Software Development using AI Supercharged Code-free Tools: Necessary Skills for Tech-driven Environments

Beatrice Epwene

Article Info

Article History

Received:
9 June 2024
Accepted:
25 October 2024

Keywords

No code
Low code
Citizen developer
Artificial Intelligence
Humanities

Abstract

Researchers in the humanities and social sciences avoid exploration of certain questions for lack of skills in developing software and computational tools needed. At best, they adapt tools developed for other purposes, at worst they recoil from research tasks for lack of tools to pursue those questions. Although technology permeates every facet of life today, with application development and use predicted to increase exponentially, software development is still not a foreground issue in the humanities and social sciences. It is imperative that scholars develop skills in application development to answer unique research questions. Humanities and social sciences need customizable software to tap into idiosyncrasies of research subjects, to explore difficult or time-consuming questions by analyzing text at scale or by deploying computational power to bring new insights to research questions. Scholars need to articulate questions they want to answer, isolate, and codify elements constituting data then enhancing research with software far more than worrying about writing code. No code development platforms help in faster application development with very little programming knowledge. Advancements in graphical user interfaces, API integrations supercharged with Artificial Intelligence capabilities, make easy-to-develop no code software powerful to meet unique research needs in humanities and social science research.

Introduction

Technology permeates every facet of life today and dependence on technology and digital transformation are predicted to increase exponentially, especially after the COVID 19 pandemic and the dawn of remote work (Sahinaslan et al., 2021). Consequently, "businesses need to change at a rate that can keep up with competitors, vendors, and the modern consumers' fleeting stream of impatient desires" (Waszkowki, 2019 p. 376). This increased dependence on technology along with evolving consumer demands and preferences call "for faster processes and automation in today's competitive landscape requiring more business applications to be built with greater speed and efficiency (Zavery, 2020; Google Workspace, 2022). It is vitally important for organizations to produce the applications needed in such a rapidly changing landscape and at the same speed (Sahinaslan et al., 2021; Google Workspace, 2022; Woo, 2020).

Additionally, this heightened tech use comes with the desire to digitize every object and every process today

(Sahinaslan et al., 2021), generating copious amounts of data, challenging data management systems and demanding processes requiring automation. Everything, every process is now a data point to be tracked, measured, and analyzed. Data, today, is the currency that influences all decisions in life now more than ever. All these changes necessitate the need to upgrade, repurpose or develop new software products used in the world of work and academics at a rapid pace. However, it is very difficult with traditional software development methods to produce flexible solutions which keep pace with such dynamic and rapidly changing demands and build rates to deliver results on time and on budget (Sahinaslan et al., 2021; Zavery, 2020).

Also, finding qualified personnel as well as the high cost of writing and updating corporate program codes can be complex especially on such a large scale. Thus, automating and modernizing is not proving easy for corporations (Sahinaslan et al., 2021; Johannessen & Davenport, 2021). Zavery (2020) admits that “the demand for faster processes and automation in today’s competitive landscape requires more business applications to be built with greater speed and efficiency. However, many companies lack the resources to address these challenges” (Zavery, 2020 Google Acquires AppSheet).

Waszkowski (2019) found that technical and development challenges were one main reason why companies did not decide to automate processes fast enough (p. 377). As a response to this dilemma, it has become expedient to equip both professional IT developers and non-IT personnel with skills in software development to tackle this enormous challenge. Organizations are increasingly turning to low code and no code software development technologies to fulfill growing demands for speedy application delivery and the creation of highly customized automation workflows (Zavery, 2020; Waszkowki, 2019; Sahinaslan et al., 2021).

This trend of embracing the low code/no code revolution is enabling organizations to reach the level of digital competency and speed of development required for the modern agile environment (Gartner, 2022). No code/low code technology and the empowerment of citizen developers has become a key enabler for faster and more impactful change allowing organizations to stay relevant by becoming more agile and resilient (Google Workspace, 2022). Gartner (2022) also noted that business technologies, enterprise-wide hyper automation and composability will be the key drivers accelerating the adoption of low code technologies through 2026.

This trend is logical and necessary. It presents businesses with a competitive advantage and is key to the future of work (Google Workspace, 2022). Gartner (2022) reports that low code application platforms (LCAP) show large market segment growth and citizen automation development is projected to grow at the fastest pace, with 30.2% growth forecast for 2023, and predicts further that by 2026, developers outside formal IT departments will account for at least 80% of the user base for low code development tools up from 60% in 2021 (Gartner, 2022). ElBatanony & Succi (2021) also note that no-code tools have been trending over the past few years, with Forbes placing the no-code or low-code movement as the most disruptive trend of 2021.

The Need for Computational Tools in Humanities and Social Science Research

This phenomenon of technology adoption and data-driven decision-making processes has not left out researchers, scholars, practitioners, and academics in the human and social sciences. Academia is part of the world of work, and researchers, scholars and academics are part of this work-world here described. Collins et al. (2011) assert that "it is difficult to gainsay the enormous change, almost a revolution, which has occurred in humanities research alongside the development of mass computing and mass communication" (p. 76). Yet, automation and this massive upskilling in new technologies, particularly software development, is still not a foreground issue to many in the humanities and social sciences (Collins et al., 2011).

Even though Struck (2018) observes that writing software has been and is increasingly a large part of scientific research this has not been the case for research in the humanities and social sciences. Chang (2017) laments that "in most other academic fields from physics to genetics, researchers rely on computers for everything from data analysis to modeling. But one area of scholarship that has gone largely untouched is the humanities" (Chang, 2017, Computational Tools). (Duca & Metzler, 2019) note that only a few software tools coming out of university incubators target social science researchers. The landscape showcases scenarios where researchers, scholars and practitioners in the humanities and social sciences still avoid the exploration of certain questions in their fields for lack of skills in developing the tools needed for such exploration.

Acknowledging this problem, Mimno, an interdisciplinary scholar in computer science and music explains that "there are real problems in computational analysis of text, from historical texts to contemporary texts, such as news articles and social media that we don't know how to solve" (as cited in Chang, 2017 Computational Tools). Up to this point, scholars have had to revise their research questions at best, and at worst they have dropped some research questions completely because of the absence of adequate software tools to help in exploring those questions, or researchers have had to bend to available tools in their research work (Morgan, 2018). Even more, scholars in the humanities at times end up modifying their original research questions in order to accommodate views shared by colleagues in IT and computer departments who would realize the development of project software and automated processes (O'Sullivan et al., 2015).

A study by Sage Publishing reveals that even though more than three quarters of humanities scholars and social science respondents to a survey agreed to have used software in their research and believe software is important or critical in their research, only 10% have developed their own (Duca, n.d.). It is safe to assume that the lack of critical technical skills in application development and tool mastery plays a key role in this dearth (Morgan, 2018). This is unfortunate as genuine questions with humanistic outlook tend to go unanswered since expertise in a field is needed to explore the details of a particular question that may not be obvious or intuitive to an out of discipline collaborator. Partners from outside a discipline may not fully appreciate the importance of an element to a discipline in the same way as one who is specialized in that field.

