INFORMATION SYSTEMS EDUCATION JOURNAL

Volume 19, No. 5
October 2021
ISSN: 1545-679X

In this issue:

iscap
Information Systems & Computing Academic Professionals

The **Information Systems Education Journal** (ISEDJ) is a double-blind peer-reviewed academic journal published by **ISCAP** (Information Systems and Computing Academic Professionals). Publishing frequency is six times per year. The first year of publication was 2003.

ISEDJ is published online (https://isedj.org). Our sister publication, the Proceedings of EDSIGCON (https://proc.iscap.info) features all papers, panels, workshops, and presentations from the conference.

The journal acceptance review process involves a minimum of three double-blind peer reviews, where both the reviewer is not aware of the identities of the authors and the authors are not aware of the identities of the reviewers. The initial reviews happen before the EDSIGCON conference. At that point papers are divided into award papers (top 15%), other journal papers (top 25%), unsettled papers, and non-journal papers. The unsettled papers are subjected to a second round of blind peer review to establish whether they will be accepted to the journal or not. Those papers that are deemed of sufficient quality are accepted for publication in the ISEDJ journal. Currently the target acceptance rate for the journal is under 40%.

Information Systems Education Journal is pleased to be listed in the Cabell's Directory of Publishing Opportunities in Educational Technology and Library Science, in both the electronic and printed editions. Questions should be addressed to the editor at editor@isedj.org or the publisher at publisher@isedj.org. Special thanks to members of ISCAP/EDSIG who perform the editorial and review processes for ISEDJ.

INFORMATION SYSTEMS
EDUCATION JOURNAL

Student Group Satisfaction Perceptions using Agile in a Project-Based Course

Andrea R. Hulshult
hulshuar@miamioh.edu
Computer & Information Technology Department
Miami University Regionals
Hamilton, OH 45011, USA

Abstract

This study aims to examine the effectiveness and value of using Agile work practices to enhance group satisfaction in project-based courses. This study explores student perceptions of using Agile in a group, project-based course to support how Agile can be utilized in higher education to positively enhance group collaboration and teamwork. Surveys completed by students indicate that students found value in using the Agile mindset and Agile practices in a project-based course. Students ranked MoSCoW prioritization method and storyboards the highest of all the Agile practices.

Keywords: Agile, group work, teamwork, project-based courses

1. INTRODUCTION

This study aims to examine student perceptions of using Agile practices to enhance group work satisfaction in project-based courses. An IS/IT education program at a public university has a project-based introductory Agile course. Students collaborate as groups to complete a project for a client, usually a local non-profit. Students use the Agile way of working and Agile practices to complete the 11-week project. At the end of the project, students answered a 14-question Likert survey regarding their perception of Agile.

When working with student groups in an introductory-level course on Agile, more is unknown than known. Students do not know how they well they will collaborate as a team and they do not know much about the project they are being asked to develop. Student groups can be challenging, even volatile, uncertain, complex, and ambiguous (VUCA). VUCA was coined by the United States military and is currently being used to describe what citizens of the globe face daily, including climate change, societal and political turmoil, and wealth inequity (LeBlanc, 2018). The unknown about student groups and projects can be described as VUCA. Volatile, uncertain, complex, and ambiguous can describe groups and group work. Student groups can be volatile with clashing personalities. They can be uncertain since there is no protocol on how to work as a team or if the team's communication is lacking. Groups can be complex due to the competing priorities of students including jobs, family life, and academics. When a group has no leadership, poor communication, or does not understand the assignment at hand, they can be ambiguous.

However, project-based learning is a staple in many higher education courses and is considered a key component of a student's higher education experience (Frame, Cailor, Gryka, Chen, Kiersma, & Sheppard, 2015). Agile is a possible approach to help students have a positive group experience and increase work quality (Hulshult & Krehbiel, 2019; Woods & Hulshult, 2018; Krehbiel et al., 2017; Pope-Ruark et al., 2011; Rico & Sayani, 2009).

This paper is organized as follows. First, for the context and background, an overview of group work. Second, a brief history of Agile is presented. Third, a brief overview of the course and project is presented. Fourth, the research process and results are discussed.

2. LITERATURE REVIEW

Group Work

Group work is a common teaching methodology that provides students with intellectual advantages and practical workforce experiences. Group work is an effective learning strategy because it requires students "to negotiate meaning with their peers, share ideas, collaborate, and reflect and report on past learning experiences" (Beccaria et al., 2014, p. 1095). When students collaborate, they learn to manage unscripted situations, work together, and navigate diverse, complex issues (Beccaria, Kek, Huijser, Rose & Kimmins, 2014; Kuh, Kinzie, Buckley, Bridges, & Hayek, 2006). Employers consistently rate teamwork as one of the most important soft skills (CompTia, 2015), so offering group work experiences benefits students in and out of the classroom.

Group work is also both academically and socially beneficial (Beccaria et al., 2014). Students develop a greater sense of group processes and group dynamics, communication and leadership styles, critical thinking, problem-solving and social skills, and they may experience personal growth" (Beccaria et al., 2014, p. 1095). Collaborative learning activities help students gain the ability to resolve problems and conflict, communicate effectively, set goals, manage time and tasks, and observe team dynamics (Beccaria et al., 2014).

While group projects are an effective tool for student learning, they also present challenges (Woods & Hulshult, 2018). Students dislike group projects for multiple reasons including, being in a group where they end up doing all of the work, having to get to know new people who they may not get along with, or having to find time to work as a group. Students have competing priorities such as other classes, jobs, or family responsibilities, which limit the amount of time they have to work on group projects (Woods & Hulshult, 2018).

Faculty have similar concerns about group work in courses. Students may have a solid understanding of the course material, but when organized into groups students may end up arguing and producing poor quality work (Woods & Hulshult, 2018).

Agile

A promising approach to enhance collaboration and group projects is Agile. According to ICAgile, "agile is not a process, methodology, or framework; it is a mindset that welcomes uncertainty, embraces challenges, empowers individuals, and views failure as a learning opportunity. Adopting an agile mindset unleashes the brilliance of people and teams, which enables rapid discovery and faster innovation" (ICAgile, Mission, n.d.).

Krehbiel et al., (2017) state that Agile is a collection of practices aimed at enhancing group collaboration. Agile was developed in the computer software industry in 2001 to manage software development projects. Agile teams focus on collectively articulating their goals, reflecting on their work and making necessary adjustments, having authentic group interactions, improving team dynamics, and encouraging innovation (Smith & Sidky, 2009). The practices built into the Agile process help teams get real-time feedback on their work, reflect on their functioning as team by discussing about what is going well and what needs improved, make adjustments to their work, and repeat. Agile teams have higher quality outcomes and better meet their customers' needs compared to traditional project management models (Krehbiel, et al., 2017).

The Agile way of working and Agile practices is slowly making its way into higher education. There is a small number of faculty researching how Agile can be applied to higher education to improve teaching and learning.

Agile practices provide teams with tools to help them work more collaboratively as a group. These Agile practices help teams to communicate (stand-ups, prioritization, user stories), share ideas (planning, retrospectives), reflect on their work and make improvements (retrospectives), and be accountable (story boards and user stories). Some of these Agile practices are discussed in Survey Results and Discussion. Agile projects also have a cadence that help team members settle into a rhythm of the project work cycle.

Agile Project-Based Course

An IS/IT education department at public university offers a three-course concentration in Agile practices. Successful completion of each course provides students with an ICAgile certification. These ICAgile certifications can help students obtain Agile jobs in the workforce. The first course is an introduction to Agile, the second course is Agile Product Ownership, and the third is Agile Project Management. The research in this study was conducted in the introduction to Agile course. In this course students earn their ICAgile Certified Professional certification. This course is a prerequisite for the other two courses in the concentration. These courses are taught by ICAgile certified faculty.

In the introductory course, students spend four weeks learning the Agile mindset, Agile practices, and the Agile project lifecycle. The remaining weeks of the course are spent working on a project. Students work in teams of four to five students to develop a solution for a customer in the local community or for a department at the university. The client for the project discussed in this paper is a non-profit organization in the local community who needed market research conducted and analyzed to develop possible fundraising opportunities. As a non-profit, this customer relies on fundraising events throughout the year for revenue. This non-profit approached the university's community engagement office, who put the customer in contact with the IS/IT department. This customer wanted students to propose new fundraising ideas that targeted the 18-25 age group with a budget to host the event. Both sections of the course were divided into three teams per section and each team conducted their own market research and developed a fundraising proposal for the customer. The project was 11-weeks in duration. This course also has a service learning designation since the class works on a project for a client.

The 11-weeks was divided into five two-week iterations mirroring an Agile project. The first week of the project was spent allowing the teams to organize and plan. The teams selected user stories to complete for each two-week iteration. At the end of each iteration, the teams would conduct an Agile showcase for the customer. The showcases were held in-person or virtually based on the customer's schedule. For this project, the customer came to the class three times, and attended virtually two times. After each showcase, teams would conduct an Agile retrospective, and then spend a day planning for the next iteration based on the customer's feedback.

3. RESEARCH METHOD

This study aims to examine student perceptions related to the effectiveness and value of using Agile work practices to enhance group satisfaction in project-based courses. This study focused on two sections of an entry-level ICAgile accredited course. One section had 14 students, and the other section had 10 students. Each section was taught by ICAgile accredited faculty.

At the end of the course, all 24 students were asked to answer a voluntary survey concerning the use of Agile practices in the course. All 14 questions in the survey used a standard five-point Likert scale with a "1" signifying strong disagreement with a given statement and a "5" signifying strong agreement. The questions in the survey were divided into four different sections. The first four questions focused on how Agile influenced their learning, use of time, teamwork, project quality, and overall Agile experience. The next three questions focused on if the class project helped them to apply the Agile practices. The next four questions asked students if the Agile practices of prioritization, estimation, storyboards, and user stories influenced their team's productivity. The last three questions asked if Agile improved their group experience and if they found Agile beneficial. Only 21 of 24 students completed the survey. Two of the 21 students did not complete the back of the survey, and therefore did not answer questions 10-14.

4. SURVEY RESULTS AND DISCUSSION

Question		**Overall n=21**
Using Agile to complete the projects lead to a more efficient use of our time.	Mean Std. Dev.	4.65 0.93
Using Agile to complete the projects made us work better together as a team.	Mean Std. Dev.	4.65 0.81
Using Agile to complete the projects made the deliverables of higher quality.	Mean Std. Dev.	4.70 0.73
Using Agile to complete the projects allowed us to deliver the project in a timely manner.	Mean Std. Dev.	4.60 0.94

Table 1: Student perception of using Agile in a project-based course.

The data in Table 1 displays the survey's results for questions pertaining to student' perceptions of how using Agile lead to a more efficient use of time, helped them work better as a team, improved the quality of the deliverables, and helped them complete the deliverables on time. As the data indicates, the questions in Table 1 received the highest scores of the survey, indicating that students perceive using Agile practices in a project-based course helped them to manage their time better, work better as a team, develop deliverables of higher quality, and complete the project on time. The mean of the survey questions in Table 1 show a trend that students believe using Agile helps a team work better together, manage their time, and create higher quality outputs on time.

The next set of survey questions focused on how the specific Agile practices of retrospectives, daily stand-ups, and project charter added value to the project. As displayed in Table 2, Agile retrospectives and daily stand-ups received a higher mean than the project charter. The survey results in Table 2 indicate that students perceive Agile retrospectives and daily stand-ups helped them to work better as a team. This survey data also indicate that students perceive using Agile practices helps them to understand the value Agile brings to a team.

Question		Overall n=21
The projects helped me understand the value of performing retrospectives to improve how my team organized, completed work, and how we worked together as a team.	Mean Std. Dev.	4.50 1.00
The projects helped me understand the value of Agile daily stand-ups.	Mean Std. Dev.	4.45 0.94
The projects helped me understand the value of an Agile project charter.	Mean Std. Dev.	4.25 1.01

Table 2: Student perception of using specific Agile practices in a project-based course.

Agile retrospectives are a reflective practice where students reflect on their work every two weeks during the project. They discuss what is going well, what is not going well, and what needs changed, so they can improve how they work together as a team. Daily stand-ups are an Agile practice where a project team meets daily for 15 minutes to discuss what each team member did yesterday, is working on today, and what obstacles are keeping them from working. Stand-ups help a team to bring elements of transparency and accountability to their daily work. A project charter is a document created at the start of a new project that defines the vision or roadmap for the project.
The data in Table 3 display how students perceive the specific Agile practices of prioritization, estimation, storyboards, and user stories helped their teams to be more productive. The Agile prioritization method is MoSCoW, which uses the designations of Must have (M), Should have (S), Could have (C) and Won't have (W) to organize tasks in order of priority. The most important tasks or requirements (think rubric or a client's requirements) are marked with an M, since they "Must" be completed. After a team completes the MoSCoW method, they have all the work or requirements they for a project prioritized. A team can start with the tasks they marked with an M. The MoSCoW method allows the most important tasks to be completed first, which assists in the most valuable requirements being completed first.

An Agile storyboard is a tool that helps teams visualize and optimize how work gets done on a project. A storyboard can be virtual or physical, such as a whiteboard. A team posts all the requirements (called user stories) for a project on the storyboard so everyone can see the work that needs completed and the priority for each item. Team members assign themselves to the user stories on the storyboard so they can be accountable for their work.
User stories are an Agile practice used to represent the requirements for a project, and they are displayed virtually or physically on a storyboard. Story cards usually contain one or two sentences that describe the requirement, feature, or function for a project (LeanDog, 2012). A good practice is to have only one requirement per story card. This practice helps to divide up the work into manageable and incremental pieces.

According to the survey results, students ranked using the MoSCoW prioritization method and storyboards the highest of all the Agile practices surveyed. Since MoSCoW helps students prioritize tasks or requirements, the students perceived that prioritizing requirements needed for a project increased productivity levels of their teams. When the most important or valuable work is completed first, the project starts to take shape. In an Agile culture, this is called developing the minimal marketable product. The minimal marketable product contains the necessary requirements to be functionable. In the case of the students who participated in the survey, they were able to deliver an initial proposal for fundraising ideas to their client within a few weeks. Students are also encouraged when they can see the results of their work so soon in the semester.

Question		**Overall n=19**
Using Agile prioritization helped my team to be more productive.	Mean Std. Dev.	4.45 0.94
Using Agile estimation helped my team to be more productive.	Mean Std. Dev.	4.36 0.83
Using Agile storyboards helped my team to be more productive.	Mean Std. Dev.	4.41 0.79
Using Agile user stories helped my team to be more productive.	Mean Std. Dev.	4.23 0.97

Table 3: Student perception of using Agile practices in a project-based course.

The data in Table 4 reflect students' perceptions of how beneficial they found using Agile in a project-based course. Survey results indicate that students found great value in using Agile. All results in this section were statistically close in score. Students indicated that Agile made project/group work more productive. Students also ranked highly that using Agile made project/group work more enjoyable. These results possibly indicate the need to continue to research using Agile in group work and project-based courses. Students perceived in this research that using Agile improved their group work experiences. Teamwork or class projects can often be challenging for students and faculty, and students ranked that using Agile helped their group to be more productive and made the group experience more enjoyable. Agile should be considered as one option to approach group work and project-based courses.

The last question ranked the highest out of this section of survey questions and indicates that students see the value in using Agile and find it beneficial when working on a project.

Question		**Overall n=19**
Agile made project/group work more productive than not using Agile.	Mean Std. Dev.	4.58 1.00
Agile made project/group work more enjoyable than not using Agile.	Mean Std. Dev.	4.58 0.87
Overall, using Agile on the projects was a beneficial experience.	Mean Std. Dev.	4.64 0.99

Table 4: Overall student perception of using Agile in a project-based course.

5. CONCLUSION

A limitation in this study was the sample size. Future research could examine these survey questions across all three Agile courses in our Agile concentration to evaluate if student's learning and application of Agile has changed as they grow in their knowledge.

The results of this research are consistent with other research where students felt there was value in using Agile practices in courses and teamwork (Hulshult & Krehbiel, 2019; Woods & Hulshult, 2018; Pope-Ruark, Eichel, Talbott, & Thornton, 2011). Incorporating Agile practices into postsecondary courses should continue to be studied since the results are positive may add value to student's learning experience. Agile is a promising solution to further explore to help facilitate group work and projects in postsecondary courses. Faculty who have experienced difficulties with group projects may want to explore implementing some Agile practices such as user stories, storyboards, and MoSCoW prioritization into their courses.

6. REFERENCES

Beccaria, L., Kek, M., Huijser, H., Rose, J., & Kimmins, L. (2014). The interrelationships between student approaches to learning and group work. *Nurse Education Today*, *34*(7), 1094-1103.

CompTIA (2015). International technology adoption & workforce trends. Retrieved June 14, 2017, from https://www.comptia.org/resources/international-technology-adoption-workforce-trends

Frame, T. R., Cailor, S. M., Gryka, R. J., Chen, A. M., Kiersma, M. E., & Sheppard, L. (2015). Student perceptions of team-based learning vs traditional lecture-based learning. *American Journal of Pharmaceutical Education*, *79*(4), 1–11.

Hulshult, A. & Krehbiel, T. (2019). Using eight Agile practices in an online course to improve student learning and team project quality. *Journal of Higher Education Theory and Practice*, *19*(3), 55-67.

ICAgile, (n.d.). ICAgile learning roadmap overview. Retrieved on May 9, 2019 fromhttps://icagile.com/Learning-Roadmap/Roadmap-Overview

Krehbiel, T. C., Salzarulo, P. A., Cosmah, M. L., Forren, J., Gannod, G., Havelka, D., Hulshult, A. R., & Merhout, J. (2017). Agile manifesto for teaching and learning. *The Journal of Effective Teaching*, *17*(2), 90-111.

Kuh, G. D., Kinzie, J., Buckley, J. A., Bridges, B. K., & Hayek, J. C. (2006). What matters to student success: A review of the literature. Commissioned report for the national symposium on postsecondary student success: Spearheading a dialog on student success. National Postsecondary Educational Cooperative.

LeanDog (2012). Agile discussion guide, version 3.1. LeanDog, Inc.: Cleveland, OH.

LeBlanc, P.J., (2018). Higher education in a VUCA world. *Change*, *50*(3/4), 23–26.

