of digital systems. REDS is recursive, as designs and architectures evolve, it maintains a common abstraction framework.

It's the holy grail of computer aided design. The fundamentals of digital logic do not change. They are constant in simplicity. The complexity is achieved by scaling AND, OR, NOT. What changes in the abstraction model. REDS will leverage the fact that all modern engineering systems are foundationally digital systems and build a common abstraction model around the core simplicity.

Integrated Development Environments (IDEs) allow rapid development of complex software applications. REDS will attempt to extend this same kind of leverage to system design.

Rather than try to devise commonality from the millions of design flows that exist in digital engineering today, we'll start with a fresh slate. The current generation of students have used software and digital logic their entire lives. It's ludicrous to think that engineers that were born in the time of slide rule could optimize a design system for the present or the future. REDS is tomorrow's system.

Education System (Vertical Axis)

REDS is a work-study program. The goal is to evolve and implement an integrated engineering design system. Sponsored by industry, REDS will develop both the tools and the engineers needed to design advanced sustainable systems for use both here on Earth and beyond. REDS will implement a recursive curriculum, so that the competency of new engineers in industry is advanced at a measurable rate.

REDS begins with a middle-school camp (MSC), where students are taught the fundamentals of digital design and the primary tool to do it - code. After completion, students can complete additional course work and become qualified for the high-school work-study program. The camps will be fully sponsored and look to apply the most local leverage.

The high school work-study program (HWS) is an employment program run by sponsor companies. Student workers perform lab experiments and are driven to advance their skill level directly in measurable and productive ways, with wage increases to match their achievement. The program will be part-time, after school and on weekends. This is a merit-based system measured by demonstrated ability. Students who show leadership participate as mentors for the summer MSC.

The University Methodology Lab (UML) is a partnership between sponsor companies and a local University. The UML develops and validates the tools and methodology to measure a set of engineering problems.

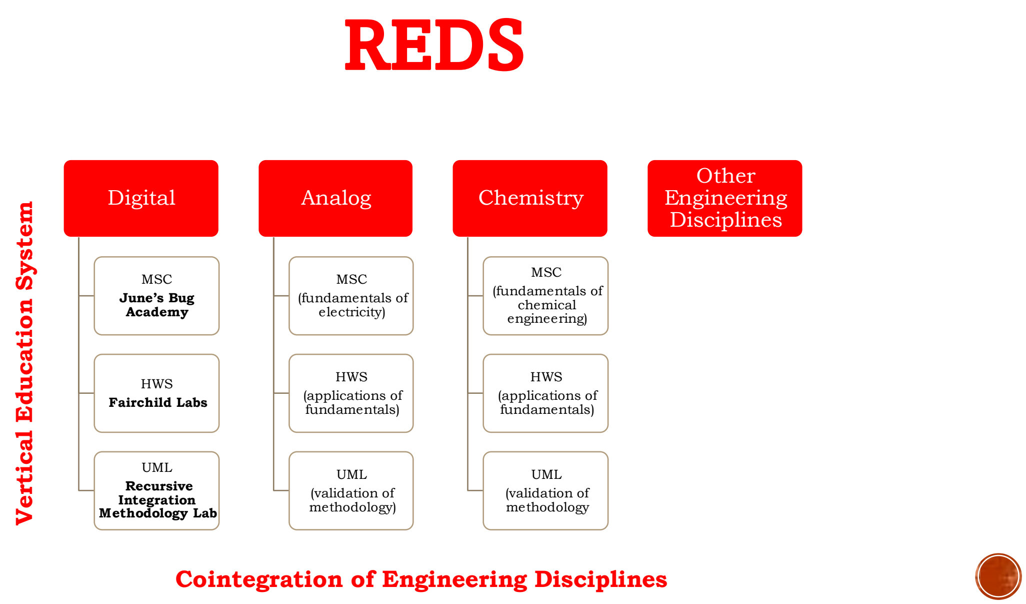
Forced Recursion

At every level, participants are expected to improve and evolve the REDS toolset while using the toolset to optimize systems. One year, students might automate test execution and analysis. The next year, the next batch of students might be expected to automate debug. Each new batch of students should try to make redundant the process work of the last class.

