

EPISODE 637

[INTRODUCTION]

[0:00:00.3] JM: Howard Marks ran two video game companies in the 90s; Activision and Acclaim. While running these companies, Howard developed a love for entrepreneurship that he maintains today. Howard is the CEO of StartEngine, a company that functions as an accelerator, a crowdfunding platform and an ICO launcher. Howard joins the show to talk about his background as an entrepreneur, as well as some of the modern alternative funding mechanisms that he's working on at StartEngine.

Hearing Howard's thoughts on building a video game company in the 90s was particularly new information to me. That's an era of software development that we have not covered much at all in Software Engineering Daily. As a side note, some listeners have asked recently why we cover subjects such as ICOs when there have been so many dubious companies that have launched with an ICO. There are two reasons why we cover this area. Firstly, cryptocurrencies are a breakthrough computer science construct and it's important for us to try to understand their implications.

Now that said, the type of ICO that we talk about in this episode is not closely related to cryptocurrencies, but it is an alternative funding mechanism and the other reason why we cover ICOs is that some technology companies require high upfront capital costs, and the amount of capital that you have available in your business affects the speed at which your engineering team can move.

New funding mechanisms can mean more capital for certain types of software companies, and this could be a good thing, or a bad thing depending on the company. In any case, if you're building a company as many people who are building new technology and new software are, software engineering and company building are almost inextricably linked, it's useful to know about the different tools available to you for getting money, because that money can be converted into new code and software and technology, so that's why we do these kinds of shows.

Before we get started, we're hiring a creative operations lead. If you are an excellent communicator, please check out our creative operations role at softwareengineeringdaily.com/jobs. This is a great job for someone who just graduated a bootcamp, or someone with a background in the arts who's making their way into technology, or somebody who is coming out of software engineering and they aren't sure that software engineering is exactly what they want to do. Maybe you want to do something more editorial. If you want to be creative and you want to learn about engineering, you can check out our creative operations role at softwareengineeringdaily.com/jobs.

[SPONSOR MESSAGE]

[0:02:57.0] JM: Citus Data can scale your PostgreSQL database horizontally. For many of you, your PostgreSQL database is the heart of your application. You chose PostgreSQL because you trust it. After all, PostgreSQL is battle-tested, trustworthy database software.

Are you spending more and more time dealing with scalability issues? Citus distributes your data and your queries across multiple nodes. Are your queries getting slow? Citus can parallelize your SQL queries across multiple nodes, dramatically speeding them up and giving you much lower latency. Are you worried about hitting the limits of single node PostgreSQL and not being able to grow your app, or having to spend your time on database infrastructure instead of creating new features for your application? Available as open source, as a database, as a service and as enterprise software, Citus makes it simple to shard PostgreSQL.

Go to citusdata.com/sedaily to learn more about how Citus transforms PostgreSQL into a distributed database. That's [C-I-T-U-S-D-A-T-A.com/sedaily](https://citusdata.com/sedaily), citusdata.com/sedaily. Get back the time that you're spending on database operations. Companies like Algolia, Prosperworks and Cisco are all using Citus, so they no longer have to worry about scaling their database. Try it yourself at citusdata.com/sedaily. That's citusdata.com/sedaily.

Thank you to Citus Data for being a sponsor of Software Engineering Daily.

[INTERVIEW]

[0:04:41.9] JM: Howard Marks, you are the CEO of StartEngine. Welcome to Software Engineering Daily.

[0:04:45.9] HM: Thank you. Happy to be here.

[0:04:47.8] JM: We'll get to talking about StartEngine, which is a crowdfunding platform, a company that you're working on today. I did want to talk some about your history, because you worked in game company development in the early 90s and through the early 2000s. The first company you were involved with was Activision. Can you describe the software engineering that was involved for game development in the early 90s?

[0:05:13.1] HM: Absolutely. Games used to be made on cartridges, the most famous was Atari. That console allowed you to play games in your home and cartridges were these plastic devices that had chips in them. Typically the way you made a game, one person would actually write a software, the same person would do the graphics and the same person, the sounds and the design and everything. You would have a game like Pitfall, was built by an engineer called David Crane all by himself. It's unbelievable. It really is. These are great games, by the way.

When I got involved and that was the early 90s, 91, it turns out there was a shift happening, which was called a CD-ROM. The CD-ROM was an external device that you would plug into your PC at that time. It was not incorporated, it was not a DVD, it was really the early stages of the first I would say plastic optical media. What was amazing about it is that you could store 800 megabytes. Now a cartridge, you could store maybe a megabyte, sometimes up to 16 megabytes and the biggest cartridges ever were 32 megabytes, but the costs of making them were too high.

Software engineering had to change when the CD-ROM came out. The reason is because you have so much more storage. How do you build a game that instead of having a footprint of a few megabytes, had hundreds of megabytes? Then you have a limitation inside of the PC, which the microprocessor limited you to 640k of memory, where most games needed more memory. One game we had called MechWarrior, which was a 3D fighting game with robots.

These mechs that you would configure had a problem, because the new chips that were coming out would allow you to go beyond these small memory footprints. Unfortunately, you had to change all the software and we recompile everything into 16-bit, instead of 8-bit. At that time, it looked like a big challenge in terms of making these changes, but it was not a big deal. We had to hire engineers who understood 16-bit architecture, who were willing to go with the new compilers that worked with these new upgrades.

[0:07:45.4] JM: This was back when the end product was – the finished console – finished product was a console game, or a CD game. This is a very different mode of software engineering than developing this continuously changing internet software that we have today. How does the engineering process differ in a world like that, where it's almost producing a movie? You work on this project, then it's a finished product and turns out the other side and then you don't really adapt it as much, unless you make a version 2, or a sequel or so on. How does that compare to the ever changing internet software engineering process?

