

**Development of Open Source Ansible Automation modules to Control AIX**  
**Low-Code as a means to provide Abstraction**

A Thesis Prospectus  
In STS 4500  
Presented to  
The Faculty of the  
School of Engineering and Applied Science  
University of Virginia  
In Partial Fulfillment of the Requirements for the Degree  
Bachelor of Science in Computer Science

By  
Richard Taylor

October 27, 2022

On my honor as a University student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments.

**ADVISORS**

Kent Wayland, Department of Engineering and Society

Rosanne Vrugtman, Department of Computer Science

Briana Morrison, Department of Computer Science

## **General Research Problem: Low-code as a method of Automation and**

### **Abstraction**

*How does the further abstraction of code through “Low-code” affect software development, and how can it be utilized for more efficient development?*

A business, individual, or organization’s ability to quickly create the most efficient solutions to problems is the key to their success. In the modern world, software provides these solutions- but there are barriers to the development of software. Most of this development is traditionally done either by in house IT departments or is contracted to software-specific companies, making application development both more time consuming and expensive (Coleman, 2022).

Enter Low-Code platforms: where once the development of complex software solutions was limited to the far echelons of the IT industry, a new form of development has arisen. Low-Code development allows for those with far less experience, to no experience at all, develop and integrate applications for their own need, resulting in a new class of “Citizen Developers”, who work in different fields than traditional software developers, and develop their own custom applications and tools (Hurlburt, 2021, p.4). Platforms range in capabilities and usages: From simple website and front-end builders such as Wix, WordPress, and SquareSpace, which are used primarily by individuals and small businesses, to powerful platforms from software companies like Salesforce, Mendix, Microsoft, and more, that can create complete, sophisticated software systems and applications for enterprises (Bendor-Samuel, 2022).

Low-Code platforms like these are predicted to grow, with experts claiming that most larger enterprises will adopt the technology by year-end (Gartner, 2021). These platforms

essentially serve as a method to abstract code from the end user, allowing for substantial change in how development of software occurs. The technical portion of this report shows one such way in which this abstraction is made possible by detailing the development of modules for RedHat's Ansible, a tool that aims to simplify IT through automation. This gives insight into how low-code works, and how it can be used to change the daily tasks of developers to enhance the efficiency of development. The research section of the report aims to address the question of how low-code abstraction provided by tools like Ansible will affect both the new "Citizen Developer" and the professional programmer by analyzing how past methods of abstraction and automation have affected both the process of software development and other "high-skill" fields, and examining the current roles of professional and Citizen Developers and the impact low-code tools may have on these roles.

The Technical Report and Research Paper together will both show the factors that shape the development of low-code technology, through the example of the development of Ansible Automation modules, as well as explore how the technology will shape the current and future ecosystem of software development.

## **Development of Open Source Ansible Automation modules to Control AIX**

*How can we more efficiently secure AIX logical devices against data compromise and theft through the creation of Ansible Automation Tools?*

Ansible Automation is a platform that aims to make development and delivery of applications more efficient by reducing complexity (Red Hat Ansible, 2020). In the summer of 2022, I worked with a team of IBM engineers to develop and maintain an open-source collection of ansible modules that automate controlling various processes on machines using IBM's AIX

operating system. An ansible collection for AIX allows for development teams using AIX machines to update and install tools, manage user accounts and partition details, and perform many other management tasks remotely, on multiple machines, all at once. Utilizing the collection allows for system administrators to manage target machines through OpenSSH connections, reducing the downtime of a system when changes need to be made and speeding up the various management tasks.

My project was to develop ansible modules to serve as wrappers for the AIX `hdcryptmgr` command, which allows for management of encryption settings on logical device partitions. Data breaches pose a significant threat to enterprises, as the loss of sensitive information can result in financial and reputational losses. Globally, it is estimated that the global annual cost of data breaches in 2019 was \$2.1 trillion, with 43% of these leaks coming from internal employees (Cheng et. Al., 2017). The `hdcrypt` modules developed reduce these kinds of leaks by simplifying the process of encrypting devices and managing various encryption settings, so that in the case of an attacker getting access to the storage devices themselves, the data contained on them is useless.

