A Conversation with Felix Kerger: Taking small steps to mitigate the universal problem of information overload

INNOVATION LEADERS

by Bushra Anjum

Editor's Introduction

In this interview, Ubiquity's senior editor Dr. Bushra Anjum chats with Felix Kerger, an experienced developer advocate working for King.com (Spain). They focus on the increasing challenges of information overload for computing professionals and students. The discussion then moves to some of the strategies that can be employed to make information easier to find and ensure the right audience receives the right information directly.

A Conversation with Felix Kerger: Taking small steps to mitigate the universal problem of information overload

INNOVATION LEADERS
by Bushra Anjum

Felix Kerger is an experienced developer advocate with more than 16 years of experience in programming. He is currently working for King, a leading mobile game developer, leading the developer relations team for King's internal game engine. Prior, Felix worked for Unity as an Enterprise Support engineer in Europe, and before that, he was an indie game developer. He also published a textbook about an open-source render engine and talked at different conferences about software development and, more specifically, game development. Felix is a certified support manager, Scrum Master, product owner, and a member of the ACM's Future of Computing Academy. He holds a master's degree from the Open University.

What is your big concern about the future of computing to which you are dedicating yourself?

The biggest challenge for the future of computing is information overload. The amount of information required to be processed by humans is continuously growing and harder to manage. Two problems arise with this growing mountain of information:

1. How to find, and store, the required information when it is needed.
2. How to ensure that the found information is accurate and helpful for the intended audience.

As computing practitioners, we mostly deal with problem no. 1, the need to find, absorb, comprehend, and sort vast amounts of information every day.

Let's take a few examples. As a developer, the choice of frameworks to develop is staggering. Similarly, the options available for python modules, C++ libraries, software engineering practices, etc. are equally daunting. Even when working on an existing project, the history of design choices, architecture, and design constraints must be understood alongside the used language, libraries, and frameworks before any new contribution is made. Resultantly, it is expected that it takes a newly hired software developer three to nine months to become a productive member of a development team.

As a student, you have to decide if you want to work in the industry or become a researcher. There are so many different industry fields and so many research options that it is almost impossible to make an informed choice.

Those two issues combined lead to the field of computing becoming hard to enter and even harder to stay up to date in. As the demand for computing professionals keeps rising worldwide, the field should not become harder to enter but easier.

Solving this challenge is my biggest concern about the future of computing.

When did you first realize the importance of spreading knowledge in an easy to comprehend and digestible format? What were some of the activities you took part in that solidified this for you?

I first encountered this concern in university. Along with three fellow students, I ran a one-week programming course for new students who did not have any programming knowledge. This prep course was necessary as the official programming course was fast-paced. From previous years, we knew that people without any programming experience were struggling to keep up. This one-week course helped new students to be prepared and introduced me to the pleasures of teaching and presenting complex information in an easy to digest way.

During my bachelor's degree, I had the chance to write and publish a textbook about an open-source engine. That introduced me to a different way of presenting information to an audience. While writing, I also did extensive research highlighting the usefulness of clear documentation and tutorials for any software product.

After graduating, I joined Unity as a Developer Relations Engineer. My job was to help solve our clients' problems, mostly created by a lack of understanding or unclear documentation. A big part of my daily work was quickly digesting information gathered by reading source code, debugging issues, and talking with our developers, and then presenting this information in different formats to our clients.

Today as a Developer Advocate for King, it is again my job to collect and spread knowledge. While it is a different product than before, the problems are the same. Every day, I witness how hard it is for developers to get their work done and stay up to date. Some projects make this more manageable due to excellent documentation and tutorials, and Stack Overflow is a big help as well. However, the majority of projects suffer.

Overall, I saw people struggle in my whole career, not with solving challenging algorithmic problems or complex hardware setups, but with getting the information they need easily.

What are some of the approaches you suggest that can help deal with information overload? What projects and initiatives are you involved with on this front?

There are different approaches to help with information overload.

1. Reduce the amount of new information being generated
2. Make information easier to find
3. Ensure the right audience receives the right information directly

It is safe to say that the first approach is unlikely to happen; therefore, I focus on the other two options, both at work and in my free time.

As a young computing professional myself, I had the challenge of selecting a fitting career path. I am also a member of the ACM's Future of Computing Academy, which allowed me to meet other young computing professionals who shared similar struggles. We agree that the struggle to choose a career path is really a part of the universal struggle related to information overload.

Based on this realization, we started a project to create a roadmap and a database about computing career paths. Our project goal is to help students and graduates get an overview of different career paths, what is required to achieve those, and first-hand experiences from people who traveled along the same path or took different paths to achieve the same position. This work focuses on approach number no. 3; ensuring young computing professionals receive the information they require to plan their careers.

At work, I develop internal tools that make it easier to spread information and, more importantly, keep it up to date. This effort comes under approach number no. 2. I am currently developing a new API documentation tool that enables developers to write their comments directly in source code, verified by unit tests. The tool then creates HTML documents out of it, similar to Doxygen, but with the added benefit of unit tests to ensure that documentation stays up to date.

The above are just two examples of how information overload can be mitigated in computing by either making relevant information easier to find or directly providing the right information to the right audience.

As mentioned in the beginning, information overload is a universal issue, not just in computing. If we do not want to drown in a sea of information, we need to act. Anyone can help reach those goals, e.g., by volunteering in different projects and initiatives and helping to sort and flag information to make it easier to find (e.g., as part of a community project like Wikipedia). Similarly, there are other ways like producing educational content for open source projects, publishing research summaries, or being an active member of Stack Overflow. Everybody can help make information an asset and not a liability to our society.

Biography

Bushra Anjum is a software technical lead at Amazon in San Luis Obispo, CA. She has expertise in Agile Software Development for large scale distributed services with special emphasis on

scalability and fault tolerance. Originally a Fulbright scholar from Pakistan, Dr. Anjum has international teaching and mentoring experience and has served in academia for over five years before joining the industry. In 2016, she has been selected as an inaugural member of the [ACM Future of Computing Academy](#), a new initiative created by ACM to support and foster the next generation of computing professionals. Dr. Anjum is a keen enthusiast of promoting diversity in the STEM fields and is a mentor and a regular speaker for such. She received her Ph.D. in computer science at the North Carolina State University (NCSU) in 2012 for her doctoral thesis on Bandwidth Allocation under End-to-End Percentile Delay Bounds. She can be found on Twitter [@DrBushraAnjum](#).

DOI: 10.1145/3447259