The goal is that engineers entering the profession through the program bring in skills that re-leverage their employers and push their senior engineering peers to higher levels of leverage – or out the door. This prevents stagnation and forces innovation – or failure.

I have not failed. I just found 10,000 ways that won't work.
Thomas Edison

Fail faster is the is mantra of modern engineering. Today, with modern tools and the right approach engineers can fail 10,000 times in the blink of an eye.



Cointegration (Horizontal Axis)

All modern engineering is digital engineering. In all disciplines, engineers depend on software and digital logic to solve problems. If I see the flight engineer break out a slide-rule, I'm getting off the plane. What our predecessor engineers accomplished without digital integration is amazing. We should challenge ourselves to push the boundaries of engineering even further by fully leveraging the tools we've inherited. Moore's Law is virtually free leverage for every profession. For engineers, it is a responsibility.

REDS will establish itself as an educational franchise, bringing together disparate engineering disciplines under the common core of digital integration.

By focusing the common student initiatives around a fully integrated design system, we can develop the engineering talent, tools and systems we need to build a self-sustaining colony beyond Earth. Sustainability is the key engineering challenge in this age and is an absolute requirement for outposts beyond our atmosphere. This is primarily a marketing angle, to get students excited and inspired by STEM careers. If there is an opportunity to integrate with actual space exploration and colonization enterprises, then the students should drive that engagement. The sponsor companies' interest should be sustainable engineering efficiency today, here on Earth.

REDS will partner with engineering leaders in all disciplines, establishing multiple vertical tracks. The MSC core curriculum will be digital logic with sponsor specific labs and projects. HWS programs focus on tactical and relevant work from the sponsors. The UML could be company or industry sponsored.

Constant Integration (Depth Axis)

The third axis of the program is constant integration. A single sponsor may choose to have multiple MSC, HWS or UML for different phases of system design. For instance, a chipmaker may have a complete vertical in Texas that focuses on architecture and validation, while another vertical in China that focuses on design integration and verification. Within a large metro area, a sponsor may have multiple disciplines with multiple local schools and universities.

The horizontal axis focuses on commonality among engineering disciples and systems, the depth axis focuses on integration with a single design process. It's a way for sponsor companies to develop talent through a common framework. Companies cannot realistically expect to source all their new engineering talent from one geography or program, but it does set an internal standard of expectation for hiring as well as a guidepost for other educational institutions that want to refine their target curriculum for industry. The key measurement is the rate of recursion.

Recursive Value Measurement

Following engineers from early education through their professional careers and measuring their skill level with common tools gives both educators and sponsors access to a powerful data set. You cannot judge the result of any educational program for young people deterministically while they are young. You must see the results of application of that knowledge and skills once they are productive adults. What we measure while students are in school is how well they retain things in their short-term memory – primarily, the ability to cram for a test. An engineer's real leverage is how they can apply things broadly. Education provides a foundation, the real measure of value is how that skill set and knowledge in retained, applied, and recursed upon – either through improvement, or by passing their knowledge and experience on to the next generation.

A recursive constantly improving toolset forces an innovation challenge through the engineering ranks of a sponsor. A recursive educational system forces application of that leverage upon the sponsor's management and their educational partners. Long term, if you are improving your workforce's engineering skill level at a measurable rate, you should be able to measure improvement with both business and academic metrics.

Recursive Education Development System

This system also forces recursion upon the education system.

My engineering mentor, Steve Fairchild, taught me some very basic rules about methodology applied to digital systems. The most important was, when any part of digital system changes, you must remeasure all parts of the system again. This is because digital systems design was all about ratios. When ratios change, system behavior changes.