For fields of study as broad as the humanities and social sciences and with questions requiring depths of exploration of human emotions, culture, context, trends, thoughts and feelings, leaving research questions to be answered only with tools either developed for other disciplines or off the shelf tools developed for other purposes pushes researchers to adapt their inquiry to available tools. Researchers also avoid exploring difficult or time-consuming questions and analysis altogether, for lack of tools; or they end up working with tools which do not go far enough or in depth enough for humanities research and exploration. It is certain, therefore, that some areas of human and social science studies are undoubtedly going unexplored and uninvestigated at this time because scholars outside the disciplines are unaware of a particular scholastic area of inquiry in the humanities for which to develop tools and scholars in the discipline with discipline knowledge lack the technical skills to develop the needed tools themselves. Reliance in exploring certain humanities research areas and topics in the past, has depended on colleagues in the field of computing and IT. The pattern has been to collaborate with experts from IT or the computer sections of universities and other research institutions and accommodate what the "experts" in tool development confirm as possible to achieve in the research process (O'Sullivan, et al., 2015), from their point of view.

Some have suggested that scholars who come from dual or interdisciplinary backgrounds in the humanities and social sciences, with background in computer technology, should bridge this gap (Chang, 2017 Computational Tools), but generally, this is rarely the case. According to Mimno, statistical models are not built to replace humanities research, but computation can provide insights. It can provide clues, and it can double check whether a theory is plausible, but it is not going to give us any conclusions (as cited in Chang, 2017 Computational Tools). Scholarship still needs to be done. For O'Sullivan et al. (2015) technology should be used to support humanities agendas rather than dictating what such agendas should be. Fortunately, their research found that while not directly contributing to tool development, humanities scholars ensure that deliverables align with cultural and humanistic purposes (O'Sullivan et al., 2015). In like manner, other studies agree that disciplinary needs remain very important, and that a "one size fits all" approach to digital provision is probably doomed to failure (O'Sullivan et al., 2015 p. 80) and that technology could not replace human input" (Collins et al., 2011, p. 84). Pursuing a similar thought in the age of AI, Cameron (2023) recommends that AI "handle the optimization processes, allowing humans to focus on creativity, collaboration and leadership." (Cameron 2023, Navigating the future of work). Writing personalized and customizable software with code-free applications makes data more accessible and interactive, providing ease with updating, analyses, and visualization.

Uses of Computational Tools and Benefits of Research Software in the Humanities

Fortunately, low code and no code tools have arrived on the scene just in time when citizen developers are even more needed than before. According to Sahinaslan et al. (2021), low code application platforms help in faster application development with very little programming knowledge, rather than traditional programming approaches that require a certain expertise in the field of programming. Today, there is a shift in thinking in the tech world where developers recognize that to meet demand for computer and software tools, emphasis needs to be more on the goal to be accomplished than on the development of a tool needed for the task. Tech developers agree that software is a means to an end and the more streamlined and automatic the creation experience the more powerful software will be.

Software systems heavily support our daily lives in various areas, including research (Pinho et al, 2022). Therefore, software is a fundamental and vital part of research, especially when it provides the source code, files, algorithms, scripts, computational workflows, and executables that were created during the research process or for a research purpose (Gruenpeter et al. as cited in Barker et al., 2022). Using software enhances transparency, believability, reproducibility and replication, elements needed for veritable research and scientific work. To achieve these benefits Fortunato & Galassi (2021) recommend the use of free and open-source software which are computer programs which allow users to work, explore and transform them as needed, a characteristic common with no code platforms. These platforms thus lend themselves to API integrations and AI algorithms easily, thereby increasing computational power (Fortunato & Galassi, 2021). Figure 1 below shows an example of a software platform which accepts API integrations.

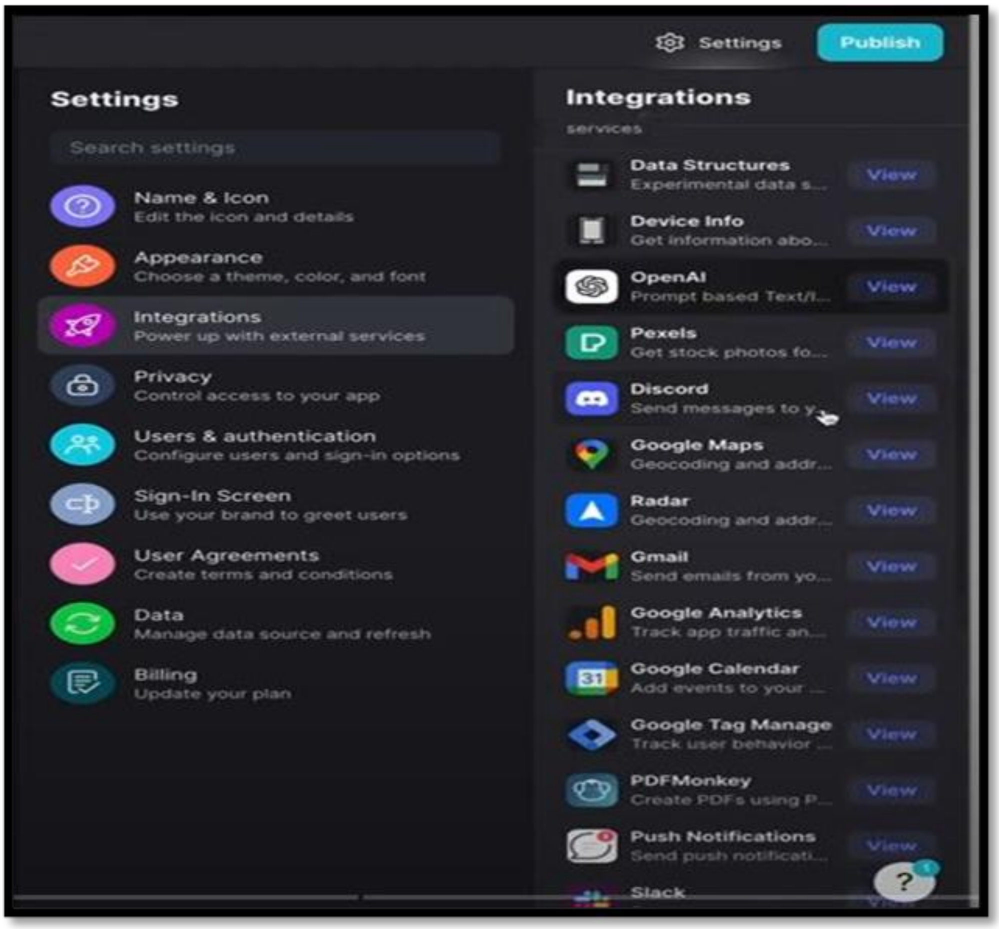


Figure 1. A No code Build Platform showing Integrations

Computational tools with such capabilities can be used during the entire research life cycle in the humanities and social sciences in a variety of ways, significantly influencing research and research results (Schnidler et al., 2020). According to Gentzkow and Shapiro (2014), software is necessary for empirical social science for asking good questions, designing statistical analysis, writing up results, digging up novel data, executing statistical analyses, simulating models, formatting results, and producing plots. They argue that software helps the researcher by introducing better ways to work so they spend less time wrestling with code and more time on the research problem. Software facilitates rerunning observations, regressions, and results without starting from scratch, thereby increasing automation, efficiency, and flexibility (Gentzkow & Shapiro, 2014). These elements help to cut time and costs once a reusable tool is built, whether the researchers are software engineers, database managers, computer scientists or not (Gentzkow & Shapiro, 2014) All these arguments strengthen why it is necessary for social science and humanities scholars to create and deploy software in their research. Duca and Metzler, (2019) concur that Software and computational tools are critical for social science research. They report survey findings in which 83% of respondents working in the social sciences said they make use of research software, with over 60% claiming that software is vital to their research, and more than half of the respondents believe software was important or very important to their work (Hettrick as cited in Duca & Metzler, 2019).

