

Diving into Life



Kristaps Dzonsons graduated from the University of Rochester in 2004 with degrees in computer science and English. In the years since, he has started two companies and done research in distributed applications, high-speed operating environments, and mathematical and computational game theory at institutions in Latvia and Sweden. He is active in the open-source community, developing BSD tools, giving talks at conferences, and gaining an online following. He has spent the years since graduation in about half a dozen countries, and he actively speaks English, French, Latvian, ASL, and basic Thai. Dzonsons currently lives on the island of Malta but spends time in Thailand, enjoying free diving and scuba. After running his own consulting business for several years, he's about to begin working with a start-up.

While you were still completing your undergraduate degree, you and fellow CSC alumnus MengZhu Wang '03 had a business already in the works. I remember a discussion about an attic full of servers in your off-campus house to handle the workload. What happened to that?

Our company, Gradient, did what most companies do—failed. I still think the idea was a novel one: a self-organizing distribution network for security events on large, volatile networks. We still have a patent on it and haven't given up on other applications of the idea, should the time present itself. But in Gradient's day, just following the dot-com bust, the upstate New York market for high-tech firms was limited. If you're a student with an idea, just do it. Pull together a team of the brightest. It's an incredibly worthwhile investment of time no matter the sleepless nights.

Tell us about the tools you have created for BSD. Which tool that you have developed for BSD is the most widely used? Why do you think it has become so popular?

Curiously, my most widely used piece of open source software is mandoc, a formatter for manpages replacing the traditional groff, apropos, man, etc. I suppose its popularity is license (BSD vs. GPL), language (C vs C++), and speed. I stepped down as the maintainer several years ago, however. All in, I actively maintain about a dozen software utilities and libraries, the most popular being kcgi, a library for CGI and FastCGI; and acme-client, a client for ACME certificate servers. I exclusively write for UNIX systems, mainly OpenBSD, and mostly in C.

Most of my open source software is available from GitHub. I think it's worth imploring computer science students to participate in the open source community. Not only is it beneficial in terms of knowledge, but it's also about being part of a community. I've met a lot of incredibly talented and interesting folks at conferences and hackathons, not even to mention casual interactions on mailing lists and IRC. Collaborating with OpenBSD folks in particular has given me a significant incentive to write clean, well-documented systems. I always assume what I'm writing is going to be looked at by somebody smarter than me, and somebody who's just had a bad day as they sit to read my code.

After you worked for a while, you went back to school to complete a PhD in theoretical computer science at KTH Royal Institute of Technology in Sweden. Where did that lead you?

I did about three years of my PhD—I started in 2008, if I recall correctly. I'd been working on my PhD when I took a leave of absence to work as a researcher for Professor Jorgen Weibull at the Stockholm School of Economics. After realizing how much I enjoyed applying systems, versus studying them, I switched to full-time research work as my own company. I think I knew the moment I started working with Professor Weibull that I'd had enough with theory—the work was so much fun and so rewarding! I think this period stretched my knowledge to the utmost, as Professor Weibull's work involved tremendous simulations and systems related to game theory and mathematical biology. There wasn't a single day that went by without needing to learn something about linear algebra, compiler optimization, or floating-point rounding. This research was the beginning of k-Consulting, my business.

I found my clients (or they found me) through word of mouth. It's not enough to write good code or good documentation. You need to be able to network, and I'm still learning to do that. Sadly, k-Consulting is currently being shut down, as I've since changed work to a company that's still in formation.



What impact has your Rochester education had on your success? What are the classes you took while at UR that have been most useful and why?

A tremendous impact, for sure! The CS and math education at UR laid the foundation for everything I later went on to learn, research, and apply—from algorithms to operating systems to abstract algebra. I've tried to follow this with an "especially," but can't do so. To name one example, I don't work directly with AI, but knowing the foundations has helped me in writing maximization functions for "lumpy" topologies.

I credit Professor Michael Scott and Professor Sandhya Dwarkadas for having the most influence on me and my work. For the former, in having a thorough understanding of operating systems, compilers, and languages. For the latter, distributed and parallel systems. Both for demanding concise, clear documentation and implementations. I use the foundations from those classes on a daily basis.