[0:08:25.0] HM: Well actually, it's a good question you have here, because it's actually very scary. At that time, when we made a game whether for cartridge, or for CD and you printed it, you're done. That's it. You have a bug that creeps in and stops the game from functioning, you're done. The idea of a patch, how do you even distribute a patch? There was no internet to send it around.

At that point, people were used to getting these diskettes. They were shared between a community of players, where they would patch things. In reality, we had to do a lot of quality assurance to make sure that the end product was as good as possible, so that it could be printed, manufactured in hundreds of thousands, if not millions of quantities, delivered by truck to the retailers like Walmart. I mean, you have to imagine, this is a big infrastructure, which today doesn't exist because everything is digital.

[0:09:28.3] JM: How many engineers did you need to ship a game like MechWarrior 2? Obviously, with one person you could make a game like Pitfall, but a 3D fighting game is quite different.

[0:09:41.8] HM: Absolutely. I think what Activision did at that time and the innovation we brought to the table in order to become successful was we recognized that the budgets had to change. It had to become a much bigger cost to making a game than in the past. We looked at this and said, “Whoa, what if we spent a million dollars on a game?” I mean, today that looks like a little tiny budget, but at that time, most games were being built was under \$200,000 in costs. MechWarrior was a big-budget game. We had a lead engineer, we had a producer and director, then we had a lead engineer who was in charge of all the engineers and that's the 3D engine that we built ourselves. All of the game, I would say logics, inside the game and we have a map building tools, then we had artists building the 3D graphics and we had artists building just the textures. It was a team of 20-plus people. It's one of the first early, I would say large game teams. That today, there are hundreds of people and spending on upwards of a hundred million dollars.

At that time, it was really smaller, but yet very ambitious. We even brought in engineers who work for NASA, JPL who knew 3D graphics the way it was built for science. We had to teach them how efficient you need to be when you make a game.

[0:11:18.8] JM: The business of making games, even back then it seems like a difficult treadmill to be on, because in contrast to something like business application software where you can develop relationships with the company that you're selling the business software to, maybe you sell them a license, maybe it's a business model that grows with the size of the businesses who are using your business applications. With games, you don't have that lock-in, you don't have the same economics you do with business application software. Why did you go into games when you had these – there were other opportunities for people to build business software?

[0:12:02.4] HM: Absolutely. In fact, we were building business software before games. We were building business software for personal computers, such as the Commodore Amiga, the PCjr and all these different small computers that people were purchasing for their homes, and we were building business software. We were not that interested in that. Frankly, that was not our primary goal. Our primary goal was to have consumer software in the hands of consumers and not businesses.

The idea of being a business-to-business business is a different feel. I guess for us, the consumer was more appealing, because you have a direct relationship with millions of people. Games fit the paradigm very well. In fact, games started becoming more and more, I would say complex in how they were built. I would say games are as state-of-the-art of any software you could look at.

I mean, business software typically is answering a need out there that you find and then you have a long sales cycle, and then you have lots of iterations to build a new version. I mean, salesforce.com is a great example of a business software company that evolved over the years into a mega-corporation, supporting hundreds of thousands of companies. As a game today, I mean, you have games today. In China especially, they have 50 million players a day, that's a pretty significant piece of software.

Can you imagine that the server architecture? The security systems? If you look at League of Legends, which is another amazing game published by Riot Games here in Los Angeles, I did a tour of that company recently with my kids. It is extraordinary. I mean, their war room looks like you would be at NASA. I mean, it's unbelievable.

[0:14:00.2] JM: Well, they have these crazy networking issues in the games, where you have people on phones, maybe not with League of Legends. I think that's a desktop game, but some of these games, like the Chinese games, we did one show about this and they have these issues where if you're trying to make a real-time strategy, or fighting game for phones, you have this internet network connection and you're trying to have a fighting game. You know what I mean? You know Mortal Kombat, which we'll get into, but that's a real-time fighting game.

Luckily, you're on the same machine in that context, but with the networking issues over cellphones, if you have a drop off, then it's how do you resolve that networking conflict that can develop? When somebody reconnects, do you just have them die because their opponent was able to strike them while they were disconnected? The engineering problems gets so much more complex with some of these modern games.

[0:14:53.3] HM: Well absolutely. If you add to that the security issues, because people are using bots and they're cheating all over the place, which kills the game, because it's no longer a level

playing field for the players. I mean, there's nothing worse in a game when that the cheaters are winning. In the mobile games, it turns out the mobile games have a little bit of a different problem than the PC games, because their ping time is much higher than a desktop game, which can connect directly to the internet using an rj45 cable.

Whereas, the mobile phone you're using the phone networks and the phone networks are cell towers and the cell towers sometimes have latencies that are pretty significant. I would say the problem on the phone is not as real-time as what you have a desktop. It would be more of a turn play. They call them real-time games, but they're not really real-time, they're turn-based games that look real-time. I don't know if you see, there's a fundamental difference. Whereas, on a desktop with a low ping time, let's say 50 milliseconds, you can pretty much play a real-time game.

[0:16:02.3] JM: Back in those days in the Activision days and in the Acclaim days, we'll get to a Acclaim in a second, but were there frameworks and libraries to build off of, or are you just writing everything from scratch?

[0:16:17.2] HM: Well, that's a good question because there were some libraries, but the libraries that we were using were mostly for the installer, so you can install the game. You didn't have to rewrite the installer every time. These are libraries that we licensed and purchased. In reality for the game itself, the 3D engine, we were building our own 3D engine. The reason was is that the processor and the memory size of the computers were pretty limited and there were 16-bit computers with limited memory, maybe two or three megabytes of memory.