Big Data and Data Intensive Future in Human and Social Sciences Research

Strategically deploying code-free software in the creation, management and curation of data helps in dealing with the volume of information available today in more efficient ways. Big data issues and issues of dealing with data at scale in the research and analysis of humanities research questions, sometimes scares scholars from pursuing certain research lines. Usually, the volume, especially of qualitative texts, to analyze for themes, patterns and outliers is often not an easy task, especially if working by hand or in analog modes. "Using computers to analyze lots of data in a short time could be useful to humanities scholars who want to examine more books than one can read" (Chang, 2017 Computational Tools).Thus, software provides the means for easy analysis with text mining, pattern recognition, thematic and key word search capabilities, tools for sentiment analyses, graphics and visualization and offers speed of analyses, making software development attractive to scholars in the humanities.

Nowadays, many researchers are no more than a few clicks or taps away from a large body of literature, data, and software (Fortunato & Galassi, 2021). Therefore, the ability for analyzing text at scale with computational power brings in a whole new range of data points to be considered and thus a whole new set of insights to research. Researchers can sift through thousands of volumes of text and identify patterns that might not be obvious to human investigators or might be too time consuming to glean manually (Chang, 2017 Computational Tools). In addition, research work in the humanities needs easily customizable software as the mostly qualitative work seeks to tap into idiosyncrasies of individual users or subjects in research cases. One way to do so is to allow intuitive and idiosyncratic uses of tools for data collection, to provide even more valid results or conclusions about concepts, phenomena, and questions under study.

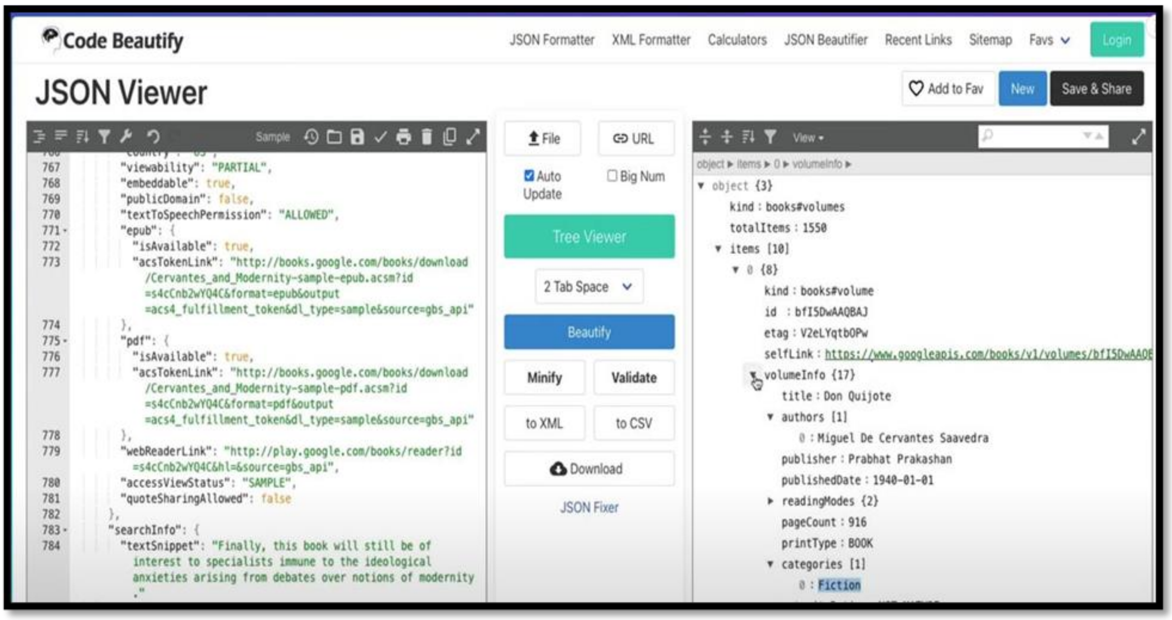
Researchers must master how to reduce large amounts of big, raw data into information and useful knowledge in any field today. Before now, according to Duca and Metzler (2019), the rapid evolution of tools for big data was a barrier to researchers looking to move into the field of computational social science but today, social scientists want to be prepared for a more data intensive future in research and thus, they want to be equipped with the skills, tools, and resources they need to work with big data and new technology. Duca and Metzler (2019) see a future in which more research is carried out with the help of technology, and they see a future in which researchers may increasingly become tool builders themselves.

No code, low code tools bring the idea or possibilities of customizing for human subjects in humanities research which could lead to answering direct and specific questions with the ability to observe and analyze human behavior down to the micro and individual level, which might not have been possible before now. This ability to customize is a very attractive reason to engage in software development in the humanities. Researchers at Wolox, a user research firm, combined purely humanistic research approaches with agile methodologies in their research on user behavior. They concluded that approaches in social and humanistic scientific research such as ethnography and in-depth interviews had to catch up to the agile rhythm that the software industry requires and that research about user behavior needed to adapt and evolve in order to fit into the fast-paced, agile software environment. They realized that the fusion of agile methodologies with in-depth ethnographic research was innovative and effective in delivering results about user behavior (Speranza, (2019).

Computational techniques can also apply to the study of cultural heritage, memory institutions, libraries, archives and digital culture and could be used in new statistical methods and data mining methods to give people a better ability to explore culture, literature and the world, the ability to create stories with more dynamic plots and personalized endings or mine data from poetry or showcase art in new ways using a variety of tools (Chang, 2017; Morgan, 2018; Pinder, 2022; Guhlin, 2018). Thus, computational tools give researchers the ability to share and disseminate their work any way they choose via all available media. Collins et al. (2011) reinforce this thought saying, “data must be discoverable, and researchers must be able to share their findings and work in progress. And new computing technologies have played an important role in helping researchers to share their data and findings, formally and informally” (p. 77).