Pope-Ruark, R., Eichel, M., Talbott, S., & Thornton, K. (2011). *Let's Scrum: How Scrum methodology encourages students to view themselves as collaborators*. Teaching and Learning Together in Higher Education. Retrieved from http://teachingandlearningtogether.blogs.brynmawr.edu/archivedissues/may-issue/lets-scrum.

Rico, D. F., & Sayani, H. H. (2009, August). Use of agile methods in software engineering education. In Proceedings Agile Conference, 2009. AGILE'09. (pp. 174–179). IEEE.

Smith, G. & Sidky (2009). Becoming Agile in an Imperfect World. Manning Publications Co.: Greenwich, CT.

Woods, D. & Hulshult, A. (2018). Using agile practices to scaffold team projects in an IT ethics course. *Journal of Computing Sciences in Colleges*, *34*(1), 17-23.

Transition to Amazon AWS from a Traditional Cluster-Based Information Technology Classroom

Vamsi Gondi
vkgondi@bsu.edu

David Hua
dhua@bsu.edu

Biju Raja Bajracharya
bajracharya@bsu.edu

CICS, CCIM,
Ball State University
Muncie IN

Abstract

In the age of COVID-19, traditional teaching techniques are under scrutiny. Most concepts and labs in the area of information technology are intended to be taught face-to-face in labs using clusters and networking infrastructure. Simulation software's such as uCertify, Pearson online, etc.., are good alternative, but are not always a viable solution. Simulation software is used primarily to understand, remember, and apply the concepts rather than analyze evaluate and create the content. Due to unforeseen conditions during March 2020, educational institutions were required to shut down. In the middle of the semester, instructors had to determine how to complete the labs and final projects which are crucial to accomplishing the intended learning outcomes of their courses. A chosen alternative was to use Amazon AWS as a cloud platform to host the labs and projects. It was initially a challenge as students had not had the opportunity to learn how to work with AWS in advance. Training sessions were provided to help them understand how to deploy and connect virtual servers in AWS to build a virtual private cloud. Prior to the pandemic, students developed and deployed their labs and projects on a departmental cluster which was on-premise at the university. With AWS, it was challenging to design an architecture to host 19 services offered as a part of the project requirements for an organization network. This paper presents the scope of this project; the intended outcomes of the project, how students were able to implement the project requirements in the AWS environment; the intended learning outcomes; and the results of student surveys to assess the learning outcome of using AWS and traditional cluster.

Keywords: Learning outcomes, Assessment, Project based learning, Blooms Taxonomy, Amazon AWS, Cloud Computing Cluster, Servers and services.

1. SCOPE OF THE COURSE AND FINAL PROJECT

CT 321 is an advanced server operating system course. The course is structured to learn 19 different Linux based services and concepts: Linux installation; modifying the Linux kernel; RAID configuration; logical volume management (LVM); network interface bonding; deployment and configuration of Dynamic Host Configuration Protocol (DHCP), Domain Name System (DNS), Secure Shell (SSH), Network Time Protocol (NTP) and Server Message Block (SMB) services, Apache server deployment and website configuration, OTRS server, Openfire server, MySQL, OpenVPN server, Nagios, Lightweight Directory Access Protocol (LDAP), and Fail2ban.

Students were to complete the lecture and labs for the first 12 weeks of the course. The final three weeks were dedicated to completing final project. Prior to the COVID-19 pandemic, the labs and final project were completed on the department's cloud cluster. During their prerequisite courses, students acquired the skills needed for the configuring and managing the cluster and network infrastructures.

Project Scope and Scenario

The students are divided into groups of three, and asked to work as a consulting firm. This consulting firm needed to design and deploy an open source data center solution for a fictitious corporation, CompTech, LLC. The project description stated that the company has slowly been growing in size since 2004. Starting with a mere 20 users and 2 IT staff, CompTech, LLC had grown to now serving 170+ users and 5 full-time IT staff. The company currently has a band-aided setup where the network core device is handling processes that ought to be managed by real servers. The company also had a storage cluster that recently failed and needs to be replaced with a newer more stable system. CompTech has been relying on mismatched and varying operating platforms for their needs and recently acquired new hardware to rebuild the data-center.

CompTech desires a new infrastructure and has enlisted outside assistance in accomplishing this task. The final result, to be effective, must be:

- Secure
- Quick to load
- Stable/reliable
- Administratively simplistic
- Well documented
- Cost effective

CompTech LLC, has 3 HP DL360 1u servers, 1 DL380 2u or 1 DL385 1u storage array already racked and ready for implementation. All servers are currently wired to network equipment as well as a Raritan KVM system to allow ease of administrative tasks. The servers are also connected to a monitored and switched power distribution system.

2. TASKS AND DELIVERABLES

The project is setup with multiple tasks:

Task 1 - Initial vendor tasks:

- Introduction of team members and consulting company name
- Collecting project information
- Identifying project implementation time-line

Task 2: Vendor testing stage tasks:

- Determining system capabilities
- Gathering configuration information
- Building test environments for services needed
- Integrating test environments to mimic real-world production environments
- Documenting all test system configurations, IP addressing, administrative information, and more.

Task 3: Implementation phase tasks:

- Implementing the proposed system on provided hardware as a proof of concept
- Testing the implementation of the systems
- Documenting all user names, passwords, configurations, settings, miscellaneous administrative information, etc.
- Preparing a formal report to provide CompTech at the end of implementation

Task 4: Vendor proposal and presentation tasks:

- Conducting a formal presentation for CompTech, LLC in professional dress. The presentation should include primary deliverables; focus on added value the project creates for the business while not being overly technical
- Providing CompTech, LLC with the proposal and findings from the test systems
- Answering questions from CompTech in regards to planned implementation

Task 5: Testing:

- CompTech, LLC should be able to use all systems to 100% of their proposed solution.

Deliverables:

Students were supposed to submit multiple deliverables during the course of the project.

- A written report about their group's chosen server platform
- Project management plan
- Specifications and requirements document
- Implementation document with screenshots and instructions on how to install, use and maintaining the servers and services
- Providing an oral presentation with slides

3. AWS TRANSITIONING

Due to COVID-19, the university was closed midway through the semester. The department cluster is isolated from the university's production infrastructure. Since the degree curriculum includes courses on wired networking, wireless networking, and cybersecurity which need to be delivered in our labs to prevent any disruptions on the university's production infrastructure. Due to this, VPN connectivity for the students to work on their projects in the department cluster is not available.

The closing of the campus forced faculty to find alternative platforms that were publicly accessible that would allow students to complete their labs and final project remotely. Many of the faculty were already experienced with the Amazon AWS (Amazon, 2020) cloud platform for research purposes. At the university level, faculty and administrators worked together to establish an institutional account with AWS Educate to provide students with no cost access to AWS.

To prepare students to work in the Amazon platform, students participated in workshops on the basics of AWS, how to access AWS and EC2 cloud. They were trained on how to create instances, connect to them remotely, transfer files and data, connect multiple instances, security configurations for port accessing, and managing the virtual private clouds. These workshops were conducted in parallel with regular online classes to ensure the course stayed on schedule.

For completing the project, students were asked to create and use EC2 instances instead of the department's physical cluster and assume the project is moved to AWS rather than in-house servers.

Major Issues:

Students faced several challenges due to AWS architecture when trying to deploy DNS servers, DHCP servers, security group configurations, and VPN to external clients. Most of the issues were resolved by students using the resources provided by AWS and researching online how to configure the services in AWS.

4. PROJECT OUTCOMES

This version of the project has been assigned for three semesters. Most of the students complete the project by deploying the server VMs and services on the departmental cluster. But the project goal is to design a system architecture which is resilient. The servers and services need to complement each other such as remote logging, RAID configuration on storage servers and server back-ups to the storage servers. This has proven to be difficult due to limited physical resources in the cluster.

The unplanned transition to AWS placed a steep learning curve on the students. Despite this requirement, four of the six student teams successfully accomplished all the tasks. The remaining two teams nearly completed the necessary tasks.

This project engaged students in learning activities that increased their subject knowledge expertise. The activities were designed in accordance with Bloom's Taxonomy to promote the understanding, application, analysis, evaluating and creating (Blooms Taxonomy, 2020).

The unplanned transition to AWS resulted in additional programmatic and student learning benefits. The outcomes have provided support for reducing programmatic dependencies on premise hardware. The low to no cost educational platforms offered by Amazon AWS and Microsoft Azure provide students with the opportunity to learn and create in an environment that more closely aligns with industry best practices. In turn, the enhanced student learning outcomes supports the need for on-going curriculum development that will keep the degree program relevant.

5. STUDENT SURVEY

After the completion of the course, a survey was given to participating students. The survey was to determine the relative benefits between using AWS or the department cluster. Of interest was whether students perceived either learning platform to be more effective accomplishing the intended student learning outcomes. Most of the students had little to no experience working with AWS. We also conducted this survey with the students who took this course in previous semesters to gage their outcome of the project.

Students who worked on the project in cluster and AWS felt the project was very complex and lot of time was invested to complete the project. Additionally, students who worked on AWS expressed even more complexity when integrating the project with AWS. On the other hand, students who completed the project in AWS felt they gained very good knowledge on AWS, how it works, and how to apply projects with AWS in the future. Also, most of the students expressed interest in working with both cluster-based and AWS combined, as they feel cluster can help on physical infrastructure management and AWS with cloud based.

Most of the students expressed interest in working with both traditional cluster and cloud-based environments. They also think the knowledge of cloud-based environments such as AWS will significantly boost their employability.

6. CONCLUSIONS

COVID-19 created an opportunity to explore another teaching tool that can be used in the classroom. Transitioning was very challenging, especially when we are dealing with the students. As a faculty, we had to prepare ourselves for every difficult scenario. This was one of those occasion we went beyond the calling. The experience of using AWS as a teaching and learning platform has encouraged faculty to reconsider their course design and deliver models. The availability of data storage and disaster recovery, using AWS and S3 storage containers allows faculty to create more robust assignments that more accurately reflect the work environments they will encounter upon entering their professional careers. We feel AWS with traditional cluster knowledge will boost student's capabilities as this project demonstrates the same.

9. REFERENCES

Amazon (2020). Amazon AWS Educate, Retrieved from https://aws.amazon.com/education/awseducate/ (On August 20, 2020)

Bloom's Taxonomy (2020). Revised Bloom's Taxonomy Retrieved from https://www.celt.iastate.edu/teaching/effective-teaching-practices/revised-blooms-taxonomy/ (on August 20, 2020)

Survey Questionnaire

1. Have you ever have experience in working in cloud environment (rate 1 to 5, 1 for low and 5 for high)
2. Which cloud environment do you have experience with
   a. Amazon AWS b. Microsoft azure c. IBM Bluemix d. others (please specify) e. none
3. At what level of experience do you have working with Amazon AWS (rate 1 to 5, 1 for low and 5 for high) before the starting of the project
4. Did you worked on the CT321 project using AWS
   a. Yes b. No
5. Did you accomplished all the tasks mentioned in the project description
   a. Yes b. No c. Somewhat (AWS)
   b. Yes b. No c. somewhat (Non AWS)
6. What level of difficulty did you face while working on the project environment (rate 1 to 5, 1 for low and 5 for 5)
   a. AWS
   b. Non AWS
7. At what level of experience do you have working with Amazon AWS (rate 1 to 5, 1 for low and 5 for high) after completion of the project
8. Have you ever completed any project with traditional cluster at BSU
   a. Yes b. No
9. Which one do you think more effective in completing the project
   a. Traditional b. Cloud based (AWS)
10. Do you think you gained experience on working AWS (rate 1 to 5, 1 for low and 5 for high)
11. How do you rate this project completing in cluster(rate 1 to 5, 1 for low and 5 for high)
12. How do you rate this project completing in AWS (rate 1 to 5, 1 for low and 5 for high)
13. In future which one do you prefer
    a. AWS b. Cluster at BSU c. none of them d. Other (Please specify)
14. Do you think learning AWS will boast your career when you enter job market
    a. Yes b. No
15. Rate this project complexity (rate 1 to 5, 1 for low and 5 for high)

Business Students as Citizen Developers: Assessing Technological Self-Conception and Readiness

Donald Thacker
dthacker@kent.edu
Department of Marketing & Entrepreneurship

Victor Berardi
vberardi@kent.edu

Vaneet Kaur
vkaur@kent.edu

Greg Blundell
gblundel@kent.edu

Department of Mangement & Information Systems
Kent State University at Stark
North Canton, Ohio 44720, USA

Abstract

Understanding students' perceptions, aptitudes, and interest in using technology, and in developing technological solutions to problems, is important for effectively incorporating it into course pedagogy. This paper presents an effort to understand student technological self-conception, especially regarding the use-of-technology versus the development-of-technology self-perception schema. Concern exists that because students are so comfortable with using technology, they are being lulled into a false sense of security and may be at risk of being replaced by it if they lack the ability to be creators of technological processes and tools themselves. We seek to broaden student conception of themselves to consider becoming creators, thereby beginning the journey towards becoming citizen developers. A survey designed with consideration of the Theory of Planned Behavior is used to understand student pre-disposition and self-conception pertaining to technology use versus development. Inspiration is also drawn from design thinking, an approach that promotes innovative development as a means of combining technology with the human element to create viable, effective solutions. The intended outcome of the overall project is to initiate the creation of citizen developers, within existing curriculum and course requirements we are bound to honor, thereby serving as a bridge towards a more holistic design thinking in students, sparking a creative process of solution and problem-solving generation and development. This first stage of the research project is focused on developing an understanding of student technology comprehension and their prior experience in not just utilizing technology but also in leveraging it to develop solutions to meaningful problems.

Keywords: citizen developer, low-code no-code, Design Thinking, Theory of Planned Behavior, workflow automation, Digital Natives, Generation Z

1. INTRODUCTION

Generation Z students are beginning to dominate the enrollment in college-level programs. These students are widely regarded as *Digital Natives*, emphasizing that they were born and raised with computerization and ubiquitous personal devices during the internet era. As educators, understanding our students' self-conception, aptitude, and interest in using technology—and in developing technological solutions to problems—is important for effectively incorporating technology into course pedagogy and across the academic curriculum more generally. This paper presents an effort to understand student perception of technology, especially regarding the use-of-technology versus the development-of-technology self-perception schema. While anecdotally it seems that students are comfortable with using smart phones and devices in their personal and work lives; less often, do students express interest in computer programming, producing apps and other development activities. Having insight into students' self-conception in this regard might provide actionable information that can be fashioned into academic interventions that spark their interest to learn and do more with technology.

A planned behavior inspired survey was developed to gather information on these items and others. After gaining preliminary insight into students via the survey, we will craft workflow automation (WFA) learning activities, implement them, and then query student participants post-activity to gauge any change in their perspective. In this effort, the concepts of Design Thinking and design education pedagogy are aligned nicely with our long-term endeavors and provide insights for our work. The pre- and post-activity surveys go beyond WFA and include forward-looking questions to gauge student interest in creating apps for others to use and even to create intelligent agents incorporated into cross-functional and cross-system solutions. The results presented in this manuscript include discussion and analysis of the pre-activity survey results with a range of possible future endeavors and research ideas identified and discussed.

2. TECHNOLOGICAL SELF-PERCEPTION

With students as Digital Natives, it would seem their comfort with computerized systems and devices might afford advantages in solving problems using technology. However, the digital native narrative belies a narrow definition of technology that may not fully align with practical demands. Using an app on a phone or other device can have value, of course. However, anecdotally it seems that student focus on technology is too concentrated on devices—like smartphones, watches, or tablets—and not enough on techniques, processes, and application software-based tools that proliferate in practice. In the book, The Culture of Technology, Pacey (1983) defines technology as "the application of scientific and other knowledge to practical tasks by … ordered systems that involve people and organizations, living things and machines" (p. 6). This definition points to the broad view of technology within society and emphasizes the interaction between humans and machines or processes. It is important to note that in consequential, and fundamental ways, the knowledge going into process design and the mechanics or techniques of task completion are important, though often under-appreciated aspects of technology.

Most Generation Z students have always possessed technology at their fingertips. They often seem, however, to lack a general conscientiousness or appreciation of innovation, problem solving, computer programming and technology development. Barak and Levenberg (2016), suggest that the issue is a resistance to change due to inflexible thinking. According to their studies, the individual learner has seemingly developed considerable fixed habits and patterns of thought, creating a resistance to change and reduced flexibility. In this case, they "don't know, what they don't know" and tend to be reluctant to search for alternative or better technology and solutions.

A concern exists that students have been lulled into a false sense of security and ease regarding technology. Because they are so comfortable with its use, students may be at risk of becoming replaced by such technology if they lack the ability to be a creator of technological processes and tools; developers that generate innovative solutions leading to efficient and positive outcomes for their organizations and society. This aligns with what might be called "citizen developers." Gartner defines a citizen developer as "a user who creates new business applications for consumption by others using development and runtime environments sanctioned by corporate IT" (Citizen Developer, n.d. para. 1). We seek to broaden student conception of themselves to consider becoming involved in creating the technological processes themselves, thereby, beginning the journey towards becoming citizen developers.

In a pre-activity survey, students are asked to self-report their technology use in their personal and work lives. Figure 1 presents the top-ten-word frequency responses regarding technology usage in students' personal (top panel) and work lives (bottom panel). Overall, from the personal perspective, students volunteered 324 responses, with another 206 from a professional view.

Personal Technology Use

Rank	Term	Count	Percent	Cumulative
1	cellphone	97	29.9%	29.9%
2	laptop	32	9.9%	39.8%
3	Excel	18	5.6%	45.4%
4	Word	14	4.3%	49.7%
5	iPhone	12	3.7%	53.4%
6	TV	9	2.8%	56.2%
7	MacBook	8	2.5%	58.6%
8	gaming systems	7	2.2%	60.8%
9	iPad	7	2.2%	63.0%
10	PowerPoint	7	2.2%	65.1%

Professional Technology Use

Rank	Term	Count	Percent	Cumulative
1	computer	36	17.5%	17.5%
2	Excel	17	8.3%	25.7%
3	cellphone	14	6.8%	32.5%
4	Word	10	4.9%	37.4%
5	POS system	7	3.4%	40.8%
6	none	6	2.9%	43.7%
7	desktop	5	2.4%	46.1%
8	iPad	5	2.4%	48.5%
9	laptop	4	1.9%	50.5%
10	QuickBooks	4	1.9%	52.4%

Figure 1. Word frequency analysis of self-reported technology used in personal (top) and organizational (bottom) settings.