I also strongly suggest the algorithms courses—every algorithm you learn, no matter how obtuse sounding, will come up again. Yes, even AVL trees. There's a lot of focus these days on new artificial intelligence and statistics. But when it comes to day-to-day work, I've found myself constantly revisiting the classics. This is doubly necessary if you find yourself working in applied mathematics or numerical analysis.

You double majored in English/poetry along with computer science at Rochester. How have you incorporated your love of literature and poetry into your busy life?

If I'd need to pick a major I enjoyed most, I'd probably tend toward English. You asked which CS courses I felt were instrumental, but

if I think of the tools I need in other walks of life, being able to write a decent email can be far more beneficial than a good A* implementation. And, as it follows in nonprofessional matters, having something interesting to say about Edward Albee will generally resonate more than the newest Intel mishap.

Many of our recent graduates dream of working remotely while traveling the world. What advice would you give them? Are there places in the world that are more conducive to having a wanderlust tech career?

My advice is first to get an accountant and a lawyer who can handle the interaction between localities, in particular for those with US citizenship. You do *not* want to overstay your entry permits, especially if you find reasons (romantic or professional) to stay longer where you are.

I'd also advise them to prepare for compromise and slow growth. Not being in constant touch with your employer (or contractors) makes it hard to integrate—which, for a time, may be what you want but also makes you easy to dispose. In time, being able to move around needs to balance against having the money to move around and having a path to move forward.

Working as a freelancer is brutally difficult but is also incredibly rewarding in being self-reliant and knowing all stages of how income gets from your client's pocket to your tabletop. And as mentioned elsewhere, open source will serve you very well in having a well-stocked toolbox! On the flip side, working as a freelancer is also important in discovering and celebrating what nonfreelancers have but don't necessarily appreciate: stability and reasonable income. Life conditions can change!

(Continued on page 26)

Continued from page 11

What do you remember fondly about URCS?

The lounge was a great place to eat lunch or take a break, with the inspiration of bottles from previous PhD students on the wall. I also remember the research group I was in with great fondness: George Ferguson, Mary Swift, Donna Byron, Lucian Galescu, Myrosia Dzikovska, etc. We were a large group (especially for such a small department), and I think we could not have made the dialog systems contributions we did if we had been just working independently.

What do you do in your spare time?

My husband and I both work in New York City, but we live in suburban New Jersey with our two cats, Rock and Roll. In my spare time I am a volunteer EMT in my town (it's great!—the opposite of research—you pick a patient up, transport them, and drop them off and then it's completely done! No "future work"!) and a bird watcher.

Continued from page 13

Tell us about your very exciting life outside of work. How many languages do you speak?

I more or less actively speak English, French, Latvian, ASL, and basic Thai. And no longer speak a few more. These days I'm trying to improve my Thai (and Maltese), but English is far and away the language I speak most. I have both Latvian and US passports but don't have much of an "originally from," having moved around a lot. If pressed, I just say I'm Latvian, as the lion's share of my life has been in Europe.

When not hacking on open source projects or working (though the two intersect fairly often), I free and scuba dive wherever and whenever I can. I prefer free diving, but I am certified as a technical scuba diver as well. Diving adventures have taken me from fighter plane wrecks at 50 meters to free diving with whales.

I currently live on the island of Malta with my wonderful partner and our dog, though I spend time in Thailand as well. When

Continued from page 23

How the company reacts to each of these challenges will determine its later success.

Skills learned while studying at Rochester have served these alumni well. Ian Christopher says "CS fundamentals are a tool I constantly leverage. More importantly though, I learned how I learn, which is essential given the number and variety of issues that will pop up at a start-up. Be it a new web framework or a new systems paradigm in our infrastructure, I know I can pick it up quickly and largely independently." Steve Gattuso '16 mentions the variety of cluster coursework that complemented his computer science classes as well as the opportunity to study abroad, which taught him how to build a life in a new country. Whether it was a particular course that built a skill set, an emphasis on logical problem solving, or building a network of future partners, the University of Rochester experience has informed their entrepreneurial efforts.

Knowing what you know now, what advice would you give to current CS students at Rochester?

Graduate school can seem all consuming—work is life, life is work. But the life of the mind can be balanced. I encourage you to make a list of what matters to you and then proactively seek opportunities in those areas and say "no" to everything else. I also encourage you to budget and track your time and make sure that the "life" buckets aren't squeezed out by the "research" and "service" buckets.