Fortunately, today, it no longer should matter whether or not a researcher knows the difference between a markup language and a programming language (Swinhoe, 2017; O’Sullivan, 2015; Woo, 2020). The goal in the world of software development has been to democratize software development with “no experience needed” (Woo, 2020) but to put the emphasis on the question a researcher is answering and less on the tool being used. According to GitHub CEO, Chris Wanstrath, the future of coding is no coding at all, and the goal is to make software development easier and more approachable. The experts at GitHub predict that there will be less coding, yet many more people will be making software and expectations and ideas about collaboration and demands around what it means to learn and build software and work together is going to be a lot more imaginative and creative (Swinhoe, 2017 Github CEO). Morgan (2018) agrees stating that tool users want the easiest experience possible because as researchers, they are already overburdened so they long for tools that will give them the ability to fully realize their imaginations to produce something new and dramatically different. In fact, with the development and advancement in graphical user interfaces, drag and drop technologies, API integrations and even developments in AI technology, it is easy and efficient today to develop powerful software that can yield results and be easily adaptable, customizable, and “programmable” (Woo, 2020).

Figure 2 shows the ease with which builders can now call up Java Script Object Notations (JSONs) (left panel below) and render those scripts in more readable formats (right panel below), increasing ease of build and customization.



```
        "viewability": "PARTIAL",
        "embeddable": true,
        "publicDomain": false,
        "textToSpeechPermission": "ALLOWED",
        "epub": {
          "isAvailable": true,
          "acsTokenLink": "http://books.google.com/books/download
            /Cervantes_and_Modernity-sample-epub.acsm?id
            =s4cCnb2wYQ4C&format=epub&output
            =acs4_fulfillment_token&dl_type=sample&source=gbs_api"
        },
        "pdf": {
          "isAvailable": true,
          "acsTokenLink": "http://books.google.com/books/download
            /Cervantes_and_Modernity-sample-pdf.acsm?id
            =s4cCnb2wYQ4C&format=pdf&output
            =acs4_fulfillment_token&dl_type=sample&source=gbs_api"
        },
        "webReaderLink": "http://play.google.com/books/reader?id
          =s4cCnb2wYQ4C&hl=&source=gbs_api",
        "accessViewStatus": "SAMPLE",
        "quoteSharingAllowed": false
      },
      "searchInfo": {
        "textSnippet": "Finally, this book will still be of
          interest to specialists immune to the ideological
          anxieties arising from debates over notions of modernity
          ."
```

```
object ▸ items ▸ 0 ▸ volumeInfo ▸
▾ object {3}
    kind : books#volumes
    totalItems : 1550
  ▾ items [10]
    ▾ 0 {8}
        kind : books#volume
        id  : bfI5DwAAQBAJ
        etag : V2eLYqtbOPw
        selfLink : https://www.googleapis.com/books/v1/volumes/bfI5DwAAQE
      ▾ volumeInfo {17}
          title : Don Quijote
        ▾ authors [1]
            0 : Miguel De Cervantes Saavedra
          publisher : Prabhat Prakashan
          publishedDate : 1940-01-01
        ▸ readingModes {2}
          pageCount : 916
          printType : BOOK
        ▾ categories [1]
            0 : Fiction
```

Figure 2: JSON (left) more Readable Text (right) (Petito, 2023)

The Build: Characteristics of No Code Low Code Tools

No code development depending on graphical user interfaces (GUI) makes the process of development palatable to those who do not write code, yet they can still produce quality software in their fields. These products are developed with certain characteristics that make them particularly attractive to non-technical developers in the social and human sciences. They are created with a high level of abstraction, they feature advanced graphical user interfaces with drag-and-drop capabilities, they encourage reusable components which enable simplified and rapid processes for developing software solutions. They mostly reside in cloud-based environments which leads to enhanced ease of use and are great with visual representations, allowing people with minimal or no programming skills and knowledge to create software (Binzer & Winkler, 2022).

Barker et al, 2022 further discuss elements that software useful for research should possess. They should be findable, searchable, provide complete metadata, be usable and reusable and be interoperable using APIs (Barker et al., 2022). No code low code applications meet all these parameters and recommendations. ElBatanony and Succi (2021) also describe the basic building blocks of software tools. They are made up of "systems" or "components" available in the form of libraries, integrations, APIs, SDKs, which other designers and developers can tap into to build and design what they want. Knowing this basic fact eases up on the learning curve for builders about what to include in a build as this provides a basic blueprint and resources for building elements. This supports the claim that “You’ve already solved the problem of creating software if you know what you want” (ElBatanony & Succi, 2021, p. 34).

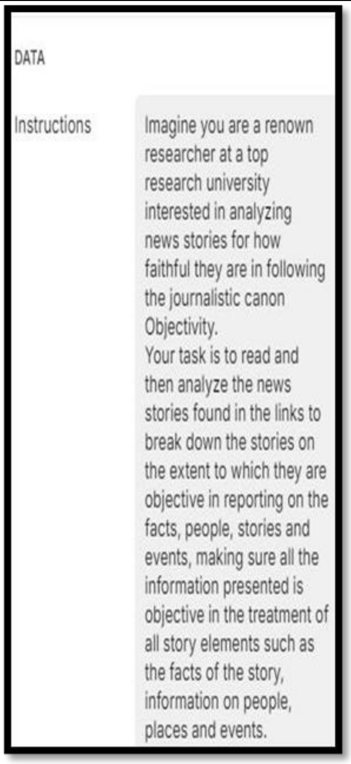
Social science and humanities builders should adopt the mantra “write once, run anywhere" advantage that software allows (ElBatanony & Succi, 2021). Thus, this paper challenges scholars to imagine ways in which the content and research materials they engage with could be codified, prepared, and reduced to forms that invite automation and application development aligning practices in the human and social sciences with the technological leap forward. According to ElBatanony and Succi (2021), the cornerstone principle that future software will rely on is the Application Programming Interface (API) which enables various pieces of software to easily and seamlessly talk to each other. They also recommend the use of open-source software designed in two stages. The first stage based on six core development principles namely, configuration driven development, APIs, open-source, cross-platform, cloud computing, and design systems and the second stage, they recommend, should involve artificial intelligence and natural language processing (ElBatanony and Succi, 2021).

Meanwhile Duca and Metzler (2019) have identified a wide variety of tools available for social and humanities research and have catalogued and described the nature and characteristics of these tools and software packages to help researchers navigate the landscape. They found tools for social media analysis and research, tools for annotation, coding, and text labeling as well as tools for surveying and sourcing research participants (Duca & Metzler, 2019). They also make the interesting and concerning remark that many of these tools are still created and built by people outside of the social sciences either as hires or collaborators. In addition, it was not clear from the study whether these tools were no code, or low code builds. It is thus safe to conclude, therefore that, even though tools might be proliferating for social science and humanities research, the users of these tools are still not the builders when it comes to social sciences and humanities work, giving even more support to the arguments raised in this paper as to why scholars in the humanities and social sciences need to build their own tools today and going forward.