The approach, I call *recursive integration*, makes it easy to debug and understand complex digital systems. This methodology itself almost always leads to simple optimized solutions. "*Recursive integration,*" is really jargon so the boss doesn't have to tell his boss, *"We fail as fast as we can and measure it so we learn to fail faster."*

It's easy to apply recursive integration to human systems problems. Applied in the profession digital engineering work environment, it would work like this; The boss only gets his raise, bonus and equity if the engineer working for him can do the bosses job by the end of the year. Engineering majors don't need to worry, this is purely hypothetical, in the real world, nobody's boss is stupid enough to approve this methodology.

That's why the education system is the way to go. You set a goal for a recursion level. Is it one year? Depends on the subject. In engineering, and other digitally leveraged scientific endeavors, it should be very rapid.

An education system to validate a colonist population for another planet is the perfect example. Why? Their lives and all the lives on the colony depend on it. There is no "Mother Earth System" to provide a backup for human children left alone with just machines to care for them. That's the challenge.

Because digital systems are the primary leverage tool of all engineers, an engineer's leverage at entry level should progress in step with Moore's Law. The three-tiered system provides a checkpointed process with multiple measurement points to ensure that it does.

Recursive Responsibility

Is this a brutal way to look at education? Not all disciplines of education leverage Moore's Law as heavily as system design, but almost all do to some extent. It's a constant factor in every aspect of education.

The challenge of Mars is a good test ground. Every potential Mars explorer and colonist needs to be a highly competent, fully qualified systems engineer. Their lives depend on it.

Brutal or not, this is real. We need simple systems that trained engineers can troubleshoot, repair, or redesign, on the fly, with zero help from the mother planet. Systems will fail, without engineers to fix them, people will die. The time or cost of Earth rescue may exceed what Earth is willing to spend.

Those systems – and the tools they use to design the systems -- should be the responsibility of the people who put their lives at risk. This is the exact same logic that the Mercury astronauts used to convince NASA to put manual controls in the Mercury spacecraft. Without ownership of that responsibility, Gordon Cooper and *Faith 7* would never have been able to return to Earth.

Engineers are responsible for the tools they use.

Recursive Risk Assessment

Can we build systems to explore and colonize Mars? That's a silly question, of course *we can*.

Should we build systems to explore and colonize Mars? At what cost?

The cost of resources, both public and private, is of great concern, both for capitalization of the endeavors, and as a measurable way of factoring the greatest expense, human lives.

Ultimately, good engineering is the optimization of designs for the most efficient systems. This is real green engineering. The metric of value for REDS is not cost efficiency, but resource efficiency, which is the required optimization when resources are limited. On Mars or on Earth, human lives are our most valuable resource. Avoiding waste and conserving resources so that humans can thrive is the optimal solution, no matter what planet you're on.

What are we willing to spend and risk to explore our solar system? The engineers that risk their lives in these endeavors need to make that evaluation, using engineering methodology and risk metrics they develop. Exploration of any kind is an inherently risky endeavor, and it has costs.

Common Abstraction Domains

To understand REDS, we can look at science fiction. When the crew of Star Trek's *Enterprise* encounter any kind of systems problem, be it communications, life support, transporter, weapons, or the warp drive, the chief engineer can go to any workstation, and within a few seconds, instantly locate the problem and usually fix it or work around the failing component. It's a common plot device, saving the average viewer from having to understand the science and engineering behind the systems. Even fictional series set in the present use these kinds of systems. It's common for a super spy to call up a nerdy hacker who can instantly get access to any bank account, every surveillance camera and even into the control room of nuclear power plants, from the same magic software application.

While these are fantastical fictional devices, they aren't far-fetched. When you reduce digital systems down to their simplest components – logical AND, OR, NOT operations -- they can be viewed through a common abstraction. Digital systems are inherently hierarchical, with components connected by data flows. The software abstractions in between are also easily represented with plumbing, and in fact, many complex application use "pipes" for inter-process communication.

Yet in practice, modern digital design is a crowded field of specialization in specific tools. In modern chip design, it's not uncommon to have dozens of engineers within the same organization looking at data with dozens of different tools. These abstractions are useful to each in their own way, but prevent leveraging the commonality. A common abstraction domain is the solution.