URCS is a small and collegial program. It's important to build your professional network through internships, conferences, and volunteering, but it's also important to enjoy the community of URCS.

in Malta, I speak mostly English—Maltese is a tricky language, although I love the sound of it. When in Thailand, I try to speak Thai, but that's a work in progress. Thais are incredibly helpful when it comes to learning the language.

One language that's been of tremendous utility to me has been sign language. (When underwater it's integral, though surprisingly it doesn't have a large following in the diving community.) There's a significant deaf community in Rochester—and a very friendly one, too! I highly recommend learning a bit and attending a deaf theater performance.

As always, I recommend spending time in whichever sea is closest to you. Because you don't know what you're missing till you've come up close to a five-meter-wide manta ray or bumped heads with a whale or learned to surf or just spent time in a rowboat with a companion of choice. Lastly, no matter where you are, find somebody to spend the time with you.

Our entrepreneurs advise anyone interested in starting a business to understand the labor of love this takes to survive. Muderick offers the following advice: "If you are the kind of person that likes to be really busy, then this is the job for you. Every day you will work a checklist of must-do tasks. At the end of every day, that list will be longer than it was in the morning. The next day you need to effectively choose which tasks you will complete. In 10 years you might just start to see light at the end of the tunnel. If that sounds like a fun challenge, you are cut out to be an entrepreneur. If all that pressure combined with working nights and weekends sounds like a panic attack, I would suggest against it." Other alumni advise to read up on the topic and benefit from others' mistakes, try an internship at a start-up to see the risk and rewards of a small business, and understand that you'll be getting very little sleep—much like your college years. But if it is your dream, there is no better time to try this than when you are young!

CONGRAULATIONS TO THE CLASS OF 2018!



Alejandro Abreu
Benjamin Ackerman
Kennedy Agusi
Sidhant Ahluwalia
Thomas Anderson
Samantha Bahr
Daniel Barnett
Bradley Beyers
Rohun Bhagat
Jeremiah Bill
Nina Bose
Steven Broida
Juan Pablo Castaño
Moreno
Yukun Chen
Noah Christiano
Joshua Churchin
Stephen Cohen
Nathan Conroy
Colin Corliss
Cesar De Hoyos
Gonzalez

Danielle Delooze
Yufei Du
Kyle Edgette
Yidi Fan
Yiwen Fan
James Fazio
Tyler Fulcher
Shengbo Ge
Ari Geller
Andreas Georgiadis
Noah Gorstein
Ali Hamdani
Alexander Hankin
Kurtis Haut
Annie He
Manan Hora
Ravi Jain
Thomas Jeffries
Zhengliang Jiang
Hunter Johnston
Adam Kravitz
Raina Langevin

Xuan Thuong Le
Evan Lea
Gayeon Lee
Yeonwoo Lee
Matt Levin
Sean Levin
Vivian Li
Shuyang Liu
Yumeng Liu
Zixiang Liu
Santiago Loane
Lilian Ludford
Gianna Macri
Alex Mai
Shir Maimon
Justin Maldonado
Abdullah Al Mamun
Nicholas Martucci
Marcelina Martynek
Aaron McClure
Graeme McGuire
Akihiro Minami

Jason Natale
Mario Natalino
Amos Newswanger
Edward Newton
Jacob Niebloom
Luis Nova
Jien Ogawa
Joshua Pachter
Yiming Pan
Michael Parrish
Naropa Perez
Raayan Pillai
Thomas Pinella
Kyle Pollock
Colin Pronovost
Aurek Ransom
Andrew Regelski
Tarin Rickett
Oscar Rojas
Anthony Russell
Jacob Schechner
Alexander Senatore

Luka Sherman
Kim Sitthisack
Grant Sorbo
Daniel Stegink
Zachary Teicher
Xiaoxuan Wang
Maria Weber-Nieves
Conrad Wehner
Joshua Whitehouse
Brandon Willett
Samuel Wlody
Ryan Wong
Charlotte Wright
Jiejiong Wu
Sophie Wu
Linxiuzhi Yang
Yiming Yang
Xinyue Yuan
Zhizhou Zhang
Wen Zhou