Artificial Intelligence Supercharges Low Code No Code Tools

According to Woo (2020), “we can incorporate data driven intelligence in apps where artificial intelligence now is able to automatically produce code, an ambition referred to as “program synthesis.” These days, computer scientists are able to write programs that write themselves where you simply describe your desired program in natural language and the computer generates the results” (Woo, 2020). The availability of code-free tools powered by Artificial Intelligence provides access to software development capabilities like never before. With algorithms and prompt engineering today, we can train AI models with specific instructions, to even further customize results for builders and users within low code no code platforms. Platforms such as Glide (www.glideapps.com) and Zapier (www.zapier.com) have now embedded AI capabilities into their software platforms, further enhancing usability and ease of customization. Adding in generative AI tools today, code free tools become even more efficient and powerful. In Figure 3 below for instance, OpenAI’s ChatGPT is prompted to analyze and assess a news story for how well it follows journalistic canons. The left column of the figure shows a prompt given to the AI model and it generates the analysis and assessment in the right column of the figure, showing a score of three on five for the parameter analyzed.

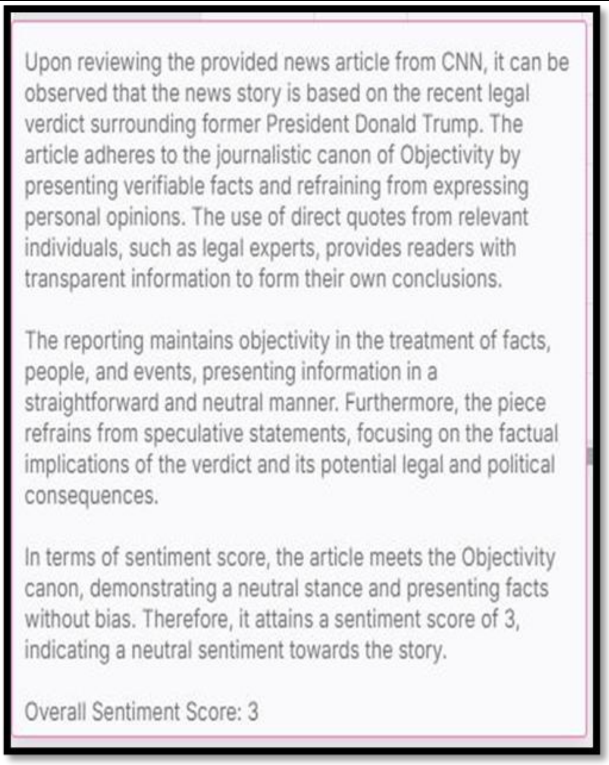
Generative AI (GenAI) even enhances accessibility to code for non-coders when they need to achieve specific functionalities within their no code low code tools. With GenAI today, a builder can call up code in most programming languages such as HTML, YAML and XML to actualize certain functionalities with a simple copy-paste to enhance their builds. With GenAI today, the non-technical developer can accomplish things on the low code no code platforms that were heretofore thought impossible as seen in Figures 1, 2 and 3 above. In Figure 4b below, Microsoft Copilot generates the HTML code for the table in Figure 4a for the builder to copy and paste in their build thereby making it easy for the non-technical developer to easily recreate the table.



DATA

Instructions Imagine you are a renown researcher at a top research university interested in analyzing news stories for how faithful they are in following the journalistic canon Objectivity.
Your task is to read and then analyze the news stories found in the links to break down the stories on the extent to which they are objective in reporting on the facts, people, stories and events, making sure all the information presented is objective in the treatment of all story elements such as the facts of the story, information on people, places and events.

Instruction Prompt for AI



Upon reviewing the provided news article from CNN, it can be observed that the news story is based on the recent legal verdict surrounding former President Donald Trump. The article adheres to the journalistic canon of Objectivity by presenting verifiable facts and refraining from expressing personal opinions. The use of direct quotes from relevant individuals, such as legal experts, provides readers with transparent information to form their own conclusions.

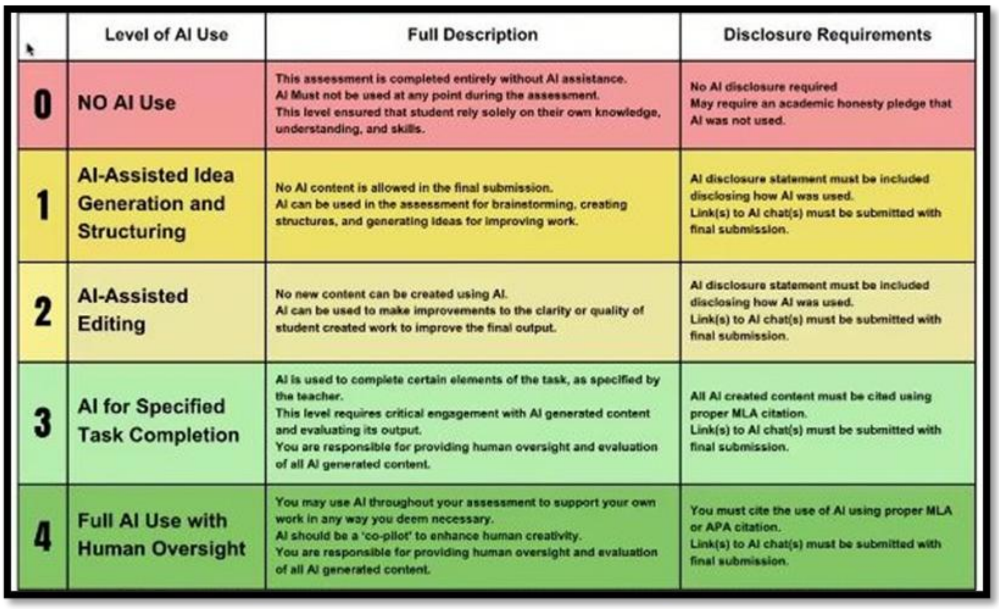
The reporting maintains objectivity in the treatment of facts, people, and events, presenting information in a straightforward and neutral manner. Furthermore, the piece refrains from speculative statements, focusing on the factual implications of the verdict and its potential legal and political consequences.

In terms of sentiment score, the article meets the Objectivity canon, demonstrating a neutral stance and presenting facts without bias. Therefore, it attains a sentiment score of 3, indicating a neutral sentiment towards the story.