We had to be super-efficient. If you bought a generalized 3D engine, you probably will find a low framerate by the time your game is done, and that kills the game. We had to cheat by creating our own 3D engines that were not very good in terms of display capabilities, but were superfast. We would realize that in a scene where you have your, for example your MechWarrior running around and fighting, we would give up a lot of features, so that we can still have a good framerate.

[0:17:26.2] JM: You bought Acclaim Games. This was a company that you bought after you left Activision. Acclaim at the time owned the rights to NBA Jam, Mortal Kombat, these are games

that will be familiar to most of the listeners and there are also several other games you got. You bought it for a 100K, which is a pretty low cost for some really strong brand assets. That was because the company was bankrupt and there was nobody else bidding for it. I think you got lucky, because you bought it during this period of time where everybody was on holiday, or something when the auction happened.

When you made that a 100K purchase for this games company, what exactly did you get? What were the assets that you got? Did you get developers? Did you get any real estate assets, or was it mostly just brand assets?

[0:18:20.3] HM: Well actually, I didn't know what I was getting, because the deal that I have from the Bankruptcy Court was an as is purchase, which means we don't know what we have, but you're buying it. It sounds a little strange, but when a company goes into liquidation, typically you would expect, "Oh, we know all the assets. We know how much inventory there are, there is. We know if there's a payroll. We know the bank statements." A lot of the things should be discoverable. Well, this bankruptcy was such a mess that even the trustee of the Bankruptcy Court did not know what they had.

I would send them a message saying, "Okay, give me the list of all of your assets." they would reply, "Well, we're not exactly sure if we own this, we don't own that," because don't forget, Mortal Kombat and NBA Jam were owned by Bally Midway. There are contracts, right? It turns out that by time I bought Acclaim, those contracts were voided by, because the company didn't, I think lost the rights, because maybe it termed out, or they didn't follow the right rules.

[0:19:35.9] JM: Bally Midway lost the rights?

[0:19:37.3] HM: No. Acclaim did.

[0:19:39.1] JM: Acclaim lost the rights.

[0:19:40.3] HM: Yeah. When I ended up buying it for a 100 grand, I got acclaim.com, .net, I got the trademarks, I got a few properties, but I didn't get Mortal Kombat, I didn't get NBA Jam, because those once we – when you go to bankruptcy, if you look at the license agreements for

these games, licenses, so you license a name like Mortal Kombat, which was a coin-op game, one of those very famous parlor games. You license the rights to make the game for a PC, or console, if you go bankrupt it says in the contract, if you file for bankruptcy, the rights revert back to the owner. That's pretty harsh.

A company gets into trouble, goes into bankruptcy, well it gets worse now, because you potentially have lost all of the rights for the games you've licensed, because of the bankruptcy. Now it turns out Acclaim had no choice but to go bankrupt. There were so many issues. I just bought it and I realized I had a lot of the rights that were exciting didn't come with it, but that was okay. I was prepared to go into the online game world anyway, so I didn't mind.

[0:20:58.0] JM: The asset you got, so you got the domain names, I guess the brand name of Acclaim. Was it a regrettable purchase, or did the brand – were you able to leverage the brand?

[0:21:10.7] HM: Well, it actually was a very good purchase, because I went and licensed a few games out of Korea and these were online multiplayer games. The brand allowed me to grow into millions of players very quickly, because people remembered the brand. When they would see the Acclaim logo, they were like, “Wow, this is great.” The brand was mostly a, I would say target towards more of a male gamer. Not really a hardcore gamer, but I would say a game fighting culture type of player. It worked well, because those games that I licensed were massively multiplayer games that were a lot of fun, was virtual items.

We innovated in the United States the first games that were free to play. There was no cost to get in and you only bought items inside the game and then he could trade those items. That in a way gave me a great experience for what I'm doing now in the cryptocurrency world, because when you are building these incredible environment-oriented games where you have economies inside of the game, and security issues inside of the games. We had one game called Two Moons and literally we got hacked. We didn't know we got hacked.

The hackers got control of our database and they started building currency. What we were observing was that inside the game, the prices for items were going up fast. We were like, “Why would it go up fast?” It turns out that the hackers were creating their own currency. They became their own Treasury Department, which is unbelievable.

Imagine our treasury in the United States printing more money than we know of and more money goes into circulation, that creates a huge amount of inflation. Well, that's what was happening. Then once we found out, we stopped it, we cured it, but the damage was done.

[SPONSOR MESSAGE]

[0:23:16.5] JM: At Software Engineering Daily, we have a web app, we have an iOS app, an Android app and a back-end that serves all of these frontends. Our code has a lot of surface area and we need visibility into problems that occur across all of these different surfaces. When a user's mobile app crashes while playing a podcast, or reading an article, Airbrake alerts us in real-time and gives us the diagnostics that let us identify and fix the problem in minutes, instead of hours.

Check out airbrake.io/sedaily to start monitoring your apps free for 30 days. Setup takes only a few minutes. There's no complicated configuration needed. Airbrake integrates with all of your communication tools, from Slack, to Github, to Jira and it enhances your current workflow rather than disrupting it. You can try out Airbrake today at airbrake.io/sedaily. If you want to monitor and get visibility into the problems that may be occurring across your application, check out Airbrake at airbrake.io/sedaily. Thank you to Airbrake.

[INTERVIEW CONTINUED]

[0:24:35.7] JM: It is amazing how the virtual economies that develop in games like Second Life, or World of Warcraft, they develop economic phenomenon that are things that economists are fascinated by, or that economists have even documented before you see them materializing in in a game. As you're alluding to, because you've borne witness to some of these strange phenomenon in the context of games, you probably have some prescience when it comes to how tokens and a tokenized world that we could potentially be living in very soon, where many more enterprises have their own currencies. You can have some intuition around how those worlds are going to play out.