Both lists are dominated by devices. From a personal perspective, cellphones (including iPhone) laptops (including MacBooks), iPads, gaming systems, and TV represent more than half the responses. The remainder of the top-ten personal uses include office applications and cumulatively represents two-thirds of all responses. The professional use perspective differs some, with Point-of-Sale (POS) systems and accounting software (e.g. QuickBooks) appearing on the list. Neither list contains any programming languages or other technology development tools or platforms.

From Figure 1, it might be inferred that students are comfortable with their own aptitude and usage of technology; however, this could be providing false comfort. The word frequency results also hint at a lack of curiosity in students regarding harnessing or developing said technology, which is concerning to us as educators but also signals a meaningful learning opportunity.

Given that environmental and societal definitions of technology are very broad, whereas student conception of technology appears less so, we seek to encourage a more holistic conception of technology to be internalized by students. We hope to have students who see developing new processes, techniques, and ways of doing things—especially with computerized tools or approaches—as a technology mastery worth investigating. Considering the self-reported student views related to technology development in Figure 1, the indication and opportunity to have a meaningful educational intervention does seem promising. Not only do students need the tools, interest, and confidence to accomplish the role of the developer, in addition, they need the encouragement and freedom to explore solutions and possibilities on their own, which is at the heart of design thinking.

3. DESIGN THINKING

Design thinking is an approach that promotes innovative development as a means of combining technology with the human element to create viable, effective solutions. Design Thinking in Education (n.d.), is described as "a mindset and approach to learning, collaboration, and problem solving," (para. 1), that in practice, "is a structured framework for identifying challenges, gathering information, generating potential solutions, refining ideas, and testing solutions," (para. 1), that "can be flexibly implemented; serving equally well as a framework for a course design or a roadmap for an activity or group project" (para. 1).

The design thinking approach can be leveraged for our purposes, by employing trigger-oriented processes of low-code/no-code programming tools, such as Microsoft Power Automate, Zapier, and IFTTT. Harnessing the power of design thinking enables individuals to foster the ability to produce innovative procedures and then iterate and expand them easily. A sense of empowerment provides the student with the ability to become a citizen developer and an innovative contributor to their organization.

Gartner Inc. notes that end users are creating "new, more powerful applications" (Yanckello and Calhoun Williams, 2019 p. 45), destined to forever change who is considered a "programmer." And this reality is closer than many may realize. The Gartner Priority Matrix for Education (Yanckello and Calhoun Williams, 2019), projects that citizen developers are within two years of meaningful impact on the marketplace, and further predicts design thinking will become ingrained at all levels of organizational activities and problem-solving efforts. The intended outcome of our project is to initiate the creation of citizen developers, within existing curriculum and course requirements we are bound to honor; to serve as a bridge towards a more holistic design thinking in students, sparking a creative process of solution and problem-solving generation and development.

Design thinking and pedagogy has a long and rich foundation. Oxman (2006), claims as "conceptual changes become the content of new pedagogical methods of design education, [the] awareness of change and conflicts can stimulate the necessary theorization and conceptualization for new approaches to design didactics" (p. 45). The author reminds us that "[d]esign thinking precedes design learning," and although recent evolutions in this method have generated new paradigms, they are filled with "conceptual conflicts between the prevailing and the new values of two design ontologies" (p. 45). Oxman concludes that these new "pedagogies can operate within this condition of the evolution and instability of ontologies [but] can do so only by directly articulating and working with conceptual structures as pedagogical material" (p. 45). In summary, Oxman (2006) emphasizes the need to properly craft the intervention and its implementation—and to communicate what is being asked of students and why—so students can not only see the value in the activity itself but may appreciate the logic, motivation and exhortations behind it, then to make it their own to solve problems of interest to them.

Luka (2014) reminds us "[i]nnovation drives improvement, either incrementally idolizing existing processes or more radically by introducing new practices" (p. 72) and reiterates what other authors have claimed that to increase student innovation is through developing design thinking skills. Luka (2014) concludes claiming "[s]tudents practice during their studies learn to make their own mistakes and realize that there are no right or wrong solutions to various problems [and] learn to explain their options and listen to others' opinions, accept untraditional ideas thus welcoming innovation" (p. 73). Wrigley & Straker (2015) are adamant that new pedagogical approaches must be introduced into higher education to adequately equip students with both the hard and soft skills that organizations prize in order to stay pace with changes in local and global trading environments. In addition, their Educational Design Ladder "provides a scaffold for organising and structuring Design Thinking units or courses in multidisciplinary contexts" (p. 11). The emergence of the low-code/no-code platforms provides an important scaffold for integrating these activities into a non-technical program such as business management and marketing, that simply did not exist a few years ago.

Vander Ark (2017) describes this methodology in its application to the world of work, as "a human-centered approach to innovation that draws from the designer's toolkit to integrate the needs of people, the possibilities of technology, and the requirements for business success" (para. 1) and concludes with interpreting the needs of all stakeholders while exercising continuous problem solving and employing inquiry-based learning that builds "character strengths, mindsets and dispositions [where] deeper learning activities including design thinking investigations are a great way to develop these new priority outcomes" (Design Thinking For EdLeaders, para. 3). So, design thinking and pedagogy have great promise of collective impact, although they both also have drawbacks or issues of concern, in that students must be properly prepared to learn in a design thinking paradigm and cannot just be thrown in and expected to thrive. This emphasizes Oxman's (2006) exhortation that students must understand what they are being asked to do and why. In short, students, the technology, and the environment, both academic and business, all must be appropriately "ready" for success to be possible.

Schell (n.d.) considers this pedagogy problem as "wicked," reminding educators that both teaching and learning this methodology, such that they result in lasting impacts "requires slowing down the learning, taking time to unfold the layers of what it means to be human-centered and to pay attention to the innate dignity of human beings" (Design Thinking's Pedagogy Problem, para. 2), and "spending focused energy practicing and receiving feedback from experts" (Design Thinking's Pedagogy Problem, para. 2).

Nonetheless, Schell (n.d.) offers a solution: first, cultivate self-regulated students of the methodology, and second, build a pedagogy to enhance their self-efficacy. Schell (n.d.), concludes to overcome the wicked problem, i.e., "the demand and authentic human need for accelerated design thinking pedagogy when the efficacious teaching of design thinking demands a decelerated model" (Conclusion, para. 1), is to avoid "accelerated design thinking education outside of academia" (Conclusion, para. 1) in favor of employing and advancing best practices with embedded options for students to self-regulate their learning and build their self-efficacy. In our efforts, we seek not to overwhelm students with unstructured and complex problems to solve to start with, but to ease students into this proposition, with articulated experiences within the curriculum we are bound to honor.

4. CURRICULAR CONSIDERATIONS

Our business program has 500 students taking classes where the plurality of majors is business management (50%) and marketing (20%), with accounting, economics, finance, CIS, and entrepreneurship making up the rest. The curriculum at our institution is in no means devoid of technology as faculty members have made purposeful choices to incorporate technology into their courses, even though it is not required, so that developing student technology proficiencies are not just a "one-and-done" mentality but, rather, reinforced throughout the program, particularly with office applications like Excel, but also Qualtrics and SPSS, amongst others.

It must be emphasized, that we have a duty to maintain fidelity to each Course Data Sheet, which is what defines the course coverage requirements and options at our institution. Through faculty choices, and within curriculum limits, though, we strive to ensure that students have a true appreciation and understanding of a plethora of technology options, thereby encouraging students to develop their own robust problem-solving developer's toolbox. So, at least in this regard, faculty choices to integrate technology even when not required, provides some of the foundation that Schell (n.d.) and Oxman (2006) emphasize is needed before design thinking can be a beneficial pedagogical tool. Low-code/no-code tools that are now becoming available, represent an important scaffold that simply was not available to us before and which provides new opportunity for non-technical academic programs to encourage such development in their students.

Pope-Ruark (2020) outlines that the new role of higher education is for institutions "to offer more options to achieve the master credential of a degree" (para. 7), and faculty "to help students chart a meaningful course through an intentionally selected variety of learning experiences, traditional and nontraditional, while helping them make meaningful connections that inform their choices about future experiences, careers, and roles as citizens" (para. 7). Our effort in this project is attempting to do just what Pope-Ruark (2020), exhorts. While we cannot change our program unilaterally, through academic freedom, we can certainly change what we do and how we do it to honor the spirit of Pope-Ruark's call to action.

Through these WFA activities, we seek to expose student self-perception to where they first recognize that low-code/no-code tools exist and then that these can be valuable tools in their current and future personal and professional lives. If this can be achieved, then we can seek to help students develop a new schema where they see in themselves—and have a measure of confidence in themselves—as actual developers who can indeed leverage low-code tools for personal and professional benefit.

5. PROJECT DESCRIPTION

This stage of the research project is focused on developing an understanding of student technology comprehension and their prior experience in not just utilizing technology but in leveraging it to develop solutions to meaningful problems. Considering that we are looking to students as potential citizen developers, it is important to identify the technological skill and experience levels of students, along with their extant perceptions. We are interested in understanding student participants' existing ability and interest in computer programming and developing automated processes utilizing computerized technologies. Beyond measurement of current applications, future intentions towards using additional technological applications and their confidence in doing so are also of interest.

Survey Instrument

A survey was created as an instrument to collect data and then to evaluate the comprehension, technological abilities, and perceived value of WFA within student's personal and professional lives. The student pre-activity survey consisted of 20 questions: 6 – Belief Scales, 3 – Attitude Scales, 8 – Behavior Scales and 3 – Behavioral /Intention Scales. Subsets of questions can be divided as technological competency, usage, experience, process development, and solution application of technology (See Appendix 1). One concern in developing the survey was that most of our students might not have any truly significant knowledge or understanding of WFA and/or low-code/no-code tools, so the survey itself would have to convey some foundational information while attempting to favorably influence perceptions and future behaviors.

Therefore, the survey was designed with consideration of the Theory of Planned Behavior (Ajzen, 2019). Behind human behavior are pre-dispositions and attitudes about what is believed. To change the behavior and on-going beliefs, one must build a beginning understanding, and then through progressions, change the beliefs that would support the desired on-going behavior. The survey questions represented a planned progression beginning with definition, self-report, and underlying beliefs. Through progressions like this, the theory posits that attitudes and beliefs can be softened or prepared for desired modification. Questions were arranged to expose student normative beliefs as well as behavioral beliefs towards technology and application development. Once existing beliefs were revealed, attitudes towards changing these existing beliefs were then queried. Finally, introduction of perceived behavior and reflection outcomes and intention to change the belief towards citizen development was measured. The underlying motivation was to induce participants to self-report their background knowledge on WFA, along with their comfort, confidence, and willingness to utilize technology as a practical and useful means.

The survey proffered low-code/no-code tools as a possible means to accomplish routine tasks through the creation and application of automated workflow processes in both the professional capacity and for their personal concerns. While a no-code/low-code approach and tools lowers the technical skill and cognitive load for users, there are also barriers related to critical thinking and problem-solving awareness more generally that we felt were important to understand when crafting effective activities. In other words, we needed to understand where students were in these regards so we could craft activities to effectively reach our audience.

Implementation Approach

The pre-activity survey evaluation began with voluntary participation sought from students enrolled in several courses that already had significant applied computer components. These courses included a basic computer applications course, an intermediate computer class focused on using information systems to solve business problems, an operations management course, and a project management course.

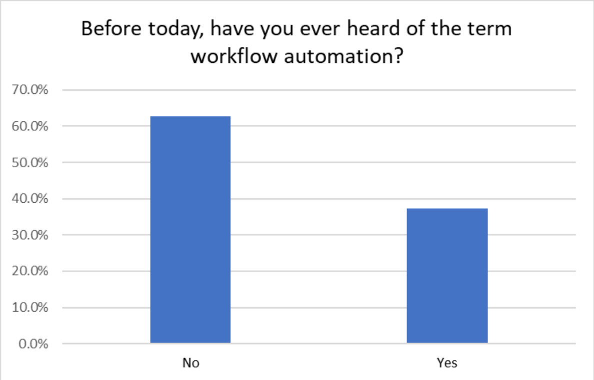
The introduction of the survey was performed early in the semester, with a preliminary review performed as a guide to creating the WFA activities. At the conclusion of the semester and activities, a thorough evaluation of the pre-activity survey results was performed, which is the focus of the remainder of this manuscript. Separate, forthcoming works will examine the efficacy the intervention and lessons learned from the endeavor overall.

6. RESULTS AND DISCUSSION

The pre-activity survey was distributed students in the classes noted above, generating 105 complete survey responses (n=105). All participants were undergraduate students of an AACSB business program at a regional campus of a major university. This represents approximately 20% of the total business student population for this regional campus with roughly 60% of students are upper division with the remainder being freshman or sophomores.

Before delving into the detailed results, note that differences between upper and lower division students was considered to determine if they differ significantly and should be treated as different populations or not. Appendix 2, has a graphical and statistical summary of eleven survey questions. These questions include aspects regarding student knowledge of and interest in WFA and low-code/no-code tools, their current use of such tools, their technological competency and confidence, and the value they ascribe to such tools. Using a graphical analysis, the upper and lower division students look very similar while from a statistical significance perspective, at a significance level of 0.10, none of the questions appear to differ significantly between upper- and lower-division students. Given these findings, the remainder of the results are analyzed from a single population perspective.

A key outcome of the pre-activity survey indicated that at the outset, most student participants had little knowledge or appreciation of WFA and limited knowledge of related technologies. In the top panel of Figure 2, it is seen that a clear majority of students (62.7%) had never heard of the term WFA before taking the pre-survey with a minority (37.3%) affirming to have heard of it. However, once introduced and informed about workflow automated processes, through the completion of the survey itself—which later in the survey included describing specific WFA applications—students are in near total agreement (91.2%) that WFA holds promise and could be of at least somewhat value. Indeed, the top panel of Figure 2 shows that more than two-thirds of students, 70.6 percent, thought WFA could be extremely valuable or valuable. It is encouraging to see that upon learning something about WFA, students recognize WFA as valuable, which suggests that they could be interested and willing to explore automation in their personal lifestyles, business interests, and academic endeavors too. In the WFA activities themselves, then, emphasis will be placed on creating cogent, relevant examples for students to complete.



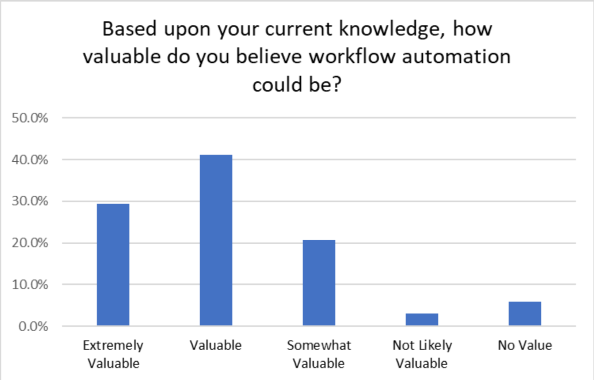
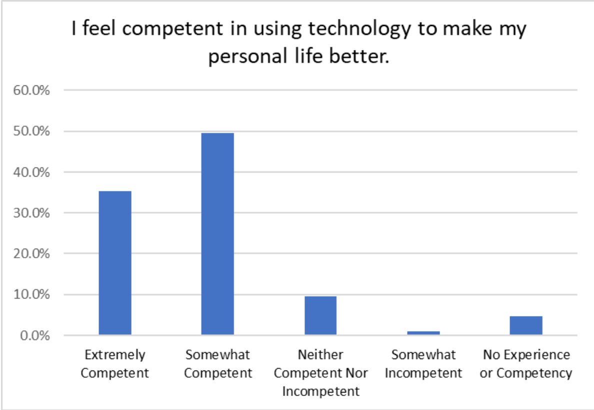


Figure 2. Student knowledge of WFA before the pre-activity survey (top) and their perception of WFA value following completion of pre-activity survey (bottom).

Not surprisingly, students expressed a general sense of comfort and competency in using technology. As seen in Figure 3, about 85 percent of students expressed favorable perceptions of their technology competency (extremely or somewhat competent) in both the personal and organizational settings; however, remember from Figure 1 that the forms of technology used by students were predominantly general hardware and software applications. There is a definite skewedness towards *utilization* of packaged and subscription software/applications by students.



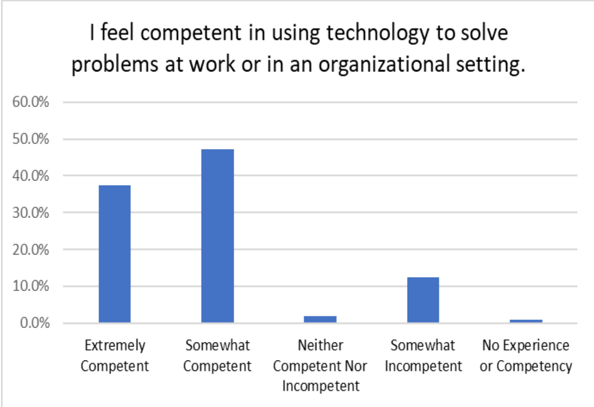


Figure 3. Student self-reported competency in using technology in personal (top) and work settings (bottom).

In contrast to being comfortable with using technology, there is a seemingly lack of interest, if not a trepidation, by students towards being a creator or developer of technological processes and tools. As seen in Figure 4, at this pre-activity point in time, there appears to be little student interest in creating automations for use by others through the development of apps or intelligent agents. Only one-in-five (19.8%) responses expressed interest or prior consideration of creating intelligent agents with two-thirds (65.1%) not interested. Students seem more interested in creating apps, as 37.4 percent have considered or are interested, while roughly half (46.1%) are simply not interested. In addition, no students reported having already created an app or intelligent agent. It should be noted that in the survey only eight students indicated any programming experience, with Java and Java Script the most common, followed by C#/Visual Basic/VBA, with one mention each for Python and PHP.