Overall Sentiment Score: 3

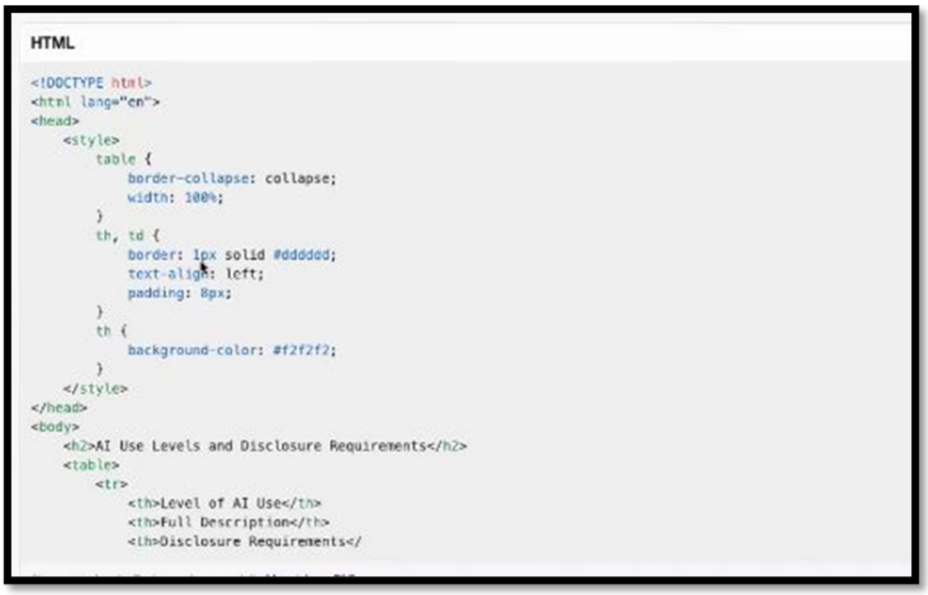
AI Output showing a score of 3/5

Figure 3. AI Prompt and AI Generated Score from Prompt



	Level of AI Use	Full Description	Disclosure Requirements
0	NO AI Use	This assessment is completed entirely without AI assistance. AI Must not be used at any point during the assessment. This level ensured that student rely solely on their own knowledge, understanding, and skills.	No AI disclosure required May require an academic honesty pledge that AI was not used.
1	AI-Assisted Idea Generation and Structuring	No AI content is allowed in the final submission. AI can be used in the assessment for brainstorming, creating structures, and generating ideas for improving work.	AI disclosure statement must be included disclosing how AI was used. Link(s) to AI chat(s) must be submitted with final submission.
2	AI-Assisted Editing	No new content can be created using AI. AI can be used to make improvements to the clarity or quality of student created work to improve the final output.	AI disclosure statement must be included disclosing how AI was used. Link(s) to AI chat(s) must be submitted with final submission.
3	AI for Specified Task Completion	AI is used to complete certain elements of the task, as specified by the teacher. This level requires critical engagement with AI generated content and evaluating its output. You are responsible for providing human oversight and evaluation of all AI generated content.	All AI created content must be cited using proper MLA citation. Link(s) to AI chat(s) must be submitted with final submission.
4	Full AI Use with Human Oversight	You may use AI throughout your assessment to support your own work in any way you deem necessary. AI should be a 'co-pilot' to enhance human creativity. You are responsible for providing human oversight and evaluation of all AI generated content.	You must cite the use of AI using proper MLA or APA citation. Link(s) to AI chat(s) must be submitted with final submission.

Figure 4a. Table for Copilot to Generate HTML Code (PSU TwT 2024)