First Recursion (Houston)

REDS Franchise

The REDS parent organization is a non-profit that provides guidance, resources, and support to students seeking to develop REDS programs in their local communities. REDS will be a participant run organization, funded by individual donors in the communities where it operates.

Creative Expression Contest

Initially, REDS will promote the concept through writing and creative expression contests. Students will be challenged to consider problems and solutions, both technological and ethical, that need to be solved for a viable space colony. REDS will assemble of panel of writers, educators, and engineers to pick winning entries. Awards will be in the form of cash and scholarships. Winners and their schools will be provided resources, support and starter funding to enable development of REDS programs in their local communities.

Digital Design Education System- Houston

The first implementation will focus on architecture, which begins with proper characterization and modeling of the engineering problem a system is designed to solve.

June's Bug Academy (MSC)

Moto: ***If you wanna ad astra, gotta fail fasta!***

A summer camp for incoming 9th graders.

Mars Theme

The camp will partner with NASA Johnson Space Center, so that students would get a firsthand look at what the complex systems that manage spacecraft today look like, as well as the engineering manpower and specialization that goes into modern space mission management.

At the end of the camp, they'll understand to have a self-sustaining colony on Mars, these types of systems need to be engineered to a level so that every single person on Mars – even a 9th grader who might have been born there – can operate, design, and debug those systems, on the fly, with little or zero help from Earth.

Prerequisites

Basic Boolean Algebra screening after a single lesson taught by participating 8th grade teachers. REDS will provide the lesson plan and materials. Teachers recommend students who are self-motivated and have the focus to learn a challenging skill.

Curriculum

The students will learn the fundamentals of digital logic through direct application using the C programming language. They will learn the fundamentals of memory operations, including pointers, function calls and the call stack. They will learn and apply mathematical, logical and bitwise operators, and control loops. Students will design, debug, and test their acquired knowledge by programming digital hardware and compete for the best solutions to challenging problems. The projects should be fun, such as robotics or drones, but might be real R&D projects from sponsors.

Fairchild Laboratories (HSW)

An industry supported lab where high-school students break down digital systems problems, characterize them through advanced testing and instrumentation techniques, build mathematical models, and then implement virtual and material models of optimizations and solutions.

Students begin with basic lab work, running tests. As they progress through the breakdown process, they will earn skill-based promotions that directly translate to hire wages. Each incoming rank of students will be expected to recurse upon the REDS toolset and improve it so that the productivity – and wages – of the next incoming rank are leveraged to a higher level.

The long-term goal of Fairchild Lab is to be a self-sustaining enterprise. The lab will investigate specific problems from sponsors. Workers may propose derivative experiments based on insight derived from their assigned work. Students will be taught to seek out problems to solve.

Recursive Integration Methodology Laboratory (UML)

The university lab will perform graduate level validation of the tools, techniques, and teams in the HWS, and look for broader and specifical application of innovations driven through the lab. They will duplicate and correlate the results of the HSW, validating the results with mathematical proofs.

The goal is to get academics competing for grants from sponsors of the HWS and MSC programs based on data generated by those programs.

REDS Advisory Panel

Advisors have read the proposal and provided direct feedback and wish to be involved in the project, either directly or as advisors, as the project evolves. There is no assumption of commitment beyond that. A formal panel will be formed with a schedule and agenda once minimum viability funding is secured.

Brandon Awbrey
Carol Fairchild
Jim Wells
Pam Wells
Justin Allen
Paris Tompkins
Tom Woodard
Mike Zandy
Mark Heerema
Ishaq Unwala

Implementation

REDS

Governance

REDS shall be formed as non-profit 501(c) corporation. REDS will seek funding from individuals who become sponsor members. A board shall be elected by the sponsor members, which each member having one vote, regardless of their contribution level. The board shall serve as a proxy for an elected Shadow board of students. The Shadow Board will be elected by participating students of all REDS programs. The Shadow Board shall set the agenda, form a budget, seek additional funding, and expand the program.