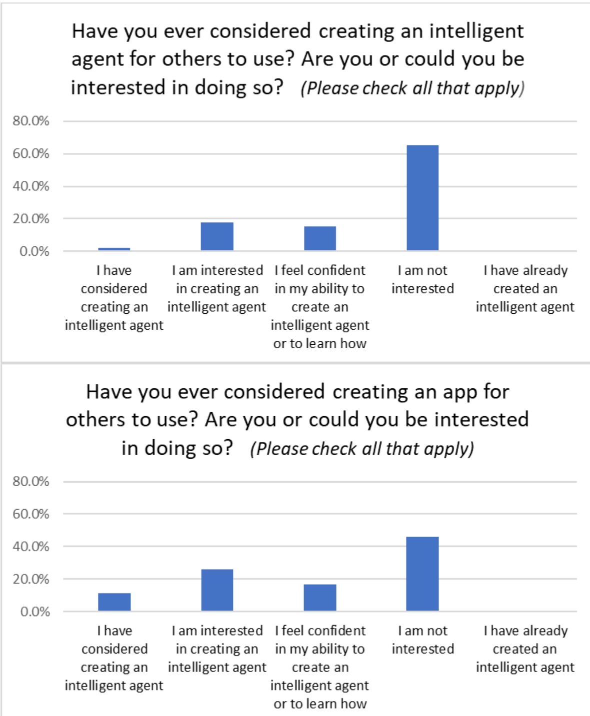


Figure 4. Student self-reported pre-activity interest in creating apps (top) or intelligent agents (bottom).

From these results, it seems there is little indication of a conscientious, individual technology development conception by students beyond that of being a user. In general, the survey results seem to signal a sense of confidence, if not overconfidence, by students that might result from their genuine comfort in using technology. But there seems to be a much more limited conception of the technology itself and how to harness technology rather than just being a user of it.

Digging deeper, Figure 5 shows that students do have some technological familiarity with a variety of technology tools. Results indicate that the main types of technological tools harnessed by students were Google Forms, WFA (including trigger automation, Zapier and IFTTT) and Smartsheets. It should be noted that tools, such as office applications that are already taught in our program, were not included as a response choice in this survey question and no students indicated other tools than those listed.

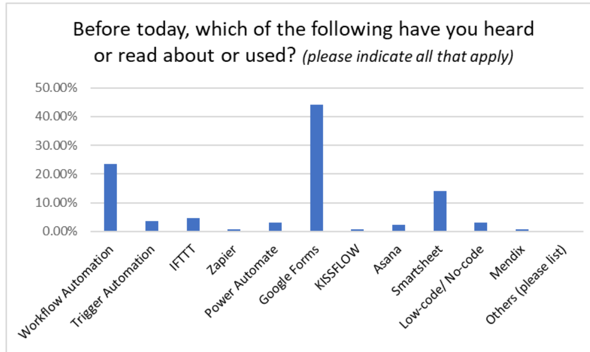


Figure 5. Self-reported use or knowledge of technology applications.

Of the minority of students that reported having used WFA in the past, Figure 6 shows the tasks they automated (top panel) and the areas they employed them in (bottom panel), be it personal, business, entrepreneurial, academic, etc. It is seen that for the students who have used WFA, they span the gamut of tasks and application areas.

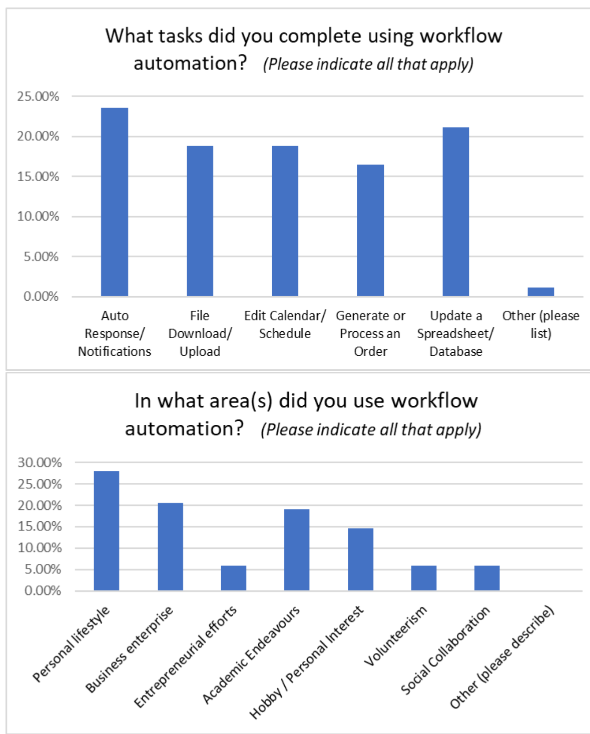


Figure 6. Student experience in creating WFAs: tasks completed (top) and areas where WFA was used (bottom).

Gauging the future intentions of students toward automated workflow development was an important aspect of our preparations, especially given that most students had never created or used WFAs or even heard of it. An aim of our planned WFA activities is to bring awareness to areas of WFA, and development of automated processes more generally, and these pre-activity survey responses indicated that a more fundamental, articulated approach to doing this, rather than an advanced one, is reasonable. At the same time, emphasizing the applicability of these tools to many tasks and areas to educate and inspire them, will be important.

Figure 7, meanwhile, shows that by the end of the survey, nearly nine-of-ten students (89.2%) agree (completely, strongly, or somewhat) that they would like to learn more about creating WFAs. In some small way, then, it seems some student conceptual evolution—or a schema shift—is being initiated through completing the survey itself: very few students even knew of WFA to start, but upon learning something about it, the vast majority believe it could be valuable and want to learn more about how to do so.

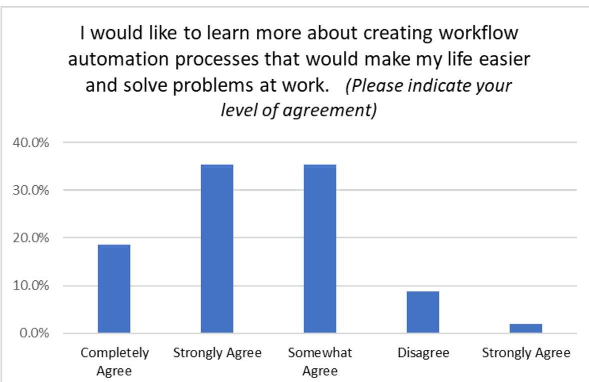


Figure 7. Self-reported interest in learning more about creating WFA processes.

In addition to questions about WFA, the survey separated out and defined what cross-functional and cross-system developments, applications and intelligent agents were. This aspect to the survey intended to alert, if not inform, students that the WFA development being discussed was not just referring to isolated, individual-only development. Indeed, we wanted students to realize that such developments had potential impact across organizational functions and in tying together disparate computer systems. Furthermore, we wanted students to realize that automations are often incorporated into apps for others to use or are components of intelligent agents, even if they did not fully understand or appreciate what that meant to begin with.

In Figure 8, less than one-fourth (22.5%) of students strongly or completely believe they can create useful cross-functional or cross-system automations with 77.5 percent of students less sure or even not believing they currently can do so.

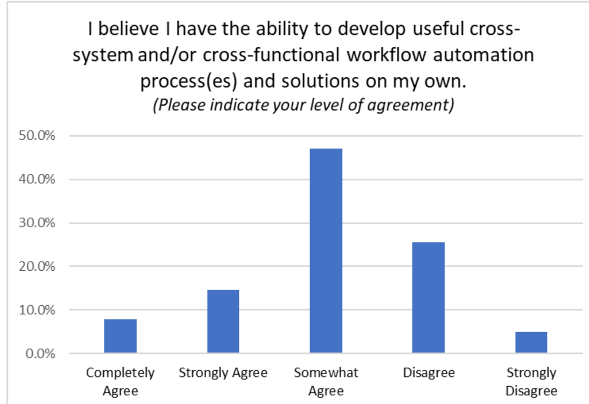


Figure 8. Student self-reported ability to develop cross-system or cross-functional WFAs.

In Figure 9, students signal a willingness to learn more about creating apps and intelligent agents (top panel) along with cross-system and cross-functional development (bottom panel). Nearly three-fourths of respondents (71.6%) are at least somewhat interested in creating apps or intelligent agents while 86.3 percent are interested in learning about cross-functional and cross-system development more generally. This is indeed encouraging.

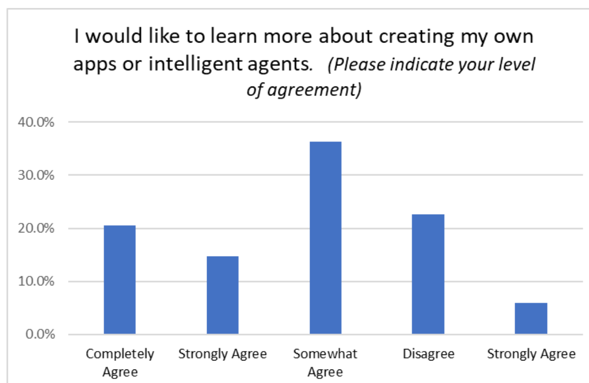


Figure 9. Self-reported interest in learning more about how to create apps or intelligent agents (top) and cross-system and cross-functional design (bottom).

7. CONCLUSIONS

In summary, the most definitive discovery of the survey stems from the recognition that students initially do have a self-perception of themselves as users of technology rather than developers of technology.

At the start of the survey, students do not see themselves as developers or programmers in any meaningful sense and might even be characterized as initially having a general disinterest in becoming one. It is unclear whether this has resulted from a lack of knowledge, a fear of technological development, a lack of motivation, or something else. By survey's end, during which they are informed about WFA, cross-functional and cross-system development, intelligent agents and app development, students show interest in learning more about all these items. At the same time, though, students are not confident in their ability in these areas.

Overall, we find this student feedback enlightening and promising. The results not only suggest students would value learning how to become developers of technology but also demonstrates to us a tremendous opportunity to develop pedagogical approaches towards these ends. Given student hesitancy and lack of prior interest or experience in development, the activities will include learning support scaffolds and designed to be highly relatable to their personal and professional experiences.

Generation Z students have always had consumer electronics at their fingertips, and it is easy to mistake this familiarity of use with having an actual command over technology. Utilizing the Theory of Planned Behavior, as well as a design thinking approach, we have an opportunity to positively affect student development and skill building. Low-code/no-code tools provide a framework for truly harnessing this technology so our students can ride the wave of technology rather than being overwhelmed by it. The survey reported within this manuscript was a way to test our ideas and to use that learning as we consider pedagogical interventions and activities for driving students towards becoming citizen developers.

8. FUTURE ENDEAVORS

After a preliminary analysis of the survey results, classroom activities were developed relating to common business scenarios using the WFA tool, Zapier. We chose to limit these activities to two scenarios: a personal use example, and a business application, while also highlighting many other applications for students. Each step in the process will be supported through not just written instructions and screenshots of the Zapier system but with step-by-step video support and implemented via a learning management system page that is self-explanatory in how to proceed, including a checklist to track progress.

As the citizen development mindset is being achieved, or really integrated into the program, design thinking can be reinforced via numerous small, frequent, and meaningful activities and assignments. Our plan is to incorporate these introductory WFA activities into our computer applications course going forward, so all our students have this foundation. Next, an upper-division course focusing on using information systems to solve business problems will implement more advanced WFAs including conditionality and multi-step processes. Applications for courses in operations management, supply chain management, marketing, and human resources are also being conceived. Then, we seek to leverage such WFA and low-code/no-code capabilities in promoting our students to businesses for internships, service-learning projects, and permanent positions.

The key is to create adaptive learning approaches to generate a new awareness and an integration of design thinking and citizen development into everyday practice, so our graduates can be successful solution architects in whatever direction the future takes them. This should help address the concerns of Schell (n.d.) and Oxman (2006), who advocated against just throwing students into a design or development pedagogy and overwhelming or frustrating them as a result. In doing so, we seek to meet the goal articulated by Pope-Ruark (2020) in providing students with "an intentionally selected variety of learning experiences, traditional and nontraditional" (para. 7) that helps them as they consider future careers and experiences.

9. REFERENCES

Ajzen, I. (2019). Constructing a theory of planned behavior questionnaire. Retrieved from https //people.umass.edu/aizen/pdf/tpb.measurement.pdf

Barak, M., & Levenberg, A. (2016b). Flexible thinking in learning: An individual differences measure for learning in technology-enhanced environments. *Computers & Education, 99*, 39–52. http://dx.doi.org/10.1016/j.compedu.2016.04.003

Citizen Developer (n.d.). In *Gartner Glossary*. Retrieved from https://www.gartner.com/en/information-technology/glossary/citizen-developer

Design Thinking in Education (n.d.). Retrieved from https://tll.gse.harvard.edu/design-thinking

Luka, I. (2014). Design Thinking in Pedagogy. Journal of Education Culture and Society*, 2*, 63-74. DOI: 10.15503/jecs20142.63.74

Oxman, R. (2006). 'Digital Design Thinking': In The New Design Is The New Pedagogy. in A. Kaga and R. Naka (editors), Proceedings of the 11th Conference on Computer-Aided Architectural Design Research in Asia (CAADRIA), Faculty of Engineering, Kumamoto University, Kumamoto, Japan, 37-46.

Pacey, A. (1983). *The Culture of Technology*. Cambridge, MA; MIT Press.

Pope-Ruark, R. (2020). Liberal Education Needs Integration, Not Unbundling. *Academe.* Retrieved from https://www.aaup.org/article/liberal-education-needs-integration-not-unbundling

Rymer, J. (2018, August 8). Why You Need To Know About Low-Code, Even If You're Not Responsible For Software Delivery. *Forrester.* Retrieved from https://go.forrester.com/blogs/why-you-need-to-know-about-low-code-even-if-youre-not-responsible-for-software-delivery/

Schell, J. (n.d.). Design Thinking Has a Pedagogy Problem … And a Way Forward. Retrieved from https://designcreativetech.utexas.edu/design-thinking-has-pedagogy-problem-way-forward

Shepard, L. (2000). The Role of Assessment in a Learning Culture. *Educational Researcher, 29*(7), 4-14.

Vander Ark, T. (2017, February 15). Design Thinking as Pedagogy – For Students and Educators. *Education Week.* Retrieved from https://blogs.edweek.org/edweek/on_innovation/2017/02/design_thinking_is_a_human-centered.html

Vincent, P. Iijima, K., Driver, M., Wong, J. & Natis, Y. (2019). Gartner 2019 Magic Quadrant for Enterprise Low-Code Application Platform. Retrieved from https://www.gartner.com/doc/reprints?id=1-1FKNU1TK&ct=190711&st=sb

Wrigley, C. & Straker, K. (2015). Design Thinking pedagogy: the Educational Design Ladder. *Innovations in Education and Teaching International,* DOI: 10.1080/14703297.2015.1108214

Yanckello, R. & Calhoun Williams, K. (2019, July 29). Hype Cycle for Education, 2019. Gartner, Inc.

Editor's Note:

This paper was selected for inclusion in the journal as an EDSIGCON 2020 Meritorious Paper. The acceptance rate is typically 15% for this category of paper based on blind reviews from six or more peers including three or more former best papers authors who did not submit a paper in 2020.

Appendices

Appendix 1: Survey Instrument

The following questionnaire was administered to students for gathering information about technological self-conception and to evaluate their readiness to become Citizen Developers. The instrument includes 20 questions with five subsets measuring constructs like technological competency, usage, experience, process development, and solution application of technology.

Q01: Please indicate your agreement with the following statement: I feel competent in using technology to make my personal life better.

- Extremely competent
- Somewhat competent
- Neither competent nor incompetent
- Somewhat incompetent
- No experience or competency

Q02: For the previous question, please explain the technology you have used (or currently use) in your personal life, if any.

- Text Response

Q03: Please indicate your agreement with the following statement: I feel competent in using technology to solve problems at work or in an organizational setting.

- Extremely competent
- Somewhat competent
- Neither competent nor incompetent
- Somewhat incompetent
- No experience or competency

Q04: Please explain the technology you have used (or currently use) in your professional or work life, if any.

- Text Response

Q05: Have you ever considered creating an app for others to use? Are you or could you be interested in doing so? (Please check all that apply)

- I have considered creating an app
- I am interested in creating an app
- I feel confidence in my ability to create an app or to learn how
- I am not interested
- I have already created an app. (please explain)

Q06: Have you ever considered creating an intelligent agent for others to use? Are you or could you be interested in doing so? (Please check all that apply)

- I have considered creating an intelligent agent
- I am interested in creating an intelligent agent
- I feel confidence in my ability to create an intelligent agent or to learn how
- I am not interested
- I have already created an intelligent agent. (please explain)

Q07: I have some experience in or I have interest in learning the following. (Please check all that apply)

- Computer Programming (Experience)
- Cross System Development (Experience)
- Cross Functional Development (Experience)
- Computer Programming (Interest)
- Cross System Development (Interest)
- Cross Functional Development (Interest)

Q08: My knowledge of computer programming or coding is best described as:

- Extremely Knowledgeable
- Very Knowledgeable
- Moderately Knowledgeable
- Slightly Knowledgeable
- Not knowledgeable at all

Q09: Please check which computer programming languages, if any, you have used.

- Java / Java Script
- C / C+ / C++
- C# / Visual Basic / VBA
- Python
- PHP
- TypeScript
- Shell
- Ruby
- Other (please describe)

Q10: Please describe your programming or coding experiences. Is your primary experience at work, academic, and/ or personal?

- Text Response

Q11: Before today, have you ever heard of the term workflow automation?

- No
- Yes

Q12: Before today, which of the following have you heard or read about or used? (please indicate all that apply)

- Workflow Automation
- Trigger Automation
- IFTTT (If-This-Then-That)
- Zapier
- Power Automate
- Google Forms
- KISSFLOW
- Asana
- Smartsheet
- Low-code/ No-code
- Mendix
- Others (please list)

Q13: Have you ever used workflow automation to make processes or situations better in your personal life, professional life, or hobby...even if you didn't know it was called workflow automation?

- No
- Yes

Q14: In what area(s) did you use workflow automation? (Please indicate all that apply)

- Personal lifestyle
- Business enterprise
- Entrepreneurial efforts
- Academic Endeavors
- Hobby / Personal Interest
- Volunteerism
- Social Collaboration
- Other (please describe)

Q15: What tasks did you complete using workflow automation? (Please indicate all that apply)

- Automatic Response or Notifications (email, text, etc.)
- File Download / Upload
- Edit Calendar / Schedule Automatically,
- Generate or Process an Order
- Update a Spreadsheet/ Database
- Other (please list out)

Q16: Based upon your current knowledge, how valuable do you believe workflow automation could be?