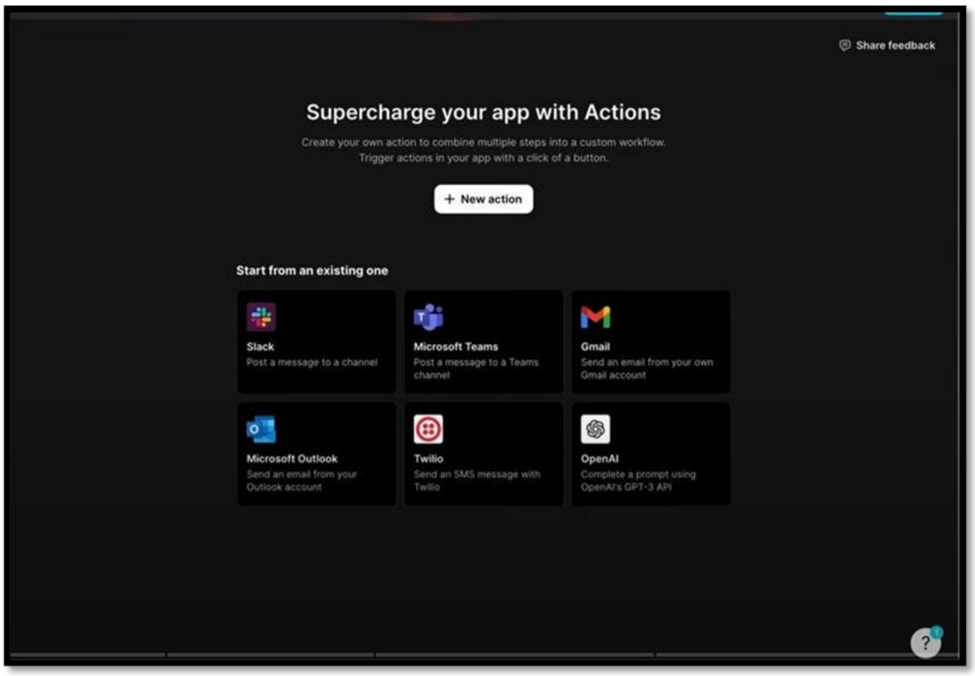


HTML

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <style>
        table {
            border-collapse: collapse;
            width: 100%;
        }
        th, td {
            border: 1px solid #dddddd;
            text-align: left;
            padding: 8px;
        }
        th {
            background-color: #f2f2f2;
        }
    </style>
</head>
<body>
    <h2>AI Use Levels and Disclosure Requirements</h2>
    <table>
        <tr>
            <th>Level of AI Use</th>
            <th>Full Description</th>
            <th>Disclosure Requirements</
```

Figure 4b. Copilot Generates HTML Code for Table in 4a (PSU TwT 2024)

ElBatanony and Succi (2021) trace the progress in computer technologies, from punched cards to the assembly language, to C, to JavaScript and Python, to APIs, cloud computing etc. and all of these iterations in computer development have introduced ease of use to the point now where any person would be able to create, edit, modify, and deploy software, with a much lower level of required knowledge. They can issue commands, add actions, build paths of executions with various strings of code that they do not have to write or know how to write but can easily find and use from the available resources and systems (ElBatanony & Succi (2021). Figure 5 shows The Action function in glideapps (www.glideapps.com), which enables developers to combine multiple steps into a custom workflow which can be triggered with a single click.



Supercharge your app with Actions

Create your own action to combine multiple steps into a custom workflow.
Trigger actions in your app with a click of a button.

Share feedback
+ New action
Start from an existing one
Slack
Post a message to a channel
Microsoft Teams
Post a message to a Teams channel
Gmail
Send an email from your own Gmail account
Microsoft Outlook
Send an email from your Outlook account
Twilio
Send an SMS message with Twilio
OpenAI
Complete a prompt using OpenAI's GPT-3 API

Figure 5. Showing Action Capabilities in Glideapps

No Code Tool Samples and Use Cases

Discussing particular no code tools as they exist today could be instructive to this discussion but because technology changes at the speed of light, tools available today may not be there tomorrow or they may undergo modifications, so this author is hesitant to recommend particular tools. However, to illustrate what this discussion is about it is best to mention a few currently available tools. Google, Amazon, and Microsoft have a variety of no code low code tools nowadays which support building business applications, and which run on major platforms such as Android, iOS, Windows, Mac, and Linux (ElBatanony & Succi, 2021). AppSheet for instance, a no code platform, was recently acquired by Google for its robustness in tackling high level enterprise workflows and democratizing software development (Google Workspace, 2022) while tools like Zapier, another no code platform allow for integrations from sources that accept API integrations (Figure I above). Canva has become the go-to tool for graphic design while Microsoft Power Apps, a no code platform has been used to replace Microsoft Visual Basics in a decision support course in a university business program with better learning outcomes for students (Wang & Wang, 2022). In Turkey, SetXRM, a tool developed for the business sector, proofs to yield better results faster and cheaper using low code development platforms according to company data (Sahinaslan et al., 2021).

Glide software application, another no code platform, has a goal of empowering a million no code citizen software builders in the near future. Wix, Webflow and Google's Flutter have significantly democratized the no-code web development arena. Flutter now supports all the major platforms (ElBatanony & Succi, 2021). Also, today, there are a slew of exciting new productivity tools with AI capabilities for a variety of workplace and everyday solutions. Otter converts speech to text, a capability which now enables transcribing qualitative research data such as lengthy in-depth interviews for analysis, Evernote is useful for note taking and organization and helps in meeting summaries and highlighting action items for users, Suno helps in creating songs, music and jingles, among several other productivity tools today. However, all these examples are either more marketplace and business solutions-oriented than are they tools to answer specific and purely humanistic questions. This is why the case for tools developed specifically for social sciences and the humanities supercharged with AI technologies is still sorely lacking and needed today.

Social Science and Humanities Experts as Citizen Developers

In profiling citizen developers Binzer and Winkler (2022), characterize them as non-IT employees who have no formal IT education, but possess great expertise. They are-tech-savvy or tech-excited and curious. They desire change and have a natural motivation for continuous learning and upskilling (Binzer & Winkler, 2022). This definition perfectly fits citizen developers in the social and human sciences who should be developing these tools in their expert fields. They are experts in their particular domain, they know and can define and ideate requirements for software in their expert domain (Rokis and Kirikova (2022) and can help proffer solutions to humanistic problems in their areas of expertise.

Scholars must, therefore, learn needed skills in developing software to solve problems in their fields and not just appropriate business software to retrofit for humanities and social science research. We need subject matter expertise, combined with the curiosity, intelligence, and innovation to produce citizen expert developers to create the types of software here discussed. Duca and Metzler (2019), noted that these types of developers are mostly found in universities and centers for research. They found that out of 481 tools they investigated over one hundred were developed within university research projects, by dedicated communities, consortia, or other nonprofit organizations (Duca & Metzler, 2019). So, developers who should be building these tools here described are mostly found in institutions of higher learning and research.

Implications for Universities and Considerations for Policy Makers in Higher Ed

However, the attitude towards the adoption of no code low code development in the academies and research centers has been lagging. Somehow building software is still seen in the academy as a matter reserved just for the computer science divisions. There might have been a time when that was the case but today with the rise of the no code low code platforms, there is a revolution in software development and engineering that is still to be embraced in the academies (Struck, 2018). Even though in a few cases research conferences and workshops are dedicated to low code development with developers sometimes invited to coauthor papers about their builds (Pinho et al., 2023), research software development whether with code or not still lacks recognition as actual and major research and academic achievement in most universities.

College administrators, protocols and promotion plans still do not reward faculty for developing skills in software building using no code low code tools most likely because of the centuries-long tradition of article publications and citeability as a prerequisite of recognition (Struck, 2018). These products and artifacts are still not considered and valued for promotion and tenure. Yet these tools take time, focus, discipline, concentration, multiple steps, and multiple iterations from ideation, through the building, testing, redesigning, and deployment phases. Even what it takes just to garner awareness, publicity and adoption of existing tools should be worthy of consideration. Hoogsteen and Borgman (in Duca & Metzler, 2019) noted that whilst many commercial tools are available, the more innovative ones are coming out of academia. Binzer and Winkler (2022) also investigated factors that determine adoption of citizen development and found that active top management support positively affects citizen development adoption. Academia, therefore, should embrace the already industry-led prevalent trend towards code-free application design, development and deployment in university curricular and promotion protocols, demystifying software development and destigmatizing code-free development.

Reality and Challenges of Tool Development: Ease not Easy

It is worthwhile to note in closing that no code low development is not a panacea (Morgan, 2018). Rokis & Kirikova (2022) discuss limitations that low code no code developers face and should be aware of as they develop these tools. Practitioners experience different kinds of challenges when customizing user interfaces, implementing logic, integrating third-party services, or solving more complex development issues which could sometimes even require access to code. Issues with interoperability, extensibility and scalability could be some restrictions that developers run into. They also offer suggestions of ways around such blockades and call for platform vendors to provide elaborate documentation and learning resources, also suggesting a recommender system that would use prior learning from previously developed applications (Rokis & Kirikova, 2022).

Glideapps, for instance, provides a document library, a “university” and an active and helpful user community for peer-to-peer support. They also have a robust template library to offer a soft start to builders. Barker et al. (2022) make the case for open source in research software so practitioners can get passed issues in working with tools already developed. Indeed, Generative AI can easily provide code and other helpful assistance now, helping builders bypass these stumbling blocks as seen in the figures and examples above. This is where pairing no code low code development with artificial intelligence really makes a difference for non- technical developers.