[0:25:31.7] HM: Absolutely. In fact, I'm writing an article now, which I haven't finished, but you'll find that a lot of people in the cryptocurrency world came from the game world. I don't know if you know the scandal of this cryptocurrency exchange called Mt. Gox. I don't know if you heard of them.

[0:25:48.3] JM: Of course. Yeah.

[0:25:49.2] HM: It was huge –

[0:25:49.8] JM: MTG online exchange.

[0:25:51.8] HM: Right. It was Magic the Gathering online exchange. That's a game, right? It started as a game exchange where you can trade cards and people start paying in Bitcoin for the cards. It's like, "Well, hold on a second, Bitcoin is more interesting than trading cards." The next thing you know, this thing becomes extraordinary huge hit, overwhelming and they get hacked.

Now I don't think we know exactly what happened there in terms of hack, but fundamentally it's easy to explain. You would say that Bitcoin is a decentralized system and it is to the extent that people are holding their own private key. The minute you relinquish your private key by transferring your Bitcoin to a centralized exchange where they hold the private key for you, actually it's fungible, so they basically have one – theoretically, they have a set of number of accounts in Bitcoin and they have the private keys.

If someone gets hold of those private keys, you've lost your coins. You've relinquished your rights to those coins and you're the beneficiary, but you don't have control anymore. In a way, you went from a decentralized system where everybody holds their own assets to a centralized system in a decentralized world. It's really strange. That's how all these exchanges work.

When they get hacked, whether it's – mostly it's probably an insider who does it, who finds a way in through people who work inside of the company. There's a group, my guess the classic hack is to someone in an inside, someone on the outside and they find a way to hack the whole thing, steal the coins, off they go.

Now Bitcoin, which is weird, Bitcoin records everything. You know there was a hack, because you saw the transfer from this wallet to another wallet, or an account to another account. That is so easy to track. Mt. Gox is a great example of people who came from the game world, built a piece of technology, how they knew how to build it, and next thing you know, it gets hacked, discredited, people lose their money and they're just shocked that this could happen.

I don't think it's shocking. Now if you look at Vitalik Buterin who created Ethereum, he was a World of Warcraft player big-time, whose main frustration was the game company which was Blizzard, Activision Blizzard would change the game the way they want to, not the way he wanted to. He was saying to himself, "Well, if we could have a decentralized way where my assets are in my control, the same ways Bitcoin is in your control, then I don't care, the game company doesn't have any more say." Now that led him to think through the idea of creating Ethereum. You see the game world has a lot of influence in the crypto world.

[0:28:50.9] JM: Yeah. Well, I think just to lend some port to your decision to make games instead of business software, games have been massively influential on my life. I played Magic the Gathering from an early age and it was so influential on how I think about the world. I think it's going to be – we're going to see that more and more people who develop innovative technologies based off of their experiences in games, I want to get into the business that you're building today. StartEngine, it started as actually a technology accelerator. You started with just a portfolio of companies that you were investing in, helping to take to market is a accelerator. People might be familiar with tech stars, or Y Combinator, sort of like that.

Eventually, the accelerator morphed into a crowdfunding platform, and then eventually into a platform where you can also issue your own ICOs. Can you describe the evolution from the accelerator to the crowdfunding company?

[0:29:52.1] HM: Sure. After I sold Acclaim, Acclaim ended up being bought through a company who sold to Walt Disney Company. Clearly at this point, I was done because what am I going to do at Disney? I mean, they were not big in games and they're trying to make games, but there was not really something I was interested in.

I decided to create StartEngine as an accelerator, because I want to help LA become a technology city. I was very frustrated by the way I saw a lot of engineers who came out of Cal Tech, UCLA, USC, Harvey Mudd and umpteen number of universities leave to go to Austin, Boulder, San Francisco, New York and not LA. Activision is in LA. It's a very successful company, thousands of employees. Yet, it's not considered a tech company when it really is a tech company.

I said, "Okay. You know what? Let me invest in teams locally." I set up shop in late 2011 and got hundreds and hundreds of applications of people who wanted to be in our accelerator. We basically invested in 20 companies a year, but it's heartbreaking when you get a 100, 200, 500 applications and you have to only pick 20. It feels very cruel. What are the other people supposed to do? This is demoralizing.

Anyway, even the 20 I picked, 90% went out of business. I made 60 investments, 6-0 and 50 went away. 50, and they couldn't raise any money. Then I realized, the business problem was not helping and mentoring, the business problem was there's access to capital, the barriers to entry were very harsh and difficult. They're biased, because typically VCs like to invest in white males from Stanford who are young. There's ageism, gender bias, geography bias, idea bias. I mean, just the bias, the list is too long.

I decided we got to fix it. How? It turns out that at that time in April 2012, the JOBS Act was voted into law. Now what is the JOBS Act? Most people didn't read it. I did read it. It basically says, that for the last 80 years, the only people who could invest in startups, I mean, I would say exciting startups and high-risk startups were wealthy people. That represents about 5% of the country, so one out of 20 people.

Most of them are late in their careers, are not young people. They invest in things that are real estate and things that are nice and boring and more secure, but they're not really angels. Angels, there are angels in our country, and maybe there's a hundred thousand plus angels, but that's not a lot. Keep in mind, we have five million small businesses in this country and 600,000 new businesses created every year.

There is a an economic problem for these businesses, because the banks won't lend to them, because unless they have a house they can mortgage. Then the VCs are very – they're biased. The angels are not enough of them. Then the only thing left are friends and family. You know what? There's so much you can get out of friends and family. I decided, what if I took the model that you see out of Kickstarter, which is very successful for projects and people who are looking to get donations and transform that into a platform that was purely about investing in getting shares in return, or revenue share, or debt or whatever, interest payments, but something that it was a financial model? With the JOBS Act, I was able to do it.