Fairchild Labs

Sponsor Expectation and Responsibilities

Sponsor companies are expected to provide challenging and tactically applicable real-world tasks and experiments so that students can be taught real world engineering skills in working lab environment. The measure of success of the lab to the sponsor is the quality and value of the work the students provide through Fairchild Labs.

There is no expectation for the sponsors to provide or facilitate employment of the students after they complete the program. The leverage of a community's skilled young adult workforce is the community's responsibility. Businesses are cyclical and dynamic, demands and requirements change with the economy and technological trends, but educating and providing employment and entrepreneurial opportunity is a permanent responsibility of the local community.

The local community's responsibility is to build a talent base and create a competitive marketplace for the skills of the local workforce, driving up wages and lifting the entire community to a higher standard of prosperity.

Mission

To teach students valuable digital design skills in a working lab environment, while providing employment and a local engineering talent development pipeline.

Goal

To reach majority funding self-sufficiency within two years.

Governance

Fairchild Labs shall be formed as a non-profit 501(c) corporation, with board seats assigned to each of the founding sponsor organizations. There shall be a board seat representing the interests of the students filled by a non-sponsored member of the local community.

Schedule

Provisional schedule would be four-hour shifts, Monday-Thursday, likely from 4pm to 8pm, as well as a Saturday shift. Students are expected to commit to a minimum schedule, and likely one day (Monday) would be required for all participants for weekly team planning.

Saturday shifts will consist of a formal programming class of two hours followed by a work session.

Transportation

During the sponsored funding period (first two years) the plan is to provide professional transportation after school to Fairchild Labs facility, and home (within the school's normal boundaries) after work.

Parent and Teacher Engagement

<<Looking for advice here: My thoughts, working at Fairchild Labs will be just like any part time after school job – McDonalds or Chick-fil-A or whatever; quality and focus is required. However, the mission is to help these students develop useful skills and opportunities. How do we balance the expectations of work versus the mission; What level of involvement can we expect from Parents? Should we make continued employment conditional on grades? >>

Timeline

Planning periods align with school Fall and Spring semester, using "season" in same context as team sports. It's a part-time job, with the same fixed schedule for all participants, with reasonable accommodation for high school students. The Lab will have the same expectations for work ethic, respect and behavior of any local employer.

Date	Item
March 2024	Fill Advisory Panel
April 2024	Seek initial funding commitment
May 2024	Planning, feasibility recruiting
Summer 2024	Possible 1-2 student training, shake down methodology. Lab setup Equipment sourcing Training
Fall 2024	6 Student feasibility work-program "season"
Spring 2025	12 Student season
Summer 2025	Expanded hours work program
Fall 2025	12 Student season
Winter 2026	Self-sustained funding

Operation

The students will start with the basics, Assembling computer hardware, installing operation systems, configurating applications, and running benchmarks. The first test is correlation; students will compare test results and breakdown the differences between systems. They'll learn to use software tools to visualize the data they collect, and statistical techniques to isolate unexpected and unusual results. At each step, they'll be encouraged to optimize the process and tools, while constantly correlating their results to expectations.

In mainstream R&D, we continually encounter data that is unexpected but not pertinent to the problem we are actively investigating. Often we note these cases as curious, and hope to come back and investigate later, but because of operational demands, they remain mysterys. Fairchild Labs will investigate all unexpected conditions, not just for knowledge, but as a way for students to improve their problem solving skills.

Advancement

Students will be taught skills in a lab environment, working on real digital systems problems. The goal is to advance both their skill and pay as rapidly as possible. Promotion will be based on merit and demonstration of skill. Advancement will not be based on age, school grade level, or time in program. As the toolset improves, the expectation is that we can raise both the expected engineering leverage and wages of the participants entering the program.