To develop these modules, I worked in an agile team with members from across the world, and engaged in social coding practices, using ZenHub for project management. Starting with broad development “epics” (support for encryption of devices, support for adding authentication methods to devices, etc.), these were broken down into deliverable user stories that could be completed within sprint cycles.

Ansible modules are written in Python, and are run through ansible scripts called “playbooks”. In each playbook, a user defines one or more target machines, a user for those machines, and a series of “plays” which specify the ansible module to run, the collection it

belongs to, and any parameters or options that it is to run with. The ansible engine parses these playbooks and runs the plays on each target simultaneously by creating SSH connections. The most major part of the project was the development of the modules, which parsed through the options set in playbooks and formulated commands to be sent over the SSH connection and ran on the AIX target machines. In addition to this, I applied the principles of test-driven development to create integration test buckets, which are run for every new release of the collection to automatically perform quality assurance on the modules. Finally, a demo playbook was included with the modules to give potential users an example and explanation of the module's usage and meanings of each parameter that could be used. Documentation for the modules was automatically generated based on a documentation string that was written in the source code for the modules.

During my time at IBM, I was successfully completed the development of 2 hdcrypt ansible modules, including rigorous test development, as well select small fixes to the existing collection that were submitted by the open-source community. As of October 2022, the most recent version of the Ansible for AIX collection has over 135,000 downloads, showing that such automation and low-code offerings are valued by developers across the IT field.

## **Low-Code as a means to provide Abstraction**

*How has abstraction changed the role of the Software Developer in the past, and how will the further abstraction provided by Low-Code affect the Software Developer in the future?*

### *Background*

Software Developers have always used abstractions to efficiently create software. High-level programming languages like Python abstract details that lower-level languages like C

require users to include. Languages like C abstract users from the machine code that it is assembled into, which itself is an abstraction of the logic gates that the CPU uses to execute programs. As these abstractions are made, more and more people are able to create software, as the amount of technical knowledge and capabilities needed are lessened. Low-Code platforms are another abstraction that goes one further down the chain: to build applications and software systems, now developers no longer need programming experience at all (O.E. Team, 2021).

This new iteration of abstraction brings non-technical professionals into the realm of being developers of software. A Gartner report found that 41% of employees outside of IT fields now create or customize their own technological solutions (2021). This implies that tasks previously withheld only for IT departments are now being completed by these so called “Citizen Developers”. As the importance of understanding technical detail and possessing programming education or experience lessens, the role of the Software Developer and the IT field will shift as well. It is important to analyze the effects of this change, as it impacts the efficiency of a major market in the global economy, and an estimated 1.6 million developers in the U.S. alone (U.S. Bureau of Labor Statistics).

### *Literature Review*

Traditionally, those viewed as “high-skill” workers (those with higher levels of education and job roles that involve more complex tasks requiring human judgement and soft skills) have been seen as safe from job-automation, but recent advancements in technology, specifically with Artificial Intelligence, Machine Learning, and Big Data, have led to the notion that these workers have tasks that can be automated with software as well (Acemoglu & Restrepo, 2018, pp. 204-205). Some of this automation has already occurred, with notable examples of venture capitalist firms electing a decision-making algorithm to its board of investors, and IBM’s claim that the

supercomputer Watson can predict medical results and outcomes better than human physicians, through learning algorithms (Acemoglu & Restrepo, 2018, p. 205). This automation is parallel to what is seen as “low-skill” automation of the past: where then, more routine and manual jobs were replaced by the assembly line and mechanical machines, now more complex and analytical roles can be replaced by the computer. As the economic impact of that automation was (and still is) widespread, the potential automation of “high-skill” jobs, and software development specifically, will be as well. Further research into the topic aims to understand that economic and social impact, as well as what changes the Software Developer will see in their jobs and training.