New Thinking, Prognostics and Recommendations

What is essential is that scholars in the humanities need to be able to clearly articulate the problem they are solving, the questions they want to answer and then examine their information sources to isolate elements that constitute data they can prepare and reduce to codification, far more than worrying about whether they themselves are writing the code or not (Swinhoe, 2017; Pinder, 2022). They must think from a logical critical approach to find elements of their problem that could be enhanced with software and think of how raw data can be conceived in ways that lend itself to the application of software (Chang, 2017; Pinder, 2022). Mimno says that “statistical models and text mining can identify broad themes and hone in on interesting topics where detailed scholarship could be most useful” (as cited in Chang, 2017 Computational Tools). This should be the new way of thinking by humanities and social science researchers.

With technology now leading the day, skills in software development, automation and productivity are needed across the board. Skills in software development will become crucial in future, therefore, quick, and agile software development skills will become even more important and valuable. Natural Language Processing (NLP) is the way forward and with Artificial Intelligence (AI) software development is now sufficiently democratized (ElBatanony & Succi, 2012). Code-free software enables quick and customizable application creation, without the time-wasting and labor-intensive processes of writing code (Woo, 2020; Waszkowki, 2019). These tools and platforms are resilient and agile. They take less time to develop, and they deliver excellent results as illustrated by the examples and discussions above. If no code platforms are good for Google and Microsoft, and Amazon, they certainly are good enough for researchers in the human and social sciences. Today, if we combine core principles of software development assisted by artificial intelligence, it will be only a matter of time for social science and humanities scholars to become the “developer in every home” as suggested by ElBatanony & Succi (2021).

Conclusion

Scholars in the humanities ask and answer questions about our very existence and our human nature daily and there are scores of questions that are still unanswered as well as new questions being created today by social media, climate change, future studies and lots of other areas that need exploration. The ability these days to research from a distance, uncover new archeological sites, recast old questions in new ways makes work in the human and social sciences to be more alive today than ever. With technology now leading the day, skills in software development will be needed across the board without exception, including in humanities and social science research.

Empowered by the intuitive, flexible, and increasingly powerful features of no code low code development tools, software products and technologies supercharged with Artificial Intelligence capabilities enable citizen technologists to develop lightweight solutions to meet business unique needs for enhanced productivity, efficiency, and agility. Today, no code, low code development has democratized software development, encouraging a legion of expert citizen developers and scholars in the Humanities and social sciences who can now tap into these platforms to develop their own tools for their own very questions. It is, thus, possible, and imperative that scholars develop these skills needed in writing software in the human and social sciences. They should go for it without further hesitation.

Notes

This article is one of three in this series on no code low code builds supercharged with Artificial Intelligence. Forthcoming are two other articles. The second article will discuss a blueprint for building no code low code tools incorporating AI and the final article will discuss a sample build called them mEditor, a platform for analyzing and labeling news stories. A website and app address to access The mEditor will also be provided then. Lookout for the next two forthcoming articles for a comprehensive look at the topic.

References

Barker, M., Hong, N., Katz, D., Lamprecht, A., Martinez-Ortiz, C., Psomopoulos, F., Harrow, J., Castro, L.,Gruenpeter, M., Martinez, P., & Honeyman. T (2022). *Introducing the fair principles for research software Scientific Data (2022*). 9:622 https://doi.org/10.1038/s41597-022-01710-x

Binzer, B., & Winkler, T. (2022). Democratizing software development: A systematic multivocal literature review and research agenda on citizen development. Carroll et al. (Eds.) *AG 2022* N. Springer International Publishing https://doi.org/10.1007/978-3-031-20706-8_17

Cameron, M. (2023, December 18). *Navigating the future of work: 10 ways to prepare for the AI workplace*. Forbes Technology Council. https://www.forbes.com/councils/forbestechcouncil/2023/12/18/navigating-the-future-of-work-10-ways-to-prepare-for-the-ai-workplace/

Chang, A. (2017). *Computational tools for the Humanities. Cornell University*. Retrieved from https://as.cornell.edu/news/computational-tools-humanities.

Collins, E., Bulger, M. & Meyer, T. (2011). Discipline matters; Technology use in the humanities. *Arts & Humanities in Higher Education*, 11(1-2) 76-92.

Duca, D. (n.d.). Using and developing software in social science and humanities research: The ecosystem of software and technologies in social science research. *Tools & Technology*, Sage Publishing

Duca, D., & Metzler, K. (2019). *The ecosystem of technologies for social science research (White paper).* London, UK: Sage. doi: 10.4135/wp191101

Fortunato, L., & Galassi, M. (2021). *The case for free and open source software in research and scholarship.* Phil.Trans. R. Soc. A 379: 20200079. The Royal Society Publishing. https://doi.org/10.1098/rsta.2020.0079.

Gartner. Press Release (2022). *Gartner forecasts worldwide low-code development technologies market to grow 20% in 2023.* Newsroom, Stamford, Conn., Dec.

Gentzkow, M., & Shapiro, J. (2014). Code and Data for the Social Sciences: A Practitioner's Guide. Chicago Booth and NBER, University of Chicago mimeo. http://faculty.chicagobooth.edu/matthew.gentzkow/research/CodeAndData.pdf

Glideapps.com (2024). *The no code platform for modern work software*. https://www.glideapps.com/platform

Google Workspace Blog (2022). Google Workspace and AppSheet join PMI citizen developer partner program. Partners, Product Announcements Retrieved from https://workspace.google.com/blog/product-announcements/google-workspace-and-appsheet-join-the-pmi-citizen-developer-partner-program

Guhlin, M. (2018). Humanities go digital: Coding and storytelling. Technotes Bloghttps://blog.tcea.org/tag/twinery/

Johannessen, C. & Davenport, T. (2021) *When low-code/no-code development works-and when it doesn't. Technology and analytics.* Harvard Business Review

Morgan. P.C. (2018). The consequences of framing digital humanities tools as easy to use. *College & Libraries, 25*(3), 211-231.

O'Sullivan, J., Jakacki, D. & Galvin, M. (2015). Programming in the digital humanities. *Digital Scholarship in the Humanities*, *30*. Supplement 1.

Penn State University (2024) *Teaching with technology (TwT-2024 spring series) Enhancing course materials using generative AI* (4.10.2024)

Petitto, R. (2023, August 21). Glide 101: *Everything you need to know about call API and Query JSON* [Video] YouTube. https://www.youtube.com/watch?v=7F4e9JatqeM&t=775s

Pinder, N. (2022). Computational thinking and data analysis go hand-in-hand. ISTE Blog Retrieved from https://iste.org/blog/computational-thinking-and-data-analysis-go-hand-in-hand

Pinho, D., Aguiar, A., & Amaral, V. (2022). What about the usability in low-code platforms? A systematicliterature review. *Journal of Computer Languages*. https://doi.org/10.1016/j.cola.2022.101185

Rokis, K. & Kirikova, M. (2022). Challenges of low-code/no-code software development: A literature review. E. Nazaruka et al. (Eds.): *BIR 2022, LNBIP 462*, pp. 3–17, 2022. Springer International Publishing. https://doi.org/10.1007/978-3-031-16947-2_1

Sahinaslan, E. Sahinaslan, O. & Sabacioglu, M. (2021). Low-code application platform in meeting increasing software demands quickly: SetXRM. *Fourth International Conference of Mathematical Sciences AIP* Conf.Proc.2334,070007

Schindler, D., Zapilko, B. & Kr¨uger, F. (2020). Investigating Software Usage in the Social Sciences: A Knowledge graph approach. A. Harth et al. (Eds.): *ESWC 2020, LNCS 12123*, pp. 271–286, 2020. Springer International Publishing, https://doi.org/10.1007/978-3-030-49461-2_16

Speranza, S. (2019, June 3). *User research: What is the role of social sciences in software development?* https://medium.com/wolox/user-research-what-is-the-role-of-social-sciences-in-software-development5dfd03381df

Struck, A. (2018). Research Software Discovery: An Overview. *IEEE 14th International Conference on e-Science* DOI10.1109/eScience.2018.00016

Swinhoe, D. (2017). GitHub CEO: *"The future of coding is no coding at all": On the eve of celebrating its first 10 years, GitHub outlines where the next 10 could go.* Opinion, IDG Connect

Wang, H. & Wang, S. (2022). Improving student performance by introducing a no-code approach: A course unit of decision support systems. *Journal of Information Systems Education, 33*(2), 127-134, Spring.

Waszkowski, R. (2019). Low-code platform for automating business processes in manufacturing. *IFAC Papers Online, 52*(10), 376-381.

Woo, M. (2020). The rise of no/low code software development-No experience needed? *Engineering, 6*, 960-961.

Zavery, A. (2020). *Google acquires AppSheet to help businesses create and extend applications-without coding.* Google Workspace, product announcements, Jan. Retrieved from https://venturebeat.com/business/google-acquires-no-code-app-development-platform-appsheet/

Author Information

Beatrice Epwene

https://orcid.org/0009-0001-5776-9473

Penn State University

777 West Harrisburg Pike, Harrisburg PA

United States

Contact e-mail: *bne5066@psu.edu*