- Extremely Valuable
- Valuable
- Somewhat Valuable
- Not Likely Valuable
- No Value

Q17: I believe I have the ability to develop useful cross-system and/or cross-functional workflow automation process(es) and solutions on my own. (Please indicate your level of agreement)

- Completely agree
- Strongly Agree
- Somewhat Agree
- Disagree
- Strongly Disagree

Q18: I would like to learn more about creating workflow automation processes that would make my life easier and solve problems at work. (Please indicate your level of agreement)

- Completely agree
- Strongly Agree
- Somewhat Agree
- Disagree
- Strongly Disagree

Q19: I would like to learn more about cross-system and cross-functional systems development. (Please indicate your level of agreement)

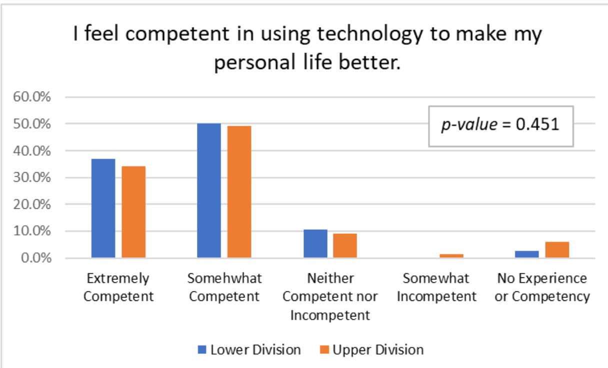
- Completely agree
- Strongly Agree
- Somewhat Agree
- Disagree
- Strongly Disagree

Q20: I would like to learn more about creating my own apps or intelligent agents. (Please indicate your level of agreement)

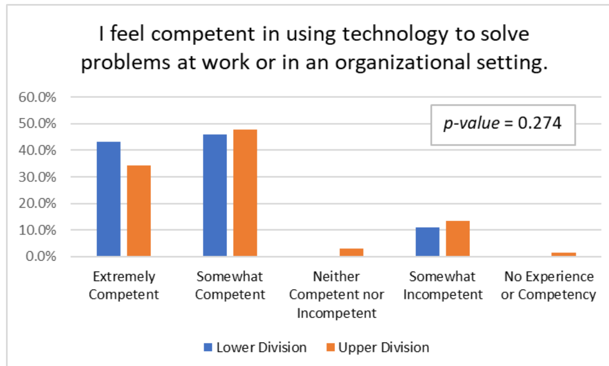
- Completely agree
- Strongly Agree
- Somewhat Agree
- Disagree
- Strongly Disagree

Appendix 2: Graphical Representation and Statistical Significance of Responses to Select Questions from the Survey as denoted in [Appendix 1] of upper-division students versus lower-division ones.

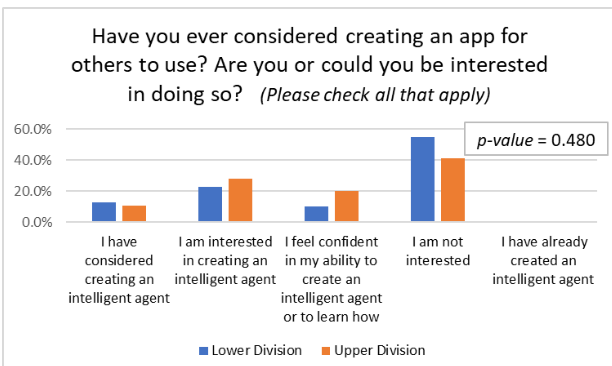
This is for testing/comparison of lower division versus upper division students to determine if they appear to be distinct populations or not for results analysis. None of the items below show statistical significance at the 0.10 level.



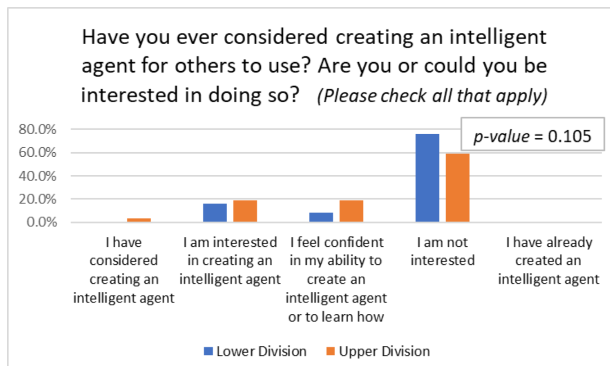
Responses to Q01



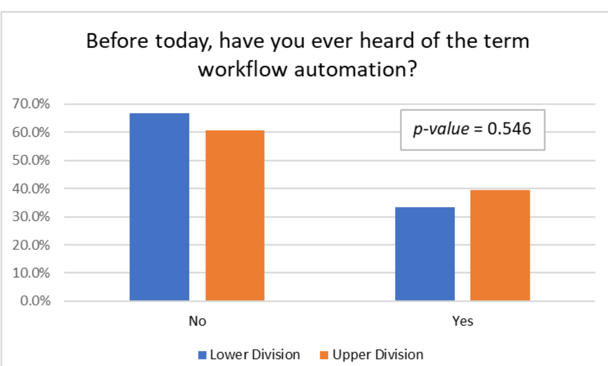
Responses to Q03



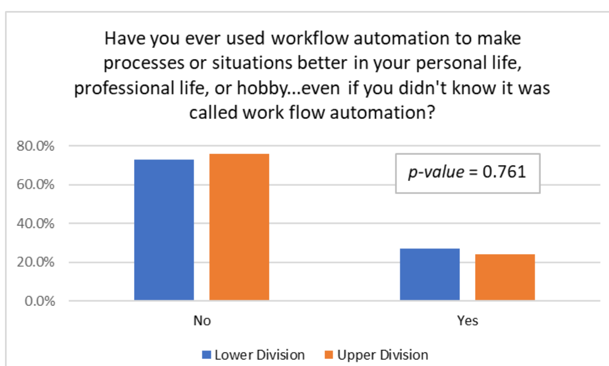
Responses to Q05



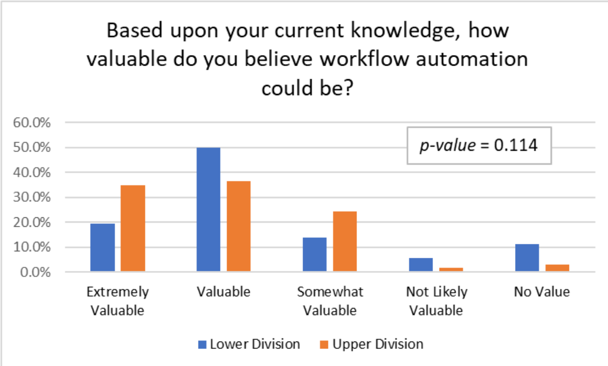
Responses to Q06



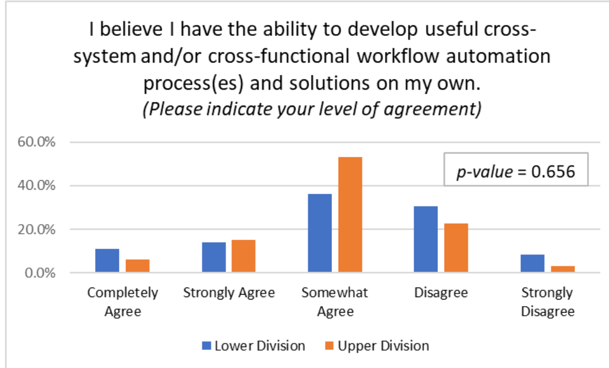
Responses to Q11



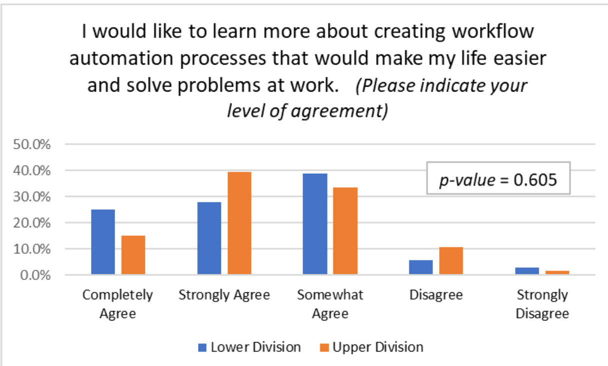
Responses to Q13



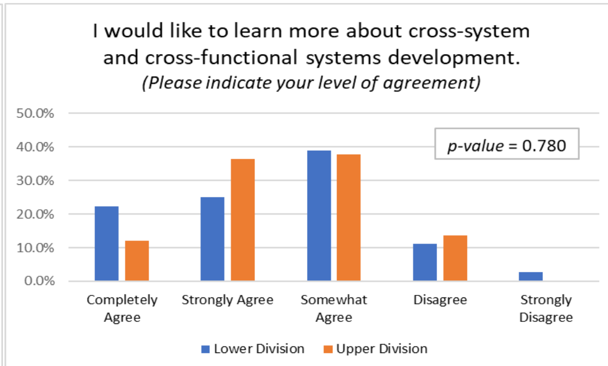
Responses to Q16



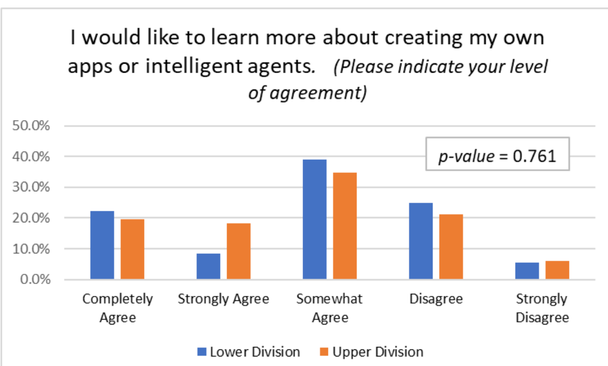
Responses to Q17



Responses to Q18



Responses to Q19



Responses to Q20

Flow Through Regression: A Guide to Interpreting and Communicating Regression Models for Data Analytics Students

Katarzyna Toskin
toskink1@southernct.edu
Business Information Systems
Southern Connecticut State University
New Haven, 06515, USA

Niki Kunene
kunenek@easternct.edu
Accounting and Business Information Systems
Eastern Connecticut State University
Willimantic, 06226, USA

Abstract

Faculty teaching data analytics at undergraduate level are often faced with the tension created by student under-preparedness, the demands of the course, and time constraints. How do faculty close this gap? In this paper, we propose the use of flow diagramming as an accessible method for interpreting regression analyses, in ways that are time efficient and not alienating to the student. Our study shows that the use of such flow diagrams has a positive effect on student understanding without additional remedial instruction. Time saved can be directed at core learning objectives of the analytics.

Keywords: Data Analytics, Regression Modeling, Flow Diagram, Flow Chart, Teaching Aid

1. INTRODUCTION

The demand for analytics knowledge has led to the incorporation of data analytics into business curricula at both graduate and undergraduate levels. Though these programs variously cater to different levels of user expertise, from the casual user to business analyst and data scientist (Watson, 2013, 2015), what they all have in common is an expectation for a foundational understanding of statistical concepts. As a result, data analytics courses typically stipulate an introductory statistics course as a minimal prerequisite because statistical understanding is foundational to explanatory statistical modeling, inferential testing and predictive analytics (Shmueli & Koppius, 2011). Yet statistics are and have often been one of the most difficult subjects for undergraduate and even graduate students to grasp. For many higher education institutions, at undergraduate level particularly, the challenge is compounded for the thousands of students who graduate high school academically underprepared for college.

Moreover, due to the large overlap between business analytics, data analytics, and information systems programs (Ceccucci, Jones, Toskin, & Leonard, 2020) many analytics courses are delivered from within Information Systems (IS) programs and therefore are taught by IS faculty. In environments where the student population is diverse, faculty teaching these courses have to manage conflicting forces such as meeting course objectives and analytics content coverage, against balancing the needs and foundational knowledge gaps of underprepared students or students intimidated by the statistical content. Faced with these constraints, it seems to us, that faculty and/or programs have one of two options. First, to accept that they have to reteach statistical foundations and yield on some of the depth in analytics content. While understandable, the downside of this approach is arguably watering down standards and increasing the cost of the program. Another approach would be for faculty to develop innovative methods and approaches that readily open access to the essential content even for underprepared and/or students with low confidence about the material. Such approaches would strategically and efficiently assist in reviewing core concepts to bring students up to speed while leaving time for coverage of analytics course objectives.

Hence, the purpose of this study is to propose an effective pedagogical, visual artifact that increases student efficacy in "relearning" important fundamental statistical concepts for analytics without explicit remedial instruction. In this pilot study, we chose the topic of regression analysis: namely, interpreting and communicating one of the most prominent and commonly used statistical modeling techniques, simple and multiple linear regression. More specifically, we use flow diagramming, an easily understandable and widely used pedagogical aid, to graphically depict the steps required to successfully interpret and communicate linear regression models. In addition, the proposed artifacts are platform independent and can be applied with a wide range of tools (i.e., SPSS, R, Excel, or Python).

2. BACKGROUND

In the experience of the authors, students taking an introductory analytics course often arrive, notwithstanding the typical statistics prerequisite, with little or no understanding of foundational statistical concepts. This lack of understanding extends both theoretically and in an applied sense. Specifically, our analysis from two data analytics undergraduate courses taught in the Fall 2019, at two regional universities, revealed the following key challenges: 80% of students struggled with interpreting and articulating the regression coefficients; 60% of students had a hard time explaining the role of R-squared; 30% of students made incorrect conclusions about the model fit.

These difficulties were not limited to regression analysis, but also extended to statistical inference. Undergraduate data analytics students, in our experience, have difficulty interpreting and communicating the results of performed analyses. In reviewing prior literature in statistics education, it shows that there are three types of reasons for these difficulties: affective (Ashaari, Judi, Mohamed, & Wook, 2011; Reid & Petocz, 2002), cognitive (Chiesi & Primi, 2010), and pedagogical (Ramsey, 1999) reasons. Weinberg & Abramowitz (2000, p. 1), researchers in statistics education, concluded that "our challenge is to find ways of presenting information to our students so that it is accessible, relevant, applicable and even vital to their own areas of interest".

Additionally, the introduction of technology in statistics education shifted the approach to teaching statistics in ways that are instructional to data analytics. In particular, technology encouraged a shift away from emphasis on computations, formulas, and procedures to an emphasis on "statistical reasoning and the ability to *interpret*, evaluate, and *flexibly apply* statistical ideas" (Ben-Zvi, 2000, p. 130) [emphasis added]. Arguably, this shift presaged the widespread adoption and use of data analytics where ability to interpret, evaluate and apply to a variety of contexts is essential for analytics students.

3. FLOW DIAGRAMMING USE FOR PEDAGOGY

Flow diagrams, artifacts in computer science and information systems were first introduced into computing by John von Neuman in the 1940s; they were introduced as a visual representation of the logical structure of a computer program (Ensmenger, 2016). At the time of the ENIAC project, it is understood that flow diagramming was chosen as a form of representation that was readily accessible to the diverse members of the team with different levels of prior knowledge. Flow diagramming was also seen as "superior to introducing a more radical departure in (logical) notation" that some members would have been familiar with (Arawjo, 2020). Priestley's (2018) historical treatment of Von Neuman's work retells how flow diagramming was used in the planning and coding reports of the project to broaden access to understanding of the work by a diverse team:

> *"[W]e have acquired a conviction that this programming is best accomplished with the help of some graphical representation of the problem. We have attempted ... to*

standardize upon a graphical notation ... in the hope that [it] would be sufficiently explicit to make quite clear to a relatively unskilled operator the general outline of the procedure. We further hope that from such a block-diagram the operator will be able with ease to carry out a complete coding of a problem" (p. 59).

Flow diagramming use increased, primarily because of Von Neuman's fame and it eventually grew to have a commanding influence on software engineering and programming for decades to come (Arawjo, 2020). Flow diagrams have since been used in a variety of contexts, namely: in modeling production processes; in aviation for training and process management (Yazgan & Yilmaz, 2018); in the accounting field to teach CPAs to communicate complex plans (Lehman, 2000); as a quality improvement tool for documenting, understanding, analyzing, and improving business processes (Nesbitt, 1993); to aid reading comprehension in the teaching of the law (Zacharias, 1986). Flow diagram use endures in aiding the teaching of introductory programming and systems analysis and design courses. Although, they have been criticized specifically for their modeling accuracy in programming (Hosch, 1977), they have lasted as both process documentation and teaching aids that make complexity readily accessible to a novice. For information systems and computer science students, they are familiar and useful aids.

4. METHOD

To illustrate the use of flow diagramming in analytics, we designed two flow diagrams: one for simple linear regression and another flow diagram for multiple linear regression. We emphasize that the focus of these artifacts is on aiding and strengthening students' capacity for interpretation and communication of analyses (implying understanding).

Description of Flow diagramming Artifacts
In this section we describe the proposed flow diagramming artifacts: to support simple linear regression interpretation (Appendix A), and to support multiple linear regression interpretation (Appendix E). We refer here to regression as a statistical method that seeks to estimate the relationship between an outcome variable and a predictor variable or set of predictors.

Simple Linear Regression
This flow diagram (Appendix A) focuses on four key elements as well as the need to articulate the regression equation: the significance F or *p-value for the F statistic*, the *intercept* or constant; the *slope* or coefficient of the independent variable and its p-value; and the interpretation of *R-squared* for model fit. We note that for simple linear regression, the instructor may want to remind students that the significance of F statistic (p-value) is equal to the p-value of the coefficient of the independent variable (IV) or slope.

Multiple Linear Regression
This flow diagram (Appendix E) focuses on five key elements as well as the need to articulate the multiple regression equation. The process is similar to simple linear regression. However, with multiple regression we assume students start with all hypothesized independent variables included in the regression model. The five key elements are: significance F (p-value for the F statistic), the *intercept* or constant; *coefficients* of the hypothesized independent variables and their respective p-values (we also assume students can, in stepwise fashion, remove non-significant variables, then re-run the model, i.e. return to Step 1 of the flow diagram); the interpretation of *R-squared* for model fit; and *adjusted R-squared*. The final step is intended to help nudge students to use *adjusted R-squared* to reinforce understanding that adjusted R-squared penalizes the addition of independent variables that do not aid in predicting the dependent variable where R-squared increases with every additional variable regardless of its effect on the dependent variable.