[0:33:40.6] JM: Yes. There is a certain ethic within at least the United States, around what kinds of things should average Joe be able to invest? Average Joe has maybe \$20,000 in savings, he is working a job and making 80 grand a year, or 70 grand a year, and wants to put some percentage of that towards investments, what should he be allowed to invest in? Maybe 20 years ago, or 15 years ago, or 10 years ago, though things were much more limited, you could only put money into publicly traded companies, or bonds, or real estate, maybe you can buy a car for example or invest in a chicken farm if you want. How did we get there? How do we get to a state where that person is not allowed to invest in more speculative types of investment?

I want to get to the point where we can talk about the ethic prior to the JOBS Act and then what the JOBS Act changed. Then eventually we'll get to talking about ICO is what I think are on the much more racy side of things when it comes to public investing.

[0:35:00.6] HM: It's simple, you have to go back to 1929. We had a huge crash in 1929, because people were able to mortgage when they were buying stocks and anybody could buy stocks. They could borrow up to 90% of the value of the stock, so they could instead of when they invest a \$100 in stock, they only put up \$10, the rest they borrow the \$90, right? That was a way to leverage your investment. That's number one.

Number two, unbeknown to the shareholders, a company could print more shares and sell more shares without other people knowing about it, which created a lot of issues, because that would create – that would reduce the value of shares, because instead of having a million shares, now you have two million. The laws around how all this process worked were, I would say challenging at best.

In order to overcome this every time there's a financial crisis and you'll see that was 2007 again, that's why the JOBS Act came out. 1929 created the security Exchange Commission. The Securities Act of 1933 basically said that you can only raise money from the consumer if you register with the Security Exchange Commission and basically you registered the shares of the company, which means you file a document, they review it, they like it, they say, "Okay, now you can sell to the general public."

There are new rules all around that as well. What happened was, then they said, "Okay, well if you have to register, not every company can register with the SEC. That's ridiculous. Small companies will never have the capital to do that." They said, "Okay. Well, we'll allow another regulation called Reg D," and that regulation allows accredited investors, people who have more than \$200,000 in income a year for the last two, or three years, or a million dollars of value outside of their home and assets, they can be an investor.

They allowed that to class system, where if your neighbor starts Apple, tough, you can't invest. Now Steve Jobs who grew up in a middle-class neighborhood. His parents, his parents friends couldn't invest in Apple, because they were not allowed to. That seems unfair, but that's how it was.

Now the job site comes in and says, "Well, we're going to now allow it." Why? Because now that the SEC has a system called Edgar, which is a database of all the filings – this is modern. This is not the old days, where you send things by mail. That database allows the public, the general public to scrutinize every filing instantly as soon as they're filed. Pretty cool. Well now, they came up with the JOBS Act which says, you can use regulation crowdfunding, raise up to a million dollars from anyone with a limit. The limit is \$2,000 or 5% of your income.

If you took your example of the person Joe, average Joe making 80,000 a year, nice salary, well that's \$4,000 they can invest in regulation crowdfunding a year. Not punitive, but not crazy enough where they lose all their savings. If you think about the JOBS Act, it changed everything. Why? It now allows your neighbor to start a company and for a normal person to participate in that endeavor is an incredible opportunity.

That's the regulation I've been using on StartEngine, called regulation crowdfunding and another one called regulation A+, which allows you to raise up to 50 million, but the filings are a little bit more involved with the SEC. It's like your mini-IPO, but it works. The JOBS Act changed everything and that's why I'm here talking to you today, because my new mission after helping LA become a tech city, which I would say in some ways worked, but now we're helping entrepreneurs achieve their dreams.

[0:39:05.1] JM: When a company raises money through some equity offering, they should be structuring that equity offering in a way that presents the investor with a real opportunity to make money. What are the characteristics of a company that should raise money through a crowdfunding system like StartEngine?

[0:39:29.7] HM: Well, it's very simple. You have to be a corporation. Typically, most corporations are Delaware C Corps, but you can also be a limited liability corporation, also called LLC in your own state. That's the first foundation. Secondly, the founders around the ones who are the first shareholders of the company, they have to be – they can't be bad actors. What is a bad actor? Is someone who in their past have had some financial fraud and financial dealings that were basically viewed as negative and that stays for 10 years.

We review all of that. We review that illegal backgrounds of the company, the backgrounds of the founders, and then we look at the presentation, the financials. Is it properly presented? Is everything disclosed the way it should be? What we don't do is we're not editors. We don't decide this is a great idea, or a bad idea. I mean, look, going back to the world of games, Tetris, very big game, everybody passed on it, except for one little shop. I mean, think about that one.

Twitter, big company, big opportunity, so many companies passed on it. The reason, this happens is that when you create the future, it's very hard for investors to understand what you're doing. Some do and these are the early investors in Uber and Twitter and Airbnb, who frankly were big risk takers. They saw something, or maybe they didn't see something, they just took it. They were lucky.

In a way, what constitutes a good investment or bad investment is hard to say, because it's hard to predict. Typically 90% of startups fail, but the ones who succeed are humongous. In a way,

we are not editors, we are builders. We help put in place the right foundation for the companies, we look at everything they're presenting, and once it's on the page, at that point, it is the crowd and the wisdom of the crowd that decides if this is a good investment or not.

[0:41:42.0] JM: Okay. You see your role as a minimal gate to make sure that this is not a fraudulent institution. To make sure that the businesses are actually trying to accomplish what they are stating, as opposed to presenting the business as a front for some fraudulent institution.