Level	Digital Systems Architecture Skills
1	Basic workplace skills; Build and Configure computer systems; Run correlation test; Manage data sets; Analyze data and generate visualization of results; Generate reports; Basic troubleshooting of HW and SW; Presentation of results to sponsors;
2	Basic scripting; Advanced analysis; Advanced visualization; Breakdown of systems; Implementation of mathematical models; Propose design improvements;
3	Basic proficiency in C and C#; Integration of lab process into REDS; Advanced statistical analysis; Advanced troubleshooting. Simulation and emulation of systems;
4	Intermediate Proficiency in C or C#; Advanced debug; Advanced emulation and simulation; Propose new designs;
5	Lab Ready Functional Engineer

Budget

This budget is a rough planning budget for funding commitments. A formal operational budget will be presented to the board before operations begin.

Two Year Summary Funding Targets

Plan	Students	Total Cash Outlay	Targeted Founding sponsors	2 year per student	Matching Goal per Sponsor
Aggressive	17	823,808.00	5	60,361.50	164,761.60
Conservative	11	466,112.00	3	55,378.29	155,370.67
Minimum	7	323,840.00	1	51,734.86	323,840.00

Aggressive Budget

Fairchild Labs - Aggressive 2 year planning budget

Cash Funded Item	Fall Y1	Spring Y1	Summer Y1	Fall Y2	Spring Y2	Summer Y2	2yr Total
Weeks (per semester)	18	18	8	18	18	8	
Hours	20	20	30	20	20	30	
Student Workers	6	11	11	17	17	17	
Wage Per Hour (mid-point)	$15.00	$20.00	$24.00	$26.00	$28.00	$30.00	
Base Wages	$32,400.00	$79,200.00	$63,360.00	$159,120.00	$171,360.00	$122,400.00	
Administrative Wage Overhead	20.00%	20.00%	20.00%	20.00%	20.00%	20.00%	
Total Wage Budget Line	$38,880.00	$95,040.00	$76,032.00	$190,944.00	$205,632.00	$146,880.00	
Transportation Subsidy	$9,000.00	$9,000.00	$4,000.00	$9,000.00	$9,000.00	$4,000.00	
Fixed Overhead (Insurance, etc)	$5,400.00	$5,400.00	$2,400.00	$5,400.00	$5,400.00	$2,400.00	
Total	$53,280.00	$109,440.00	$82,432.00	$205,344.00	$220,032.00	$153,280.00	$823,808.00

In Kind gifted by sponsors	Fall Y1	Spring Y1	Summer Y1	Fall Y2	Spring Y2	Summer Y2	2yr Total
Dev PCs	$6,000.00	$5,000.00	$0.00	$6,000.00	$0.00	$0.00	
DUTS	$12,000.00	$10,000.00	$0.00	$12,000.00	$0.00	$0.00	
Rent (write off)	$9,000.00	$9,000.00	$4,000.00	$9,000.00	$9,000.00	$4,000.00	
Cloud	$9,000.00	$9,000.00	$4,000.00	$9,000.00	$9,000.00	$4,000.00	
Network	$1,800.00	$1,800.00	$800.00	$1,800.00	$1,800.00	$800.00	
Total	$37,800.00	$34,800.00	$8,800.00	$37,800.00	$19,800.00	$8,800.00	$147,800.00
Total Value (In-Kind + Cash)	$91,080.00	$144,240.00	$91,232.00	$243,144.00	$239,832.00	$162,080.00	$971,608.00

