



KENDALL WEIHE



SOFTWARE DEVELOPMENT ENGINEER

(502) - 216 - 5761 | kendallweihe@gmail.com

EMPLOYMENT



Software Developer, KFC YUM! Brands

Oct. 2017 - Present



Junior Software Developer, ElToro.com

May 2017 - Oct. 2017



Research Assistant, Dr. Brent Seales

May 2016 - May 2017



Software Engineer Intern, Appriss

May 2015 - Aug. 2015



Student Assistant, UK Athletics IT

Aug. 2013 - Aug. 2015



Apple Certified Technician, MacAuthority

July 2012 - Aug. 2013

EDUCATION



Computer Science, UK Engineering

Aug. 2013 - May 2017

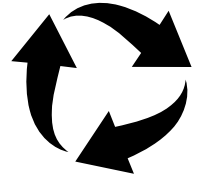
EXTRA CURRICULAR



Mentor, Code Louisville

Jan. 2018 - Apr. 2018

The Tao of Kendall's Software



1. Identification
2. Reaction
3. Solution
4. Codification
5. Automation

Kendall's Software Axioms

- Identify unbiasedly
- React blamelessly
- Solve with wisdom
- Codify value assets
- Automate, automate, automate

Kendall's Software Role

Externally:

- Kendall engineers maintainably, robustly and timely.
- Kendall delivers consistently.
- Kendall has excellent technical articulation skills.

Internally:

- Kendall is driven by financial security, freedom of schedule & creativity, and the ability to continuously learn & grow.
- Kendall plays the wealth game more frequently than the status game.

Kendall is a problem solver at his core. At Kendall's current position, he plays a crucial role in team success. At various points during the day Kendall may be researching solutions for unsolved tasks, designing well-known solutions, playing the role of the scrum master, developing source code and any associated tests, defining infrastructure as code, automating builds and releases, integrating his solution through to a production environment, and identifying & supporting client-impacting issues.

Kendall's Software Skillsets In His Own Words

Computers are like magic made real to me. The amount of complexity to create this resume alone is absolute wizardry. My abilities are rooted in my ambition to leverage the magic computers enable. Originally, I thought that meant learning to program. So I learned to write computer code. Then I learned how I could ship my code to the cloud, and what it meant to execute my code. Then I learned that writing tests (via code) enabled quicker and more resilient code. Then I learned that if I could abstract an architecture outside of any implementation, then the implementation would be easy to write, automate, and deploy. Then I learned that if I could codify and automate an entire product (for example, a cloud platform) then that meant I could build & deploy an insanely valuable product/service in a small amount of time (imagine building a money-making skyscraper in a matter of minutes).

Now I'm at the point of, *wow I have all these technical skillsets to create large scale products and services but how do I integrate them into real world problems that earn wealth?*

Skillsets relevant to a language or framework or toolset or vendor or whatever it may be... are fleeting in nature. I'm capable of picking up new skillsets very quickly, and if I don't use said skillsets on a regular basis, then I forget -- and that is **by intention**.

Kendall's Software Projects

KFC & Grubhub Digital Integration

In early 2018, KFC YUM! Brands announced a \$200 million contract with the food-delivery company Grubhub. Part of the contract demanded a "digital" integration between the Grubhub cloud technologies and the existing, and admittedly technologically out-dated, 4000+ KFC restaurants. At a high level, the integration required Grubhub customer orders to automatically enter the closest KFC restaurant kitchen so that delivery driver's could conveniently deliver hot and fresh fried chicken. This was a gigantic investment by the KFC corporation, estimated to generate approximately \$50 million in revenue the first year, even considering zero marketing efforts.

Kendall was one of two Software Development Engineer's to develop the integration platform -- this was a massive undertaking for such a lean team. Kendall and his surrounding team of one other developer, a consultant, a team of quality assurance analysts, a business analyst and two senior managers, completed the full integration within 6 months.