[0:42:03.6] HM: That is correct and that's where if we want to go into the ICO conversation, you will see right away why what we're doing makes a lot of sense, because we are in a way making sure that the investment opportunity is as stated, or when you go into the ICO world and was reported in the Wall Street Journal, hundreds of ICOs were complete utter fraud. There was no gatekeeper, there was no anybody saying that the photos on the page were false, the white paper was made up. Maybe if you're savvy and your friends was Vitalik Buterin and you can call them up and say, "Hey, what do you think about that?" He will tell you in two seconds this is bullshit.

Unfortunately, most people who don't have that access. A lot of people got duped by ICOs. Unfortunately, there was no regulation like regulation crowd funding, regulation A+ to prevent the general consumer from being defrauded.

[0:43:09.7] JM: How do you define an ICO?

[0:43:12.2] HM: Basically, an ICO is an initial coin offering. Basically, it's a company, or a project, some people call them projects, or a foundation, or some entity that is going to issue a coin, a token that is typically an ERC 20 smart contract on the Ethereum blockchain, and it will limit the number of tokens and they will say, "Okay, here's the distribution of the tokens. If you give me a Bitcoin, you'll get so many tokens. By the way, because there's a limit, you should be assured, like if you were a shareholder in a company, that your tokens have value because they're not unlimited. Then we're going to be put putting these tokens on these following exchanges so that you have some liquidity opportunity." That's what I see an ICO is.

[0:44:09.2] JM: How do you define the difference between an ICO, like what is the meaningful difference from the point of view of an investor, from the point of view of a company between an ICO and other funding vehicles?

[0:44:22.6] HM: Well, an ICO as it was portrayed last here was a funding mechanism for a company, or a project, or an entity that issued in-service issuing shares in a corporation, issued tokens on a blockchain application. Most of the time, I would say most of the time, probably over 90% of the time, the application did not exist. What they were issuing is a token that was made up and using an ERC 20 called a smart contract without any company behind it. I mean, without anything functional behind it. It had no utility. Even though they called it the utility token and they said, "Oh, yeah. It has this functionality. Look at my white paper." It turns out that was not the case.

I would say, most people who invested and they were investors, didn't realize that what they're investing in is just a token that has no anything else functionality than sending it from one person to another. I mean, anybody can write an ERC 20 contract, smart contract in half an hour to an hour. Theoretically, you can create an ICO in an hour by just creating a smart contract and then making a white paper and a webpage and then go around the world pitching it and raising money.

In my view is, and that's where we came in as I said, "Okay, look. This SEC, the Security Exchange Commission issued a report on the DAO, which was a failed – I mean, I guess a failed ICO." Said, this is a security sale. You're selling securities. Even though it's a token, it doesn't represent shares of a company, it's not that – it's a security, it's an investment contract. That was a seminal bullet and that was published by the SEC last, I think was last July. This changed everything, because the companies who were issuing ICOs realized quickly and their lawyers did too that they're going to have to use regulation.

That changed in February of this year, because the SEC start doing enforcement actions in a very large scale. They did a little bit last year and end of the year, but in a large scale. You're talking about anybody who's an American company, or a company that has American investors, is receiving a letter from the SEC. That is significant. The cost to dealing with that is high in the hundreds of thousands, if not millions of dollars and it can kill a business.

The enforcement is not even concluded, because the SEC is probably going to be issuing enforcement actions all over the place in the next two years. It takes a long time for a regulator, like the SEC to do it. It's a commission. They have rules themselves. You have to imagine that it's very disruptive. Where StartEngine came in in this story of the ICO as we said, hold on a second, we're experts in a JOBS Act. We can help. We believe that we should bring the ICOs out of the shadows into regulation and using the JOBS Act, and that's very much what we did in the last 12 months.

[SPONSOR MESSAGE]

[0:47:45.8] JM: You listen to this podcast to raise your skills. You're getting exposure to new technologies and becoming a better engineer because of it. Your job should reward you for being a constant learner and Hired helps you find your dream job. Hired makes finding a new job easy. On Hired, companies request interviews from software engineers with upfront offers of salary and equity, so that you don't waste your time with a company that is not going to value your time.

Hired makes finding a job efficient and they work with more than 6,000 companies from startups to large public companies. Go to hired.com/sedaily and get \$600 free if you find a job through Hired. Normally, you get \$300 for finding a job through Hired, but if you use our link hired.com/sedaily, you get \$600, plus you're supporting SE Daily. To get that \$600 signing bonus upon finding a job, go to hired.com/sedaily.

Hired saves you time and it helps you find the job of your dreams. It's completely free. Also, if you're not looking for a job but you know someone who is, you can refer them to Hired and get a \$1,337 bonus. You can go to hired.com/sedaily and click refer a friend.

Thanks to Hired for sponsoring Software Engineering Daily.

[INTERVIEW CONTINUED]

[0:49:28.6] JM: First, there's this question of security tokens versus utility tokens. The utility token idea was pioneered by Bitcoin and the idea is you can pay money in Bitcoin to transact with other people across the blockchain. You have to pay some transaction cost and you have to pay that in Bitcoin, and then the miners are rewarded for verifying transactions by getting paid in Bitcoin. This is the first utility token.

Then you have Ethereum, which has a similar mechanism, but for verifying and purchasing compute on a blockchain. You have Filecoin, which is a system for verifying and transacting in storage space across a blockchain. Then you had a lot of companies that sorta try to present something that looked like a utility token, but was not really a utility token. I talked to some companies that were like this, where they presented something, and this is where the having played games comes in, where you can look at a currency that is being offered by one of these companies and they can say, "Hey, this currency is really useful for buying this thing." You can look at the economic system, you can especially contrast with other economic systems and say, "Actually, there's no reason why anybody would use that for the way that you are marketing it, but nice try."

You can do that with a lot of these utility tokens, where if you really squint in the way that they're telling you to squint, it looks like a utility token, but in actuality it really does nothing. If it doesn't perform the utility, it actually has no value whatsoever. You are not purchasing a percentage of the equity that this overall ecosystem is going to create for the creators of the project. You're not purchasing anything with any monetary value. It's just a BS "utility token" that actually has no utility.

Now on the other hand, you have the idea of security tokens, which would be a percentage of profits that could be reaped by the system, which is much more the idea of a share in a company. I think you would probably agree with me. Is that an accurate representation of security tokens versus utility tokens?

[0:51:45.8] HM: Yeah, I think it is. Let me give you my perspective on it. Again, I'm a game guy who has become a finance person. I took all my financial exams and our companies become licensed, trying to become a broker dealer as well and build our own trading platform, all regulation, right? Front door, everything by the book.

Security is interesting. There are lots of tests. One of them is called the how-we test. These tests are trying to determine if what you're selling is a security versus a commodity. The easiest test for your audience to understand is this, are you investing in it because you want to use it, or are you investing because you want to make money?

If you're investing to make money, most likely it is a security, and that's fine. The problem, reason why these companies don't want it to be a security is because all of the so-called exchanges, like crackend, Bitrix, BitFenix, shapeshifter, they won't take a security. Well, they've taken securities now, we know that because those tokens turned out to be securities, but at the time, it was not defining security.

If you define your token as a security token, they won't take it because they know clearly that they're not a broker dealer. They can't do it. The idea of a security token is a token that is purchased for an investment. You have to have several things, several features in it, the utility, right? The feature of a security token is one governance. What is governance? Let me give you a sense of what governance is.

If you sell shares in a company, in order to issue more shares, create more shares in the company, you need a vote of the shareholders. How is that? That's governance. You'll see that in some proposed blockchains, like EOs, we're just voting – if you have tokens, you vote for them, that's proof of stake. This is called governance. It turns out corporations have tons of governance; you have a board of directors, they have to sign documents. When you want to issue more shares, you need a vote from shareholders, then the board of directors have to sign it. There's a process for doing things that protects the investor.

A security token has governance around it. Then the other thing is how do you make money? Well, if it's shares as you mentioned, if the company is sold, you get a piece of the sale. If there's dividends paid, you get your pro rata share of your dividend. If it's a revenue share investment, if they have revenue, you get your piece of revenue. That is how as an investor you make money, because your investment, the whole sole purpose is to get a return on your investment and to make money.

A security token, the main definition is you need two components; you need a way to make money, so that's a rev share, debt interest payment, or equity payment, or a combination of that. Number two is governance around it, so you're protected.

[0:54:45.6] JM: How does an ICO, if we're talking about security tokens, if I'm starting a company, how does the issuance of an ICO of a coin, how does that differ from me simply issuing equity to my early shareholders?

[0:55:06.7] HM: I would say to answer this question is it's probably similar in a way. We took a stance and I wrote some papers about it that the idea that you have a token, that is a security token, that is neither debt, neither equity, neither revenue share, it has none of that, and some people call it an investment contract, we believe us as a company that it's not necessarily the right approach today, because there's no governance for the investors, and there's no guarantee that this token that you're issued is a security token can be usable on a blockchain application, because if it's a security token and remains a security token, when you go on to use it on a blockchain application, you have to do all sorts of tests.

Imagine going to a Starbucks with shares in a stock certificate in Starbucks and you say, "I want a cappuccino." When you're about to pay, you give them the stock certificate. They look at it and they have to call the broker to make sure you have it, it's still legal, they have to ask to sell the shares trade them that takes two days to clear the trade and then they give you change back in dollars. It doesn't seem practical at all.

Our view was to introduce this idea of two tokens. We call it the real agreement for tokens and equity. The first token you're issuing is shares in your company, for example and the company owns a lot of those tokens, so that's treasury. The benefit of those tokens that the company owns are for the shareholders. They don't have to distribute the tokens to the engineers, to the advisers, they just keep them in the company, that's your reserve and that's a value, and then the rest of them are sold to investors.

You have investors who have tokens, you have the company who has tokens, so if the company, if those tokens get valuable, big, great; the company has the value, the shareholders get a piece of the value. Totally fine. Or, you can distribute those tokens to the individual

shareholders when it's available. I call it the two tokens strategy. First token, it's either equity, debt, revenue share. The second token is an if and when. If the blockchain application is launched and when it is launched, then we will distribute the tokens to you guys, to the shareholders, or keep them in treasury for your benefit.

That two-token strategy gives a model that allows money to be raised legally using a regulation crowdfunding, regulation A, regulation D5 or 6C, online was a website, totally legal within rules of the SEC. Then the second token is a true utility. Why? Because the blockchain application does exist at the time it's issued and it's usable. It has a utility. That's a perspective that we've brought to the marketplace that seems to be resonating right now.

[0:58:11.2] JM: With these tokens, one advantage that people often cite is that they are more liquid. It's easier to transfer ownership of shares in a company, or utility coins between each other, then it might be if these were equity in a private company. Does that factor in to the model that you're presenting for the ICO, the small business, or the upstart ICO?

[0:58:43.3] HM: Actually not really, because it is true that trading a utility token is the easiest thing in the world. There's no country boundary. You can be in the US, you can be in Japan, there is no regulation in a sense that you have to make sure that you still own the token, because if you have the private key, you're the owner. That's why utility tokens are extraordinarily simple and inexpensive to trade, and it works. Look at all the big exchanges like Binance. They're billions of dollars of tokens easily.

Now, let's talk about security tokens. Now if you have a security token, it's the same thing as trading shares on a stock exchange. This is complicated. It's heavy, it's expensive, it's time-consuming. You need a broker dealer, right? Now we're going to be building a system where we're going to try and make the security token nearly similar. Not exact to a utility token, in the sense that you can instantly trade on our platform.

However, in order to achieve that, there's some problems we need to solve that we figure out already how to do it. One of them is you need a unified new smart contract that actually takes into account the fact that the security tokens are owned by people, not by wallets. Let me explain, when you buy shares in Starbucks, it turns out that there's a transfer agent that you

don't know exists that is recording your ownership and those shares. Should you lose your start certificate, should something happen, you can go to the transfer agent with a sworn affidavit and get your shares back. Done. There's no loss.

Or for example, if a spouse steals your shares, cashes them out and then walks away, well there's a legal process to recovering your property. That's called the court systems. No big deal. Everybody is aware of it. It exists and it's to the service of the population. Now the way we solved it is we're now going to be issuing tokens as security tokens with the transfer agent using a new smart contract called LDGR. A LDGR, think of this as like an ERC 20 that is compliant with security laws.

Basically, it's simple. The company says I want to do an ICO, they go to a LDGR, they get a stock symbol and then they issue the tokens as security tokens to an umpteen number of people. Then when the people want to trade, they can't just go on BitFenix, or crackend. They can't, because first of all, they won't be – they won't take them, but then they need to go on these new alternative trading systems that are coming out soon that are broker dealers, who are able to do those trades. With this smart contract LDGR, they can't do the trade because there is a transfer agent in as a trusted source for the smart contract.

The Ethereum model of creating smart contracts includes a trusted source, the Oracle. The trusted source is a transfer agent, which it turns out StartEngine owns one, called Secure, StartEngine Secure and that's a transfer agent. If someone hacks your smart contract, you will not lose your shares in a company, your tokens. Your tokens are not losable by hacking. The only way you lose them is if you sell them.

If you lose them, let's say you've lost your private key, or someone hacked your private key, then what happens is it gets reissued. The reason it works is because there is a trusted source in the smart contract that will block any transfer. When I re-issue tokens to someone who lost them, original contract is locked forever because the transfer agent is aware that this was a loss and will not permit any transfers.

[1:02:55.1] JM: Okay. Just to wrap-up, what – if I am a prospective investor and I want to, and I'm considering investing – let's say some company goes on to StartEngine, they're raising

money for their, I don't know, some kind of business, name your blockchain-based business, where they want to issue a token, or it doesn't have to be blockchain-based business, just some business where they want to issue a token. They issue a token through StartEngine, what guarantees does the investor have when purchasing those tokens?

[1:03:31.9] HM: If those tokens were issued using a regulation such as regulation crowdfunding, regulation A+, or Regulation D5 or 6C, here's what they get; they get either an investment contract that is signed with the company, they get a share of the company with the understanding of how many shares are outstanding, because that was disclosed, they get financials in the company, they get the knowledge that the company who was hosting the offering when and reviewed and did due diligence to the best of their abilities to make sure that the offering is in compliance with security laws, they get all of that. Plus, they get governance around the issuance, which is regulated by each state and the federal government as well. They get a lot.

[1:04:24.3] JM: Interesting. Okay, where do you hope to take StartEngine in the next five or 10 years?

[1:04:29.8] HM: Well, I'm trying to build a modern financial firm. What I mean by that, is a firm that allows hundreds of thousands of companies to raise capital and have access to capital without any barriers in front of them, removing all the bias and the barriers to access, help entrepreneurs achieve their dreams, build those new jobs that are important for our country, create entrepreneurship and allow investors to have access to early stage investment opportunities, and get liquidity very quickly, whether it's within a year or instantly when they make their purchase.

I'm creating a replication of what you see out there and where the big boys, like the Goldman Sachs, the Nasdaq, the New York Stock Exchange, that whole infrastructure is great for large companies, but it's a disaster for the small little businesses in America. I want to bring that those tools, those extraordinary abilities, to the small little businesses in our country in a cost-effective manner, so that anyone can raise money and any investor decides to have access to these investments can trade and get liquidity for the investments, whether they make or lose money that is really not up to us, but at least the opportunity is there for them to take advantage of.

[1:05:51.3] **JM:** Howard Marks, thank you for coming on Software Engineering Daily. It's been great talking to you.

[1:05:54.3] **HM:** Thank you. Nice talking to you, Jeff.

[END OF INTERVIEW]

[1:05:59.4] **JM:** Nobody becomes a developer to solve bugs. We like to develop software, because we like to be creative. We like to build new things, but debugging is an unavoidable part of most developers' lives. You might as well do it as best as you can. You might as well debug as efficiently as you can. Now you can drastically cut the time that it takes you to debug.

Rookout rapid production debugging allows developers to track down issues in production without any additional coding. Any redeployment, you don't have to restart your app. Classic debuggers can be difficult to set up. With a debugger, you often aren't testing the code in a production environment, you're testing it on your own machine, or in a staging server.

Rookout lets you debug issues as they are occurring in production. Rookout is modern debugging. You can insert Rookout non-breaking breakpoints to immediately collect any piece of data from your live code and pipeline it anywhere, even if you never thought about it before or you didn't create instrumentation to collect it. You can insert these non-breaking breakpoints on-the-fly.

Go to rookout.com/sedaily to start a free trial and see how Rookout works. See how much debugging time you can save with this futuristic debugging tool. Rookout integrates with modern tools like Slack, Datadog, Sentry and New Relic. Try the debugger of the future. Try Rookout at rookout.com/sedaily. That's R-O-O-K-O-U-T.com/sedaily.

Thanks to Rookout for being a new sponsor of Software Engineering Daily.

[END]