We note that to use both flow diagrams, we make rudimentary assumptions related to prior instruction. For instance, for multiple regression (Appendix E – Step 5), we assume that prior instruction already covered that adjusted R-squared is based on R-squared adjusted for the number of predictors and sample size. We use regression equations, in both cases, without the error term. We also assume students are previously instructed on the fundamentals underlying regression analysis including the checking of regression assumptions: linearity, normality of errors, homoscedasticity, independence of errors, and the role of residuals in assessing regression assumptions.

Lastly, the proposed pedagogical flow diagrams are intended to be used over time, with other related and ideally concept repeating assignments. In other words, they can be used again for predictive analytics based on regression models. Below we provide sample assignments and their rubrics (Appendix B, C, and D for simple linear regression; and Appendix F, G, and H for multiple linear regression). It is not our intention that these particular flow diagrams be used for instructing regression analysis from scratch per se. Rather we propose that instructors use them as a remedial mechanism, to close the gap of the forgotten or previously misunderstood concepts, and for review. The purpose is to aid students in gaining proficiency on how to *interpret* and *communicate* regression analysis results by focusing on essential information.

5. EXPERIMENTAL CONTEXT

To test the efficacy of the proposed pedagogical aids, students enrolled in a business analytics undergraduate course, in the Fall of 2020, at a regional public university were given an opportunity to use the flow diagrams as a mechanism to review assumed prior knowledge and provide feedback for this study. All students admitted to the course were required to have previously completed an introductory statistics course as well as an introductory information systems course, which covered introduction to databases, data analysis and Excel.

Study Design

Our experimental design used a pre-test/post-test approach (Campbell & Stanley, 2015) to test the effect of using the above-mentioned pedagogical aids on student understanding with respect to interpreting and communicating regression analysis results.

Students were presented with four different problems: two for simple linear regression and two for multiple linear regression. Each problem included a hypothetical scenario describing the student's role and the problem being investigated, model output (generated using Microsoft Excel data analysis tools), and seven different questions about the model.

<u>Step 1:</u> students received a pre-test for simple regression model (Appendix B), followed by a post-test (Appendix C). Although both simple linear regression models used the same data set, the variables used in each model were different. During the post-test, students were asked to use the simple linear regression flow diagram (Appendix A) as an aid to formulate responses to the assigned problem questions.

<u>Step 2</u>: for multiple linear regression, the process was similar to Step 1 above except a new data set was used to generate the models. Students were presented with a pre-test (Appendix F), followed by a post-test (Appendix G). Variables in the post-test model were changed; and only the post-test included the flow diagram (Appendix E).

To consistently evaluate students' responses to the assigned pre-test and post-test problems, grading rubrics were designed using the same criteria and scoring schema (see Appendix D for simple linear regression rubric and Appendix H for multiple linear regression rubric).

Data Collection – Participants and Procedure

Each student in the course was asked to complete the four problems. Responses were recorded using a Qualtrics survey where each problem was presented in a single screen and students were not permitted to go back to a previous screen. This ensured students could not change or correct their answers in the pre-test while completing the flow-diagram aided post-test. No review of linear regression was conducted in class, students had to rely on (assumed) prior knowledge. Out of nineteen students invited to complete the survey, fourteen students participated in the study (73% response rate).

6. RESULTS

The data was analyzed using paired sample t-tests. The results presented in Table 1 show statistically significant differences in the mean test scores between the pre-test and post-test for both simple and multiple linear regression. These results indicate that both flow diagrams had a positive effect on student understanding and interpretation of the statistical models presented in the problems. More specifically, in the simple linear regression assignment, we observed the biggest improvements in responses relating to the interpretation of the model significance, explanation of the model fit i.e., R-squared, and articulation of the findings. In the multiple linear regression assignment, the improvements were even more prominent and widespread. The biggest differences were evident in the interpretation of the intercept and model fit including adjusted R-squared, formulation and interpretation of the regression equation, calculation of a predicted value of Y as well as articulation of the overall findings. In addition, when asked "How useful did you find the flowchart aid in interpreting the data?", almost 90% of students responded that they found the flow diagram useful ranging from slightly to extremely useful. Furthermore, 93% of students said, they were somewhat likely to extremely likely to use similar flow diagramming aids for other topics in data analytics field.

	Pre-test		Post-test		
Outcome	Min	Max	Min	Max	
SLR	0	79	0	95	
MLR	0	77	0	96	
	Pre-test		Post-test		
Outcome	M	SD	M	SD	t
SLR	36	27	45	34	-3.20**
MLR	34	29	45	34	-3.94**

Note. SLR – Simple Linear Regression. *MLR* – Multiple Linear Regression, ** p < .01, n = 14

Table 1. Descriptive Statistics & t-test Results

7. DISCUSSION AND CONCLUSION

In this study, we proposed the use of strategically designed flow diagrams focused on specific knowledge gap areas in the most prominent and commonly used statistical modeling techniques, simple and multiple linear regression. We selected flow diagrams because they have been proven to foster conceptual understanding; are good alternative to lecturing; and are both time effective and time efficient. Using such flow diagrams can assist faculty in reviewing core concepts to bring students up to speed while leaving time to focus on new analytics content.

From a student perspective, flow diagrams are easy to understand and perhaps even familiar for students in information systems and computer science; ease of use and familiarity are precursors to favorable affective evaluation. We believe creating mechanism for underprepared students to quickly feel more comfortable or less intimidated by the demands and prior knowledge assumptions of analytics courses can increase student retention and avert conditions where students struggle or prematurely drop out of the course.

Additionally, our study shows that the test scores were statistically significantly higher when using flow diagrams. Similar methods may not only help student understanding of individual concepts but also may serve as important tools for managing remedial work in analytics classes.

Finally, information systems students, in particular, could be encouraged to create their own flow diagrams for other analytical processes they find individually challenging, thus unlocking complexity for themselves. e.g., systematically checking regression assumptions, hypothesis testing from problem statement, data analysis to drawing correct conclusion.

8. LIMITATIONS AND FUTURE RESEARCH

The proposed pedagogical flow diagrams do not make a claim to comprehensively cover all issues related to regression analysis. For instance, an iterative part of the analysis includes examining linear regression assumptions inclusive of examining and interpreting residuals; these flow diagram aids do not incorporate that. To minimize the complexity and maintain the effectiveness of the aid, we believe it would require a different but similar flow diagraming aid. Such an aid could, "walk" a student through how to use/interpret the diagnostic features and charts for assessing residuals generated by most statistical tools like R, SPSS and Excel. For example, another flow diagram could be used to aid a student needing remedial activity on how to run the output to examine residuals, remove outliers, or log-transform the data and re-run the regression model before interpretation. Likewise, we do not explicitly model or review steps for performing stepwise regression for multiple regression. For remedial activities, instructors can design such aids.

Finally, this study is limited by a small sample size. However, we reported qualitative observations about students results and specifically which questions to the problems were addressed with more success while using the aids, to supplement our quantitative analysis and increase the validity of this study.

9. REFERENCES

Arawjo, I. (2020). *To Write Code: The Cultural Fabrication of Programming Notation and Practice.* Paper presented at the Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems.

Ashaari, N. S., Judi, H. M., Mohamed, H., & Wook, M. T. (2011). Student's attitude towards statistics course. *Procedia-Social and Behavioral Sciences, 18*, 287-294.

Bahr, P. R. (2008). Does mathematics remediation work?: A comparative analysis of academic attainment among community college students. *Research in Higher Education, 49*(5), 420-450.

Bahr, P. R. (2010). Preparing the underprepared: An analysis of racial disparities in postsecondary mathematics remediation. *The Journal of Higher Education, 81*(2), 209-237.

Ben-Zvi, D. (2000). Toward understanding the role of technological tools in statistical learning. *Mathematical thinking and learning, 2*(1-2), 127-155.

Campbell, D. T., & Stanley, J. C. (2015). Experimental and quasi-experimental designs for research. Ravenio Books.

Ceccucci, W., Jones, K., Toskin, K., & Leonard, L. (2020). Undergraduate Business Analytics and the Overlap with Information Systems Programs. *Information Systems Education Journal, 18*(4), 22-32.

Chiesi, F., & Primi, C. (2010). Cognitive and non-cognitive factors related to students' statistics achievement. *Statistics Education Research Journal, 9*(1), 6-26.

Ensmenger, N. (2016). The multiple meanings of a flowchart. *Information & Culture, 51*(3), 321-351.

Hosch, F. A. (1977). Whither flowcharting? *ACM SIGCSE Bulletin, 9*(3), 66-73.

Lehman, M. W. (2000). Flowcharting made simple. *Journal of Accountancy, 190*(4), 77.

Nesbitt, T. E. (1993). Flowcharting business processes. *Quality, 32*(3), 34.

Priestley, M. (2018). *Routines of Substitution: John Von Neumann's Work on Software Development, 1945–1948*: Springer.

Ramsey, J. B. (1999). Why do students find statistics so difficult. *Proccedings of the 52th Session of the ISI. Helsinki*, 10-18.

Reid, A., & Petocz, P. (2002). Students' conceptions of statistics: A phenomenographic study. *Journal of Statistics Education, 10*(2).

Shmueli, G., & Koppius, O. R. (2011). Predictive analytics in information systems research. *MIS quarterly*, 553-572.

Watson, H. J. (2013). The business case for analytics. *BizEd, 12*(3), 48-54.

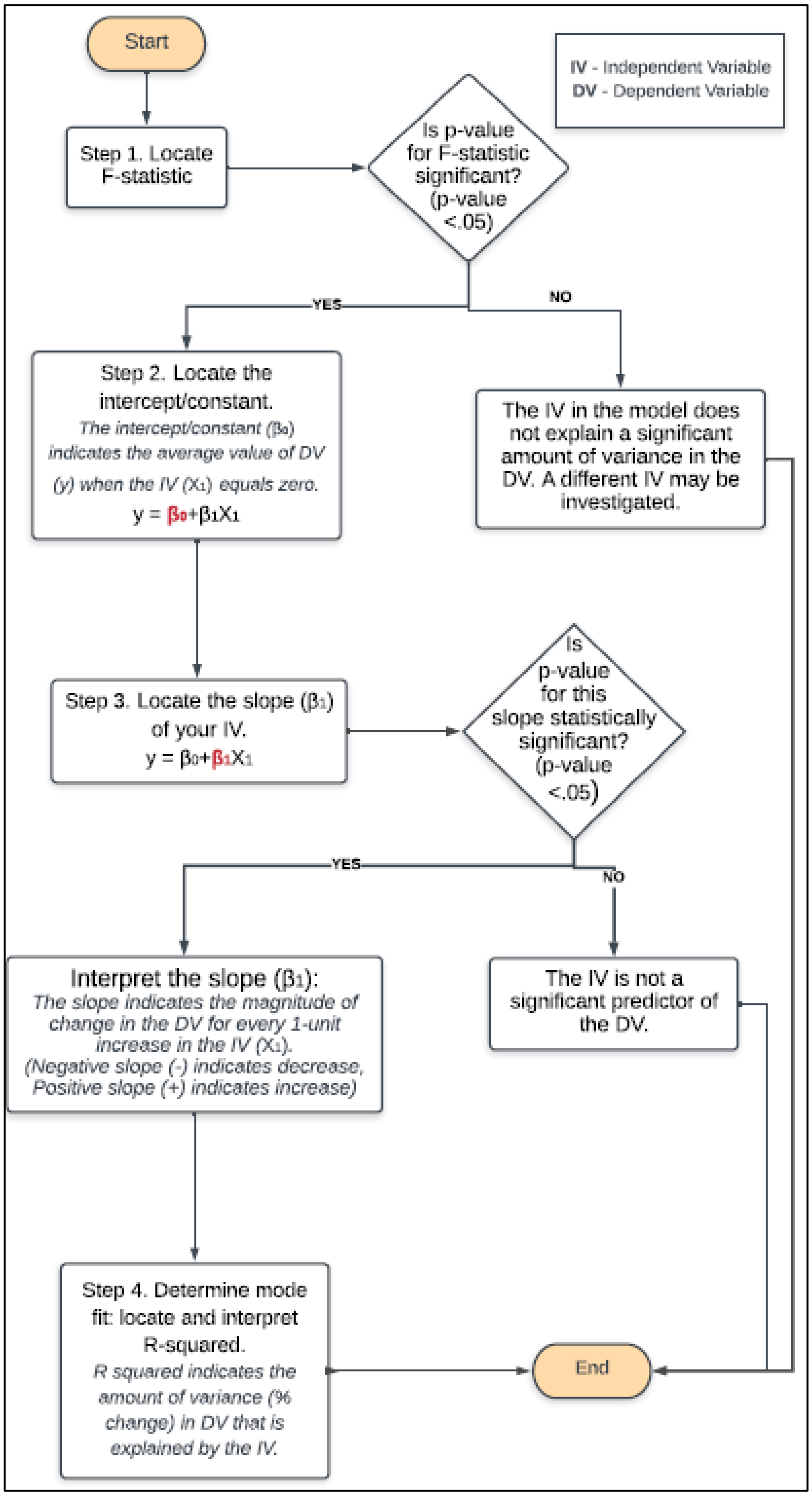
Watson, H. J. (2015). Should you pursue a career in BI/analytics. *Student Edition of the Business Intelligence Journal*.

Weinberg, S., & Abramowitz, S. (2000). Making general principles come alive in the classroom through an active case studies approach. *Journal of Statistics Education, 8*(2), 15.

Yazgan, E., & Yilmaz, A. K. (2018). Designing flow chart for aircraft type training in aviation training process management. *Aircraft Engineering and Aerospace Technology*.

Zacharias, F. C. (1986). Flowcharting the First Amendment. *Cornell L. Rev., 72*, 936.

Appendix A:
Simple Linear Regression Flow Diagram



Appendix B:
Simple Linear Regression Assignment – Pre-test

Instructions: You are a research analyst for an automotive industry. Your boss has asked you to investigate if the engine displacement size (in liters) affects the highway fuel economy (miles per gallon) based on a sample of 234 vehicles. Given the following input from a regression analysis, please interpret the results by answering the questions listed below.

SUMMARY OUTPUT

Regression Statistics	
Multiple R	0.766020021
R Square	0.586786672
Adjusted R Square	0.58500558
Standard Error	3.835984908
Observations	234

ANOVA

	df	*SS*	*MS*	*F*	*Significance F*
Regression	1	4847.833384	4847.833384	329.4533329	2.03897E-46
Residual	232	3413.829009	14.71478021		
Total	233	8261.662393			

	Coefficients	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*Lower 95.0%*	*Upper 95.0%*
Intercept	35.69765105	0.720367619	49.55476915	2.1235E-125	34.27835256	37.11694954	34.27835256	37.11694954
Displacement	-3.530588806	0.194513697	-18.15084937	2.03897E-46	-3.913827848	-3.147349764	-3.913827848	-3.147349764

Questions:

1. Interpret the overall significance of the regression model. Report the statistic you used to answer this question and its value. (10 points)
2. Report and Interpret the intercept/constant. (10 points)
3. Report and Interpret the slope/regression coefficient of the independent variable (10 points)
4. Report and Interpret the model fit (10 points)
5. Formulate and Interpret the regression equation for this model (10 points)
6. Use the model to predict the value of Y given X1=3.7. (10 points)
7. Write a precise summary of your findings to your boss. Remember that your boss is not a data analyst so make sure to articulate your findings in an easy to understand language. (40 points)

Notes: Please see below, the description of the variables and a preview of the first 10 records of the data set.

Manufacturer	Model	Displacement	Year	Cylinders	Transmission	Drive	City	Highway	Fuel	Class
audi	a4	1.8	1999	4	auto(l5)	f	18	29	p	compact
audi	a4	1.8	1999	4	manual(m5)	f	21	29	p	compact
audi	a4	2	2008	4	manual(m6)	f	20	31	p	compact
audi	a4	2	2008	4	auto(av)	f	21	30	p	compact
audi	a4	2.8	1999	6	auto(l5)	f	16	26	p	compact
audi	a4	2.8	1999	6	manual(m5)	f	18	26	p	compact
audi	a4	3.1	2008	6	auto(av)	f	18	27	p	compact
audi	a4 quattro	1.8	1999	4	manual(m5)	4	18	26	p	compact
audi	a4 quattro	1.8	1999	4	auto(l5)	4	16	25	p	compact
audi	a4 quattro	2	2008	4	manual(m6)	4	20	28	p	compact

Manufacturer – name of the manufacturer of the car
Model – model name
Displacement – engine displacement (size) in liters
Year – year of car manufacture
Cylinders – number of cylinders
Transmission – type of transmission such as automatic or manual
Drive – type of drive such as f = front wheel drive, r = rear wheel drive, 4 = four-wheel drive
City – city miles per gallon
Highway – highway miles per gallon
Fuel – fuel type such as p = premium, r = regular
Class – type of car

Appendix C:
Simple Linear Regression Assignments – Post-test

Instructions: You are a research analyst for an automotive industry. Your boss has asked you to investigate if the number of cylinders affects the city fuel economy (miles per gallon) based on a sample of 234 vehicles. Given the following input from a regression analysis, **as well as using the flow chart included at the end of this assignment**, please interpret the results by answering the questions listed below.

SUMMARY OUTPUT

Regression Statistics	
Multiple R	0.805771408
R Square	0.649267562
Adjusted R Square	0.647755785
Standard Error	2.525910807
Observations	234

ANOVA

	df	*SS*	*MS*	*F*	*Significance F*
Regression	1	2740.13386	2740.13386	429.4728925	1.06838E-54
Residual	232	1480.212294	6.380225406		
Total	233	4220.346154			

	Coefficients	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*Lower 95.0%*	*Upper 95.0%*
Intercept	29.39043476	0.626831425	46.88730267	2.4998E-120	28.1554252	30.62544432	28.1554252	30.62544432
Cylinders	-2.127983841	0.102683449	-20.72372777	1.06838E-54	-2.330295079	-1.925672604	-2.330295079	-1.925672604

Questions:

1. Interpret the overall significance of the regression model. Report the statistic you used to answer this question and its value. (10 points)
2. Report and Interpret the intercept/constant. (10 points)
3. Report and Interpret the slope/regression coefficient of the independent variable (10 points)
4. Report and Interpret the model fit (10 points)
5. Formulate and Interpret the regression equation for this model (10 points)
6. Use the model to predict the value of Y given X1=8. (10 points)
7. Write a precise summary of your findings to your boss. Remember that your boss is not a data analyst so make sure to articulate your findings in an easy to understand language. (40 points)

Notes: Please see below, the description of the variables and a preview of the first 10 records of the data set.