The technology stack was composed of technologies both in the cloud and the physical restaurants. On one end of the integration was Grubhub (in the cloud) and at the other end was the existing point-of-sale systems developed by NCR (in the restaurants). The cloud technologies were to be hosted in the Azure Cloud and consisted of resources such as: Azure Functions, CosmosDB, API Management, Blob Storage Containers, Service Bus, Event Hub and Stream Analytics. The in-restaurant technologies, constrained by the existing systems, consisted of: .NET Framework applications, HTTP client libraries, the Microsoft Messaging Queue, TCP & SSL, and NuGet packages such as JSON.NET & AutoMapper. Kendall and team were able to automate all of the build & release process through Azure DevOps services such as: Git repositories, trunk-based branching, PR policies, automated unit tests, peer code reviews, automated resource deployments through IaC, and staged environment deployments.

```
{
  "projects": [
    {
      "name": "KFC Grubhub Integration",
      "description": "A means of delivering delicious Kentucky Fried Chicken as seamlessly and conveniently as possible",
      "businessValue": {
        "rating": 10,
        "reason": "Initial projections estimate, a mostly previously untapped, ~$50 million annual revenue increase."
      },
      "personalImpact": {
        "rating": 8,
        "reason": "One of two SDE's"
      },
      "coolness": 6,
      "techStack": [
        "Azure Function Apps",
        "CosmosDB",
        "API Management",
        "Blob Storage Containers",
        "Service Bus",
        "AppInsights",
        ".NET Core",
        ".NET Standard",
        ".NET Framework",
        "Azure DevOps Build & Release Pipelines",
        "Azure DevOps Git Repositories",
        "Powershell",
        "Azure Resource Management Templates",
        "OpenAPI Swagger contracts",
        "OAuth 2.0"
      ]
    }
  ]
}
```

Invisible Ink Identification Of Damaged Herculaneum Scrolls

In 2016, Seales et al. [1] published *From damage to discovery via virtual unwrapping: Reading the scroll from En-Gedi to Science Magazine*. Dr. Brent Seales, Chair of the University of Kentucky Computer Science Department led a research initiative that virtually unwrapped a partially damaged & fragile scroll from En-Gedi, the oldest scroll in Hebrew outside of the Dead Sea scrolls, which unveiled the second oldest known book of Leviticus. The unwrapping was not possible prior to Dr. Seales' use of micro-CT volumetric scanning coupled with an algorithmic unwrapping of a three-dimensional scan. Shortly after the publication, the story was taken up by every major new organization in the United States, even making the front page of the New York Times. Kendall was fortunate enough to be part of Dr. Seales' research team through a University funded grant program.

Kendall joined Dr. Seales' team after the completion of the 2016 virtual unwrappings. At the time when Kendall joined, the team was geared towards applying machine learning to identify seemingly invisible ink written on the scrolls uncovered from the Library at Herculaneum, located in a sister city to the notorious eruption of Mt. Vesuvius at Pompeii in 78 B.C.E.. The Herculaneum scrolls differed from the scrolls uncovered at En-Gedi in that the writing was scripted with a carbon-based ink, as opposed to an ink potent in heavy metals like the scrolls at En-Gedi (heavy metals light-up under micro-CT scans, making the writing easily visible). The team was fortunate enough to have a single fragment of an already-unwrapped Herculaneum scroll wherein the ink was visible to the human-eye -- this could be used as a source of "ground truth" in a machine learning model. Kendall and his team members applied machine learning and computer vision techniques & toolsets as a means of identifying ink on the virtually unwrapped, but seemingly blank, Herculaneum scrolls. The team researched and experimented on micro-CT volumetric datasets ranging from 50GB-100GB in size with: Python, Tensorflow, Convolutional Neural Networks, various deep-neural-network models, unsupervised and supervised learning. In 2019, Seales et al. [2] proved the theory that, although carbon-based ink may seem undetectable to the human eye under micro-CT scans, with the application of machine learning, the carbon-based ink has an undeniable artifact detectable only through well-trained computer models on large datasets.

[1]: Seales, William Brent, et al. "From damage to discovery via virtual unwrapping: Reading the scroll from En-Gedi." *Science advances* 2.9 (2016): e1601247.

[2]: Parker, Clifford Seth, et al. "From invisibility to readability: Recovering the ink of Herculaneum." *PloS one* 14.5 (2019): e0215775.