Adoption of low-code for the Citizen Developer is a means to make software development more accessible. Allowing non-IT personnel the ability to control and create their own software systems, as opposed to relying on IT teams for tools, allows for more efficient development of solutions that serve immediate purposes better. Low-code, for the non-coder, is “about democratization, about extending the ability to be creative with computers” (O.E. Team, 2021). This democratization brings a shift in who is known as a “Software Developer” and what exactly it is they do. This research paper attempts to understand that change, and how it may affect future developers both in and out of IT.

### *Evidence and Analysis*

To understand the changes (if any) that low-code will have on the field of software development, this report will first attempt to understand how such changes have occurred before, both in IT and other fields, as well as what factors lead to the adoption of low-code platforms, and the factors that lead away from its adoption as well. To achieve this, I will collect evidence regarding past abstractions in programming, such as changes from low-level languages to high-level ones, as well as automation in other industries, and analyze how those changes affected

their respected fields. Furthermore, I will gather current views on low-code and no-code platforms from both current Software Developers and Citizen Developers, as well as analyze how the platforms are being used currently to understand what changes have already been made, and what many are predicting will happen in the future.

### *Conclusion*

From this research, I hope to further the understanding of how low-code will act as a form of abstraction that is to a degree that has yet to be seen in the field of software development. Through analyzing the development of such a low-code platform in the technical report of this thesis, insight into how such platforms can be developed and used, and the purpose for which they are used will be gained, which in turn will help further the understanding of the affect low-code will have. Through the examination in both of these projects, I hope to uncover the ways in which software is made more efficient through abstraction, and what affect this efficiency will have on the field of software development as a whole.

## Works Cited

- Acemoglu, D., & Restrepo, P. (2018). Low-Skill and High-Skill Automation. *Journal of Human Capital*, 12(2), 204 - 232. <https://doi.org/10.1086/697242>
- Bendor-Samuel, P. (2022, June 8). *What Enterprises Need To Know About Low-Code Software Development Platforms*. Forbes. <https://www.forbes.com/sites/peterbendorsamuel/2022/06/08/what-enterprises-need-to-know-about-low-code-software-development-platforms/?sh=24baed607ff9>
- Cheng, L., Liu, F., & Yao, D. (2017). Enterprise data breach: causes, challenges, prevention, and future directions. *WIREs Data Mining and Knowledge Discovery*, 7(5), 1-14. <https://doi.org/10.1002/widm.1211>
- Coleman, S. (2022, April 22). *Low-Code vs. Traditional Development*. Low-Code Connection. <https://lowcode.com/articles/low-code-vs--traditional-development.html>
- ElBatanony, A., & Succi, G. (2021). The Pareto Distribution of Software Features and No-Code. *Proceedings of the 1st ACM SIGPLAN International Workshop on Beyond Code: No Code*. ACM Digital Library. <https://doi.org/10.1145/3486949.3486964>
- Gartner, Inc. (2021, February 16). Gartner Forecasts Worldwide Low-Code Development Technologies Market to Grow 23% in 2021. <https://www.gartner.com/en/newsroom/press-releases/2021-02-15-gartner-forecasts-worldwide-low-code-development-technologies-market-to-grow-23-percent-in-2021>
- Hurlburt, G. (2021). Low-Code, No-Code, What's Under the Hood?. *IT Professional*, 23(6), 4 - 7. <https://doi.org/10.1109/MITP.2021.3123415>
- IT Automation with Ansible*. (2020). Red Hat Ansible. <https://www.ansible.com/overview/it-automation>
- Team, O. E., (2021). *Low-Code and the Democratization of Programming*. O'Reilly Media, Inc.
- U.S. Bureau of Labor Statistics. (2022). *Software developers, Quality Assurance Analysts, and testers: Occupational outlook handbook*. <https://www.bls.gov/ooh/computer-and-information-technology/software-developers.htm>