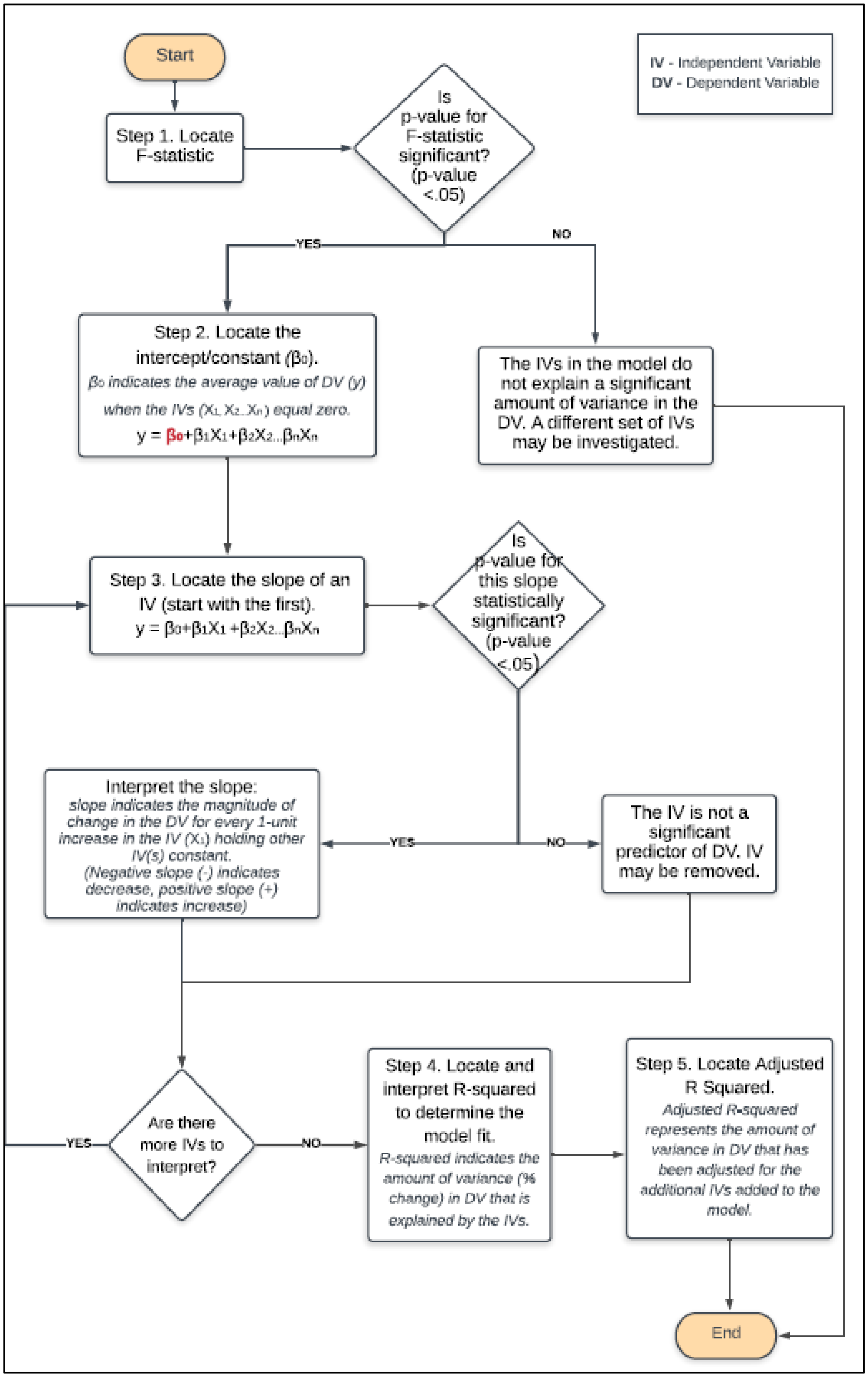
Manufacturer	Model	Displacement	Year	Cylinders	Transmission	Drive	City	Highway	Fuel	Class
audi	a4	1.8	1999	4	auto(l5)	f	18	29	p	compact
audi	a4	1.8	1999	4	manual(m5)	f	21	29	p	compact
audi	a4	2	2008	4	manual(m6)	f	20	31	p	compact
audi	a4	2	2008	4	auto(av)	f	21	30	p	compact
audi	a4	2.8	1999	6	auto(l5)	f	16	26	p	compact
audi	a4	2.8	1999	6	manual(m5)	f	18	26	p	compact
audi	a4	3.1	2008	6	auto(av)	f	18	27	p	compact
audi	a4 quattro	1.8	1999	4	manual(m5)	4	18	26	p	compact
audi	a4 quattro	1.8	1999	4	auto(l5)	4	16	25	p	compact
audi	a4 quattro	2	2008	4	manual(m6)	4	20	28	p	compact

Manufacturer – name of the manufacturer of the car
Model – model name
Displacement – engine displacement (size) in liters
Year – year of car manufacture
Cylinders – number of cylinders
Transmission – type of transmission such as automatic or manual
Drive – type of drive such as f = front wheel drive, r = rear wheel drive, 4 = four-wheel drive
City – city miles per gallon
Highway – highway miles per gallon
Fuel – fuel type such as p = premium, r = regular
Class – type of car

Appendix D:
Rubric for Grading Simple Linear Regression Pre-test and Post-test

Grading Rubric: Simple Linear Regression Assignment							
Student Name:							
	Needs Improvement	Satisfactory	Excellent				
	0-1 points	2-3 points	4-5 points	Weight	Points Possible	Points Earned	Weighted Points
Interpretation of overall significance of the model: F statistic/Null Hypothesis	The significance of the model is not reported using the correct statistic. The interpretation is missing or completely incorrect.	The significance of the model is reported correctly and interpretation is partially correct.	The significance of the regression model is reported correctly and interpreted accurately.	0.1	5		0
Interpretation of the intercept/constant: B0	Intercept/constant of regression line is not reported correctly. The interpretation is missing or completely incorrect for the constant/slope.	Intercept/constant is reported correctly and interpretation is partially correct.	Intercept/constant is reported and interpreted accurately.	0.1	5		0
Interpretation of the slope/regression coefficient: B1	Slope/regression coefficient is not reported correctly. The interpretation is missing or completely incorrect for the slope/regression coefficient.	Slope/regression coefficient is reported correctly and interpretation is partially correct.	Slope/regression coefficient is reported and interpreted accurately.	0.1	5		0
Interpretation of model fit: R-squared	R-squared is not reported correctly. The interpretation is missing or completely incorrect for R squared.	R-squared is reported correctly and interpretation is partially correct.	R-squared is reported and interpreted accurately.	0.1	5		0
Formulation and Interpretation of regression equation Y hat = B0+B1X1	The equation is not formulated correctly. The interpretation of the equation is missing or completely incorrect.	The equation is formulated correctly and interpretation is partially correct.	The equation is formulated and interpreted accurately.	0.1	5		0
Calculate the predicted value of Y using a specific value of X1	The predicted value of Y is not calculated correctly.	The predicted value of Y is partially correct.	The predicted value of Y is calculated accurately.	0.1	5		0
Articulate your findings: Write a summary of your findings.	The articulation is incorrect or missing.	The articulation is partially complete and partially correct.	The articucation is complete and accurate.	0.4	5		0
						Total Score	**0**

Appendix E:
Multiple Linear Regression Flow diagram



Appendix F:
Multiple Linear Regression Assignments – Pre-test

Instructions: Department of Public Safety (DOS) has hired you as a consultant to help them explore historical data collected from 50 states of the United States of America. Their first priority is to study if Population size (in millions) and Income per capita (in thousands) affect murder rate (%). Given the following output from a regression analysis, please interpret the results by answering the questions listed below.

SUMMARY OUTPUT

Regression Statistics	
Multiple R	0.461732393
R Square	0.213196803
Adjusted R Square	0.179715816
Standard Error	3.343410573
Observations	50

ANOVA

	df	*SS*	*MS*	*F*	*Significance F*
Regression	2	142.3612699	71.18063494	6.367697658	0.003571525
Residual	47	525.3845301	11.17839426		
Total	49	667.7458			

	Coefficients	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*Lower 95.0%*	*Upper 95.0%*
Intercept	14.34337892	3.490659875	4.109073767	0.000157786	7.321077034	21.36568081	7.321077034	21.36568081
Population	0.338434824	0.109381813	3.09406851	0.00332034	0.118386999	0.558482648	0.118386999	0.558482648
Income	-0.189425027	0.079472426	-2.383531448	0.021239504	-0.349302926	-0.029547127	-0.349302926	-0.029547127

Questions:

1. Interpret the overall significance of the regression model. Report the statistic you used to answer this question and its value. (10 points)
2. Report and Interpret the intercept/constant. (10 points)
3. Report and Interpret the slopes/regression coefficients of the independent variables (10 points)
4. Report and Interpret the model fit (10 points)
5. Formulate and Interpret the regression equation for this model (10 points)
6. Use the model to predict the value of Y given X1=10, X2=50. (10 points)
7. Write a precise summary of your findings to DOS. Remember that your boss is not a data analyst so make sure to articulate your findings in an easy to understand language. (40 points)

Notes: Please see below, the description of the variables and a preview of the first 10 records of the data set.

	Population	Income	Illiteracy	LifeExp	Murder	HSGrad	Frost	Area
Alabama	3.615	36.24	2.1	69.05	15.1	41.3	20	50708
Alaska	0.365	63.15	1.5	69.31	11.3	66.7	152	566432
Arizona	2.212	45.3	1.8	70.55	7.8	58.1	15	113417
Arkansas	2.11	33.78	1.9	70.66	10.1	39.9	65	51945
California	21.198	51.14	1.1	71.71	10.3	62.6	20	156361
Colorado	2.541	48.84	0.7	72.06	6.8	63.9	166	103766
Connecticut	3.1	53.48	1.1	72.48	3.1	56	139	4862
Delaware	0.579	48.09	0.9	70.06	6.2	54.6	103	1982
Florida	8.277	48.15	1.3	70.66	10.7	52.6	11	54090
Georgia	4.931	40.91	2	68.54	13.9	40.6	60	58073

Population – population estimate in millions as of July 1, 1975
Income – income per capital as of 1974, in thousands
Illiteracy – percent of population who is illiterate as of 1970
LifeExp – life expectancy in years from 1969-1971
Murder – murder and non-negligent manslaughter rate per 100,000 population as of 1970
HSGrad – Percent of high-school graduates as of 1970
Frost – average days with minimum temperature below freezing between 1930-1960
Area – land area in square miles

Appendix G: Multiple Linear Regression Assignments – Post-test

Instructions: Department of Public Safety (DOS) has hired you as a consultant to help them explore historical data collected from 50 states of the United States of America. They would like to investigate if Murder rate (%) and High School graduation rate (%) affect Life expectancy (number of years). Given the following output from a regression analysis, **as well as using the flow chart attached on the second page of this assignment**, please interpret the results by answering the questions listed below.

SUMMARY OUTPUT

Regression Statistics	
Multiple R	0.81415359
R Square	0.662846068
Adjusted R Square	0.648499092
Standard Error	0.795871738
Observations	50

ANOVA

	df	*SS*	*MS*	*F*	*Significance F*
Regression	2	58.52864627	29.26432314	46.20110018	8.0161E-12
Residual	47	29.77035573	0.633411824		
Total	49	88.299002			

	Coefficients	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*Lower 95.0%*	*Upper 95.0%*
Intercept	70.29708402	1.015667882	69.21266811	5.90943E-49	68.25382379	72.34034424	68.25382379	72.34034424
Murder	-0.237090095	0.035285233	-6.719244063	2.18051E-08	-0.308074828	-0.166105361	-0.308074828	-0.166105361
HSGrad	0.043887299	0.016126888	2.721374355	0.009088366	0.011444185	0.076330413	0.011444185	0.076330413

Questions:

1. Interpret the overall significance of the regression model. Report the statistic you used to answer this question and its value. (10 points)
2. Report and Interpret the intercept/constant. (10 points)
3. Report and Interpret the slopes/regression coefficients of the independent variables (10 points)
4. Report and Interpret the model fit (10 points)
5. Formulate and Interpret the regression equation for this model (10 points)
6. Use the model to predict the value of Y given X1=8, X2=70. (10 points)
7. Write a precise summary of your findings to DOS. Remember that your boss is not a data analyst so make sure to articulate your findings in an easy to understand language. (40 points)

Notes: Please see below, the description of the variables and a preview of the first 10 records of the data set.

	Population	Income	Illiteracy	LifeExp	Murder	HSGrad	Frost	Area
Alabama	3.615	36.24	2.1	69.05	15.1	41.3	20	50708
Alaska	0.365	63.15	1.5	69.31	11.3	66.7	152	566432
Arizona	2.212	45.3	1.8	70.55	7.8	58.1	15	113417
Arkansas	2.11	33.78	1.9	70.66	10.1	39.9	65	51945
California	21.198	51.14	1.1	71.71	10.3	62.6	20	156361
Colorado	2.541	48.84	0.7	72.06	6.8	63.9	166	103766
Connecticut	3.1	53.48	1.1	72.48	3.1	56	139	4862
Delaware	0.579	48.09	0.9	70.06	6.2	54.6	103	1982
Florida	8.277	48.15	1.3	70.66	10.7	52.6	11	54090
Georgia	4.931	40.91	2	68.54	13.9	40.6	60	58073

Population – population estimate in millions as of July 1, 1975
Income – income per capital as of 1974, in thousands
Illiteracy – percent of population who is illiterate as of 1970
LifeExp – life expectancy in years from 1969-1971
Murder – murder and non-negligent manslaughter rate per 100,000 population as of 1970
HSGrad – Percent of high-school graduates as of 1970
Frost – average days with minimum temperature below freezing between 1930-1960
Area – land area in square miles

Appendix H:
Rubric for Grading Multiple Linear Regression Pre-test and Post-test

Grading Rubric: Multiple Linear Regression Assignment							
Student Name:							
	Needs Improvement	Satisfactory	Excellent				
	0-1 points	2-3 points	4-5 points	Weight	Points Possible	Points Earned	Weighted Points
Interpretation of overall significance of the model: F statistic/Null Hypothesis	The significance of the model is not reported using the correct statistic. The interpretation is missing or completely incorrect.	The significance of the model is reported correctly and interpretation is partially correct.	The significance of the regression model is reported correctly and interpreted accurately.	0.1	5		0
Interpretation of the intercept/constant: B0	Intercept/constant of regression line is not reported correctly. The interpretation is missing or completely incorrect for the constant/slope.	Intercept/constant is reported correctly and interpretation is partially correct.	Intercept/constant is reported and interpreted accurately.	0.1	5		0
Interpretation of the slope/regression coefficient: B1 & B2	Slope/regression coefficients are not reported correctly. The interpretation is missing or completely incorrect for the slope/regression coefficients.	Slope/regression coefficients are reported correctly and interpretation is partially correct.	Slope/regression coefficients are reported and interpreted accurately.	0.1	5		0
Interpretation of model fit: R-squared and Adjusted R-squared	R-squared and Adjusted R-squared are not reported correctly. The interpretation is missing or completely incorrect for R squared.	R-squared and Adjusted R-squared are reported correctly and interpretation is partially correct.	R-squared and Adjusted R-squared are reported and interpreted accurately.	0.1	5		0
Formulation and Interpretation of regression equation Y hat = B0+B1X1+B2X2	The equation is not formulated correctly. The interpretation of the equation is missing or completely incorrect.	The equation is formulated correctly and interpretation is partially correct.	The equation is formulated and interpreted accurately.	0.1	5		0
Calculate the predicted value of Y using a specific value of X1 & X2	The predicted value of Y is not calculated correctly.	The predicted value of Y is partially correct.	The predicted value of Y is calculated accurately.	0.1	5		0
Articulate your findings: Write a summary of your findings.	The articulation is incorrect or missing.	The articulation is partially complete and partially correct.	The articuation is complete and accurate.	0.4	5		0
						Total Score	**0**

Distributed Project Teams and Software Development

An Introduction to the use of Git and GitHub for ASP.NET MVC Development

Thom Luce
luce@ohio.edu
Analytics and Information Systems Department
Ohio University
Athens, OH 45701

Abstract

This paper describes changes, precipitated by the Covid-19 pandemic, to a capstone MIS class using Microsoft ASP.NET MVC for team development with live-clients. The advent of the pandemic required that the entire development effort of the class immediately transition from a largely in-class development effort with local SQL Server and Web Server Instances to one requiring all development be done in a virtual desktop interface (VDI). The VDI was the only way for students to get to both the SQL Server instance and the web server where they published their applications. Code availability, version control and joint development issues were resolved with Git and the Visual Studio interface to GitHub. This paper summarizes the development issues faced by the student teams, how they were resolved and provides a brief introduction to the use of GitHub from within the current version of Visual Studio. The paper is descriptive, and the subjective nature of the live-client project deliverables made any significant statistical analysis impossible.

Keywords: Version control, Source code management, Git, GitHub, Capstone course, Pandemic

1. INTRODUCTION

A few years ago, there was an active discussion on the Computer Science Educator's Stack Exchange (Anon, n.d.) about why more universities didn't teach revision control. While this discussion focused on computer science and engineering programs, many of the considerations also apply to IS and MIS programs.

The (Anon n.d.) article says that one reason may be that it is hard to teach some of the concepts if students don't have an opportunity to practice the concepts and that normally requires team projects and longer projects than are typically assigned. They also say that students typically don't see a reason for using source code management (version control) until they get into more complex problems than those they typically get in the classroom. The course described in this paper fits those criteria with a three-part individual project where each part builds on the previous part and a major team project involving the development of an application for an external client.

Uzunbayir (2018) points out that source code management is especially appropriate for continuous software development projects which are based on agile practices. The course and project described here and further documented in (Luce 2017, 2020) use Scrum with two-week Sprints and a published, potentially deliverable product at the end of each Sprint.

According to Andersson (2018) and Marko (2019) good version control should include, among other things

1. Making and committing small changes
2. Committing frequently
3. Not committing generated code
4. Only committing verified, test code

5. Making good commit comments.

The pandemic induced version of the class, using GitHub as described here forced students to make and commit small changes frequently because they have to commit every time they left the VDI.

