Thesis Project Portfolio

Microservices: Consolidating Functionality Across Multiple Services
(Technical Report)

Automation's Negative Effects on the Common Workers
(STS Research Paper)

An Undergraduate Thesis

Presented to the Faculty of the School of Engineering and Applied Science
University of Virginia • Charlottesville, Virginia

In Fulfillment of the Requirements for the Degree
Bachelor of Science, School of Engineering

Jarod Johnson
Spring, 2023
Department of Computer Science

Table of Contents

Sociotechnical Synthesis

Microservices: Consolidating Functionality Across Multiple Services

Automation’s Negative Effects on the Common Workers

Prospectus

Sociotechnical Synthesis

Automation technologies have the potential to enhance efficiency across numerous sectors, freeing up time for individuals to focus on more crucial tasks. These technologies can automate operations that pose hazards to humans, thereby preserving lives. However, despite the benefits of using automation technology in our society, the implementation and use of it poses a serious problem. Automation technologies are simply more efficient than humans, making them more cost efficient than employees in the long run. This gives companies incentive to create and adopt these efficient technologies, which, in turn displaces human workers. As the employment opportunities for a specific career diminish, the supply of unemployed workers with those skills increases. This causes the wages for the working class to decline, and also leaves many searching for jobs that may not exist in the near future. In my research, I found that this has happened in many industries over the course of the 20th century. It is estimated that it will continue to happen in more job classifications as technology advances. This poses the question: how can we lessen the struggles of the displaced workers while also allowing society to adopt useful technologies? There are some influential figures who have proposed potential solutions to help the affected workers through the tough transition including Bill Gates and Elon Musk. The most significant challenge in this situation is retraining displaced workers to learn valuable skills that will allow them to find employment in other careers. Given the cost of higher education and the age at which many of these workers are displaced, the best course of action is to have retraining programs that can be run federally or privately. However, federal programs would require tax based funding. Due to the profits that are made from replacing human workers, legislation should be passed which would create a separate tax, known as the “robot tax.” Companies that benefit from and profit off of automotive technologies would pay this tax in order to help remedy the

issue at hand. It would fund federal retraining programs, and allow the government to pay a universal basic income of sorts to those that are still training or searching for new jobs.

The following describes my technical project, which was a summer internship at a small financial technology company. This work had no relation to my STS research paper. The microservices platform of Amount Small Business, a subsidiary of Amount Inc., facilitates loan application processing and decision-making for small banks. However, the legacy functionality of the PDF document microservice, PDX, stores and generates reports with extraneous information. To address this issue, the software team decided to replace this feature with logic that extracts PDFs from third-party financial institutions such as Equifax, ThreatMatrix, and Paynet. As part of this initiative, I was responsible for replicating the PDF storage and retrieval capability of the third-party reports in another microservice, UDX. To achieve this, I created new database tables for PDF objects in UDX and migrated the logic from PDX to UDX. I utilized tools such as Orika object mappers, JOOQ database query library, and Reactive Java (JavaRx) for development. The functionality was tested through local HTTP requests using Postman and unit tests using Mockito testing library for Java. The new functionality was successfully migrated with minor adjustments required by the team, including modifying the front-end React.js service to pull PDFs from UDX instead of PDX.