Conservative Budget

Fairchild Labs - Conservative 2 year planning budget

Cash Funded Item	Fall Y1	Spring Y1	Summer Y1	Fall Y2	Spring Y2	Summer Y2	2yr Total
Weeks (per semester)	18	18	8	18	18	8	
Hours	20	20	30	20	20	30	
Student Workers	5	5	7	11	11	11	
Wage Per Hour (mid-point)	$15.00	$17.00	$19.00	$21.00	$23.00	$25.00	
Base Wages	$27,000.00	$30,600.00	$31,920.00	$83,160.00	$91,080.00	$66,000.00	
Administrative Wage Overhead	20.00%	20.00%	20.00%	20.00%	20.00%	20.00%	
Total Wage Budget Line	$32,400.00	$36,720.00	$38,304.00	$99,792.00	$109,296.00	$79,200.00	
Transportation Subsidy	$9,000.00	$9,000.00	$4,000.00	$9,000.00	$9,000.00	$4,000.00	
Fixed Overhead (Insurance, etc)	$5,400.00	$5,400.00	$2,400.00	$5,400.00	$5,400.00	$2,400.00	
Total	$46,800.00	$51,120.00	$44,704.00	$114,192.00	$123,696.00	$85,600.00	$466,112.00

In Kind gifted by sponsors	Fall Y1	Spring Y1	Summer Y1	Fall Y2	Spring Y2	Summer Y2	2yr Total
Dev PCs	$5,000.00	$0.00	$2,000.00	$4,000.00	$0.00	$0.00	
DUTS	$10,000.00	$0.00	$4,000.00	$8,000.00	$0.00	$0.00	
Rent (write off)	$9,000.00	$9,000.00	$4,000.00	$9,000.00	$9,000.00	$4,000.00	
Cloud	$9,000.00	$9,000.00	$4,000.00	$9,000.00	$9,000.00	$4,000.00	
Network	$1,800.00	$1,800.00	$800.00	$1,800.00	$1,800.00	$800.00	
Total	$34,800.00	$19,800.00	$14,800.00	$31,800.00	$19,800.00	$8,800.00	$129,800.00
Total Value (In-Kind + Cash)	$81,600.00	$70,920.00	$59,504.00	$145,992.00	$143,496.00	$94,400.00	$595,912.00

Minimum Budget for Viable Program

Fairchild Labs -Minimum 2 year planning budget

Cash Funded Item	Fall Y1	Spring Y1	Summer Y1	Fall Y2	Spring Y2	Summer Y2	2yr Total
Weeks (per semester)	18	18	8	18	18	8	
Hours	20	20	30	20	20	30	
Student Workers	5	5	7	7	7	7	
Wage Per Hour (mid-point)	$15.00	$16.00	$17.00	$18.00	$19.00	$20.00	
Base Wages	$27,000.00	$28,800.00	$28,560.00	$45,360.00	$47,880.00	$33,600.00	
Administrative Wage Overhead	20.00%	20.00%	20.00%	20.00%	20.00%	20.00%	
Total Wage Budget Line	$32,400.00	$34,560.00	$34,272.00	$54,432.00	$57,456.00	$40,320.00	
Transportation Subsidy	$9,000.00	$9,000.00	$4,000.00	$9,000.00	$9,000.00	$4,000.00	
Fixed Overhead (Insurance, etc)	$5,400.00	$5,400.00	$2,400.00	$5,400.00	$5,400.00	$2,400.00	
Total	$46,800.00	$48,960.00	$40,672.00	$68,832.00	$71,856.00	$46,720.00	$323,840.00

In Kind gifted by sponsors	Fall Y1	Spring Y1	Summer Y1	Fall Y2	Spring Y2	Summer Y2	2yr Total
Dev PCs	$5,000.00	$0.00	$2,000.00	$0.00	$0.00	$0.00	
DUTS	$10,000.00	$0.00	$4,000.00	$0.00	$0.00	$0.00	
Rent (write off)	$9,000.00	$9,000.00	$4,000.00	$9,000.00	$9,000.00	$4,000.00	
Cloud	$9,000.00	$9,000.00	$4,000.00	$9,000.00	$9,000.00	$4,000.00	
Network	$1,800.00	$1,800.00	$800.00	$1,800.00	$1,800.00	$800.00	
Total	$34,800.00	$19,800.00	$14,800.00	$19,800.00	$19,800.00	$8,800.00	$117,800.00
Total Value (In-Kind + Cash)	$81,600.00	$68,760.00	$55,472.00	$88,632.00	$91,656.00	$55,520.00	$441,640.00