The use of. gitignore allows the students to control which files are committed each time. The one exception is the .ddl file created when the application is built, and this can be handled by essentially ignoring it and always keeping the local version.

Point number 4 is harder to police but students in general understand the idea of garbage-in, garbage-out and soon learn that committing untested and unvalidated code leads to the propagation of errors among their team members.

The last point becomes clearer when students try to find and reverse changes that they or a teammate committed to the repository.

2. PROBLEM DEFINITION

Eighty-two students, most of them graduating seniors and all taking a capstone live-client software development project, left campus for spring break on March 7, 2020. They never returned to campus. While they were on break the Covid-19 pandemic struck with full force causing the University to shut down all face-to-face classes, close campuses and move all classes online.

The students, working in teams of 4-6, were four weeks into an eight-week live software development effort, developing an application for a remote client using Microsoft ASP.NET MVC tools (Note: We are planning to more to MVC Core during the next academic year) and managing their projects with Scrum. Prior to the campus closure all students had access to computer labs with all the development software they required, the college network with a full SQL Server instance and a shared web server where they could publish their applications.

Additionally, the students had access to a virtual desktop interface (VDI), implemented with VMWare Horizon (VMWare Horizon 7. (n.d.)). The VDI gave them access to the college network and servers, including the SQL Server instance and a web server from anywhere in the world and allowed them to work on projects when they weren't working on a lab computer. Unfortunately, the VDI installation at the time was configured to present a clean environment every time a student logged in. This meant students were unable to store files on the virtual desktop or anywhere on the virtual machine. They could pull files into the VDI from their local computer but that didn't help when they were working remotely. There were two UNC (universal naming convention) networked drives available from the labs and in the VDI environment where students could store files. Unfortunately, students could not work directly on applications from the network drives because a number of important ASP.NET MVC development operations, such as data migration and publishing, didn't work properly when the source code was stored on a networked drive.

Access to the VDI and the networked drives was password protected and limited to registered students, so sharing files with team members was difficult. MVC applications tend to be too large to email. For example, the compressed version of the sample application developed for this paper started at over 75Mbytes. As a result, students were unable to share work via email. Students were also unable to share network resources unless they were also willing to share passwords and, since these passwords were used in numerous University systems, most students weren't willing to do that.

Students learned how to develop ASP webforms applications in earlier classes and we had established a series of shared folders on networked drives that would allow students to save applications and to share work. Unfortunately, work on MVC applications is more complex than work on webforms applications and shared folders didn't provide a good solution. As mentioned previously, a number of important development steps don't work correctly in the Visual Studio ASP.NET MVC environment if the source code is stored on a network drive. Because of this, the entire MVC application had to be copied back and forth between the network drive and a local drive to be used. Then there was the most important consideration of all, because of all the interrelated files in the MVC environment, only one student at a time could safely work on application development. Students could not simply work on different parts of the application and then copy the individual files to one central location. Additionally, copying the application folder back to the shared drive did not enforce any kind of version control and the newly revised work simply replaced the existing files on the drive. Size limitations on the networked drives made simple version control via file and folder renaming impractical if not impossible.

While we had tried different approaches to version control software in the past, trying systems like Microsoft Team Foundation Server, now known as Azure DevOps Server ("Microsoft Team Foundation Server," n.d.) and git ("Git," n.d.), but our students preferred to work in our labs, finding the version control systems either too complex or too foreign to what they knew. One reason for this is that our program has always focused on solving business problems using technology, rather than learning technology for its own sake. We limit the number of software platforms and development environments students are exposed to and focus on what needed to be done by any set of tools. We had, in fact, standardized our MIS assignments around Microsoft Visual Studio, C# and SQL Server a number of years ago for precisely that reason. Because of this approach, most of our students have never seen a command line process like that used by git or the Package Manager in Visual Studio. While it could be argued that learning to use a command line tool would be good for their education, our program wanted to focus is on solving business problems, not learning technologies or new, or old, interfaces.

So, despite our attempts to introduce version control software the biggest criticism of the course, semester after semester, was the inability to have more than one person working on the application at a time.

Things started to change in the fall of 2019 when we realized that Visual Studio (VS) had improved support for git and GitHub within the Visual Studio environment (Nadagouda, 2020). We introduced the Visual Studio interface to GitHub at the beginning of a three-part individual software development assignment designed to help prepare students for the team development effort that would follow. We attempted to sell it on the idea that they would now be able to work from anywhere, would eventually be able to do co-development, wouldn't be limited to working in our computer labs and wouldn't need to copy large applications back and forth to servers every time they wanted to work.

As might be expected, students generally didn't see the purpose while they were working on individual assignments but started to understand how it might be useful when they began the team project. During that first semester they could, however, avoid the VDI environment and work on our lab computers where their files were typically preserved for the entire semester and, being busy students, many didn't do much project work outside class and lab times.

The pandemic outbreak in the spring of 2020 changed all of that. Students could no longer use lab computers. They could no longer meet face-to-face with their teammates. They could no longer avoid the VDI environment with its access to SQL Server, the web server and networked drives. They could no longer avoid the problem that things left on the VDI desktop or on a virtual disk wouldn't be there after they left the environment. Finally, they could no longer avoid the fact that multiple team members needed to work on the project from different locations, often at the same time.

The solution to these problems was the improved graphic interface to GitHub built into Visual Studio 2019. This paper is a short overview for instructors on how to use GitHub in Microsoft Visual Studio Community Edition 2019 (Version 16.5.x). Much of the information contained here has been converted to videos to help our students use GitHub.

3. GETTING STARTED

The latest version of Microsoft Visual Studio Community Edition generally comes with support for git and GitHub preinstalled but if they aren't installed, you need to start the Visual Studio Installer, select the current version of the software and then select modification of the installation. This should bring up a window (Figure 1, Appendix) where you can select the Workloads you wish to install or modify. For most web development in ASP.NET MVC and ASP.NET Core, the ASP.NET and web development workload should work. You can also include Python, Node.js and other workloads if you use those.

To be sure you have the components for git and GitHub, click the Individual Components link and make sure the "Git for Windows" and "GitHub Extensions for Visual Studio" are selected.

Once the extensions are installed, start Visual Studio and create a new project. At the bottom right-hand side of the window (Figure 2) you will see a message that says, "Add to Source Control" and a popup that says Git when you mouse over it and click the small white arrow on the right.

Click on Git and the Team Explorer window should open to something like Figure 3A. Visual Studio supports different source code management systems including Azure DevOps and GitHub. Click "Publish to GitHub".

You will then be asked to login to GitHub, or create an account, and to name the new repository – EdSigCon2020 in this case (Figure 3B). Once you have done that, press Publish.

After the project is published the Team Explorer window should look something like Figure 3C.

At this point you can switch back to the Solution Explorer. As shown in Figure 4A, the Solution Explorer now has a small icon at the top that allows you to switch between Folder View (shown in 4B) and solution view (Shown in 4A). The window needs to be in Solution view for normal development work (creating Controllers, testing MVC pages, etc.). Folder view allows you to see all the files in the folder, including files that are hidden in Solution View.

Among the hidden files is .gitignore. This file lists files and folders that should be ignored, not copied, when a local repository is synchronized with the remote repository. The list includes various system files and compiled files that should not be replaced when a repository is cloned to a local computer. Many of these are bin and system package files. Unfortunately, ASP.NET MVC uses a number of binary files and package files that won't necessarily be on a local computer and that do need to be synchronized. To see that this happens we must edit .gitignore and comment out two specific lines.

The exact location of these lines varies but one says

**/packages/*

and the other is

[Bb]in/

To convert these lines to comments, type ## in front of each line and save the file.

Once the edited .gitignore file is saved the entire application should be saved (committed) to the local repository and to the distributed repository on GitHub. The process requires several steps, but the steps will be essentially the same every time changes are committed.

To commit changes, first switch to the Team Explore window and click the small house icon under the title bar. The Team Explorer windows should look something like Figure 5A. Next click Changes and enter a comment in the yellow comment box (now white because a comment has been entered). The comments are an important part of the changes you are committing, and you won't be able to actually commit the work until a comment is entered. As you can see in Figure 5B, the Team Explorer window shows you what changes you are about to commit. After entering a comment, press Commit Staged (sometimes this button will just say Commit All).

The commit process saves/commits changes to the local repository. Prior to the commit all code modifications since the last commit can be undone and haven't been added to the local repository. After committing changes, you need to synchronize the changes with the remote distributed repository. First press the Sync link shown in Figure 6A and then Push the changes to the remote depository (Figure 6B). Things may not always work quite as smoothly as this and you may need to merge changes (discussed below).

4. CLONING A PROJECT

One major benefit of distributed source code management is the ability to make an exact copy, a clone, of the project at any time. In the virtual desktop interface environment (VDI) described earlier this means a student can login to the VDI, clone a project and resume work. Once done the project must once again be committed and pushed to the server as previously described.

Figure 7 shows the Visual Studio 2019 startup screen where you can select to Clone an existing project. After selecting Clone, the Clone or check out code window opens (Figure 8, underneath). To clone from a remote repository, click GitHub. If you aren't already signed into GitHub you will have to do that first and then select the project you wish to clone (Figure 8, upper).

After the project is cloned the Team Explorer window will look something like Figure 9A. Click on the Solution Explorer and then the Folder View Icon (Figure 9B) and select the sln view before you start working on the project.

5. CHANGING AN APPLICATION

Any number of people can clone and work on a project at the same time. Each developer commits changes to their local repository and then attempts to sync them to the remote repository. Conflict occurs when one developer has successfully pushed changes to the remote repository that the current developer doesn't have. Figure 10A shows the error message that appears when this happens.

The error message referred to in the window says: "Error encountered while pushing to the remote repository: rejected Updates were rejected because the remote contains work that you do not have locally. This is usually caused by another repository pushing to the same ref. You may want to first integrate the remote changes before pushing again."

To resolve this problem, it is necessary to first Pull the remote changes and then resolve any conflicts between the local and the remote versions. Figure 10B shows the Team Explorer after Pulling the remote code. Notice that three conflicts are reported. To fix this, first click on the Conflicts link and then click on one of the files reported to be in conflict, in this case EdSigCon2020.dll.

Figure 10C shows the conflict resolution window that appears. This window has a "Merge" button, a Compare Files button and buttons that allow you keep the remote version or the local version of the code. In this case the file in question is a ddl file created when the project was built and one that will be recreated every time the project is built and run so it doesn't matter which version we use, but we will Keep Local.

The next file selected was called EdSigCon2020.csproj, a file used to manage the project. In this case there are differences between the remote and local code that need to be resolved for the project to work correctly. To see the differences, click the Compare Files link shown in Figure 10C.

Figure 11 shows the resulting display. The version on the left, with the code highlighted in pink is code from the remote repository while the version on the right with code highlighted in green is in the local repository. In this example the developers of both the local and the remote versions of the application had added a View to the application.

To resolve the differences, click the Merge button shown in Figure 10C. The display in Figure 11 is replaced with a similar display shown in Figure 12. The new display lists the two files side by side and highlights the differences both with color and with boxes that highlight the difference. When you click the checkbox next to one of the code samples it is automatically copied to merged code at the bottom of the screen. When you click the checkbox next to the other set of code it is merged with the first set of code.

At this point you can manually edit the merged code to make any final adjustments. Once you are satisfied with the changes, click the appropriate button to take the remote version or keep the local version, but which one is it? Notice the green color in the merged section of Figure 12. This is the local version, so you need to click Keep Local.

After merging change to EdSigCon2020.csproject there is one other file to check. The two versions of this file, HomeController.cs, have different, new methods that can be merged into one file as shown in Figure 13.

6. BRANCHING

The senior capstone project class where this was introduced has a three-part individual development assignment prior to the team project. Each part of the assignment builds on the previous parts and students often get to part 2 or part 3, make a coding mistake they can't undo and have to go back and repeat all the previous parts of the assignments before they can continue. This problem is only magnified once the team development project begins.

Git's branching capability can help solve this problem. With branching the user creates a copy of the current project, a branch, and then works on the branch while leaving the original alone. If the user makes a mistake and needs to start over, they can simply delete the branch and continue to work on the original code or make a new branch and work on it. Once the code in the branch is thoroughly tested it can be merged back into the master branch as illustrated in Figure 14

Figure 15 shows how to start the process in Visual Studio. When the Task Manager's Branches button is clicked (Figure 15A) the Task Manager displays a Create Branch window (Figure 15B) and after the branch is created the resulting list of branches in the repository is displayed (Figure 15C). To switch between a branch and the master, simply double click on the desired branch. If there are unsaved changes you will have to commit them but once that is done, you will be working on the selected branch. Figure 16 shows the bottom of the Visual Studio window with the SampleBranch selected.

A branch can be modified and tested without affecting the master or any other branch. It is also possible to create a branch from a branch.

Once the branch is tested and complete it can be merged back into the master. Figure 17A shows the Branches view of the Task Manger configured to merge the current branch back into the master branch. Figure 17B shows a conflict window since the branch contains code not found in the master branch and Figure 17C shows the conflict resolution window. The process from this point is exactly the same as the conflict resolution process outlined above.

7. CONCLUSIONS

The latest GitHub additions to Visual Studio make source code management easier than ever, especially for students with limited technical backgrounds. Students can now add projects to source code management, commit changes to local repositories and push them to remote repositories, resolve code conflicts and merge files, create and work with branches and eventually merge branches, all within Visual Studio.

The ability to perform these operations easily means students can practice version control techniques and it no longer matters where a student works. They can work on campus computers if those are available, they can work at home or even work in environments like the VDI that don't allow them to save anything and provide a clean desktop every time they run.
Students with continuing projects can use source version control on projects that build on each other, can easily create code branches, work on the branch and merge back with the original after testing is complete. In these situations, they can also avoid large amounts of rework after disastrous changes by discarding a damaged branch, reverting to an earlier branch and then moving forward with a new branch.

Students working on large, individual projects can use these tools to implement good version control practices while students working on team projects no longer have to take turns coding or figure out how to copy and paste different sets of code together. Team members can clone a project, work on their own from anywhere and then commit changes to the common shared remote repository.

All of these are skills that students will need if they continue with development careers after graduation, but they are skills that even non-technical students can learn and use now.

8. REFERENCES

Anon (n.d.), "Why don't more universities teach revision control?" (n.d.), retrieved from https://cseducators.stackexchange.com/questions/3590/why-dont-more-universities-teach-revision-control

Andersson, Mikka, "7 Version Control Best Practices for Developers,", 2018, retrieved from https://resources.collab.net/blogs/7-version-control-best-practices-for-developers, July 2020.

Git. (n.d.) retrieved from https://en.wikipedia.org/wiki/Git

Luce, T., "Adding MVC to the MIS Capstone," *Issues in Information Systems*, Vol 18 (1), pp 118-127, 2017.

Luce, T., "Evolution of an IS Capstone Class," Information Systems Education Journal, Vol 18 (1), pp 40-47, Feb 2020.

Marco, "Best Practices for Version Control in 8 steps", 2019, retrieved from https://ruleoftech.com/2019/best-practices-for-version-control-in-8-steps, July 2020.

Microsoft Team Foundation Server (n.d.) retrieved from https://docs.microsoft.com/en-us/azure/devops/server/whats-new?view=azure-devops

Nadagouda, Pratik, "Improved Git Experience in Visual Studio 2019", retrieved from https://devblogs.microsoft.com/visualstudio/improved-git-experience-in-visual-studio-2019/ , Mar 2020.

Uzunbayir, 2018, "A Review of Source Code Management Tools for Continuous Software Development", 3ed International Conference on Computer Science and Engineering, IEEE, 2018, pop 414-419

VMware Horizon 7. (n.d.) retrieved from https://www.vmware.com/products/horizon.html

Appendix – Figures

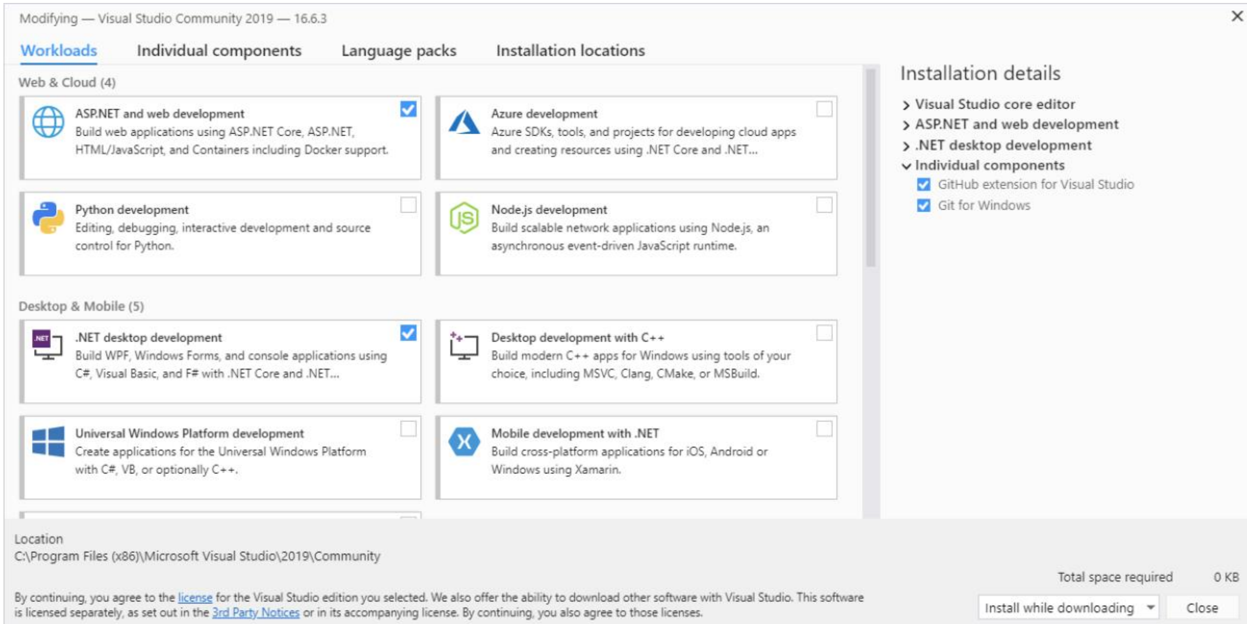


Figure 1. Visual Studio Workload Installer window

