Podcast Title: *ACM Bytecast*
Episode Title: Bryan Cantrill

Welcome to the *ACM Bytecast* podcast, where researchers, practitioners, and innovators share about their experiences, lessons, and future visions in the field of computing research. In this episode, host Rashmi Mohan welcomes Bryan Cantrill, Computer Scientist and Software Engineer with a rich career in system software and building solutions to key problems in computing. His path breaking work on DTrace won him multiple awards and a spot in the MIT TR35 list. Today, he is cofounder of Oxide Computer Company.

The conversation begins with a look into Bryan's background in computing. Bryan refers to himself as a child of the 80's, when personal computing was a huge revolution for the time. Until he arrived at university, he didn't know computing was even an academic possibility. When he took his first computer science course, it felt like a lightbulb had gone off. His first position in the industry was an internship with QNX during college. Bryan stresses the importance of following your own path regardless of what is popular and being intrinsically motivated by the problems you're solving. In 1996, there were people telling him that OS was officially dead. Bryan was part of the company that proved this untrue. Don't make decisions based on fear of what could happen. Rather, make decisions based on what you find motivating and worthwhile.

Then, they discuss DTrace. Bryan had the initial idea of running kernel text in a way that wasn't stopping a program as an undergraduate. He was confused as to why such a tool didn't exist yet. His first day working at Sun, he was given the empowerment to make his idea come to life. Eventually, there came a point when Bryan knew they had to stop just talking about DTrace as an idea and actually go do it. It was about 3-4 months in before they had proof points of heading in the right direction. In this developmental stage, Bryan recalls being of the mindset that he himself was the customer he was designing for. He believes we always do our best work when we are our own customers.

Because DTrace was such a new thing, Bryan saw the need for a new way to talk about it. DTrace was considered precise answers to arbitrary questions. To Bryan, software observability is the property to actually observe what the software is doing. Without modifying software, you don't actually know what it's doing. Through observability, there is an opportunity to understand it. The term has become much more widely used as people are becoming more interested in the functionality of these softwares. Asking questions about the system poses its own challenges. The need for observability is even more prevalent now with more advanced softwares. Many companies are beginning to see this opportunity. While nobody wants outages, many people aren't so willing to invest much up front in order to prevent such occurrences. When a system is being developed, it's challenging to get them to think about its debuggability.

Then, they discuss crucial questions for systems and how to measure the answers. Rather than dichotomize what we need, there needs to be an understanding of what a system is doing when it misbehaves. Metrics allow us to ask questions, but if you can't answer the question there is a risk of performative metrics. Sometimes, there are so many metrics in the system that it can be hard to see what's happening. Simplicity is key.

Bryan believes it a great disservice to think about machine learning AI as intelligence rather than the sophisticated pattern recognition it is. It's not that ML can't be integrated into this space, but it hasn't happened yet.

As a software engineer, Bryan recalls early awaiting the source licensing with operating systems. He never forgot that feeling and found source code availability as essential. DTrace was the very first OS they open sourced, which took much complexity and organizational willpower. The good proprietary software out there is more of an exception rather than a rule. With the rise of cloud computing, they observed a new angle in which open source projects were being run as proprietary services. Bryan feels that is dangerous, corrosive and hopes it is only a blip in history. Discussing remote-friendly workplaces, Bryan touches on challenges of monetization.

Finally, they discuss Bryan's mentorship and influence with Jeff Bonwick, who really allowed Bryan to flourish and take on hard problems. We lead by example and it is important to be inclusive of new thinking and those outside of the domain. Knowledge is something to be shared, not hoarded. He has tried to pass this mindset on to others throughout his own career. In closing, Bryan gives insight into what he is most excited about in the next 5 years within the computing field.

0:27 Introduces guest Bryan Cantrill
1:00 - Bryan's background
9:24 - Following your own path
12:21 - The initial idea of DTrace and how it came to life
17:15 - You are your own customer
21:30 - Discussing the idea of observability
25:08 - How advanced softwares reflects the need for observability
27:42 - Crucial questions for new systems
31:28 - Application of machine learning
33:15 - Discussing open source
41:40 - Remote-friendly workplaces
42:58 - Mentorship and influence
46:27 - What is Bryan most excited about in the field of computing in the next 5 years?

Links:
Learn more about DTrace
Connect with Bryan on LinkedIn and Twitter
Learn more about ACM and *ACM Bytecast*
Follow ACM on Facebook, Twitter, Instagram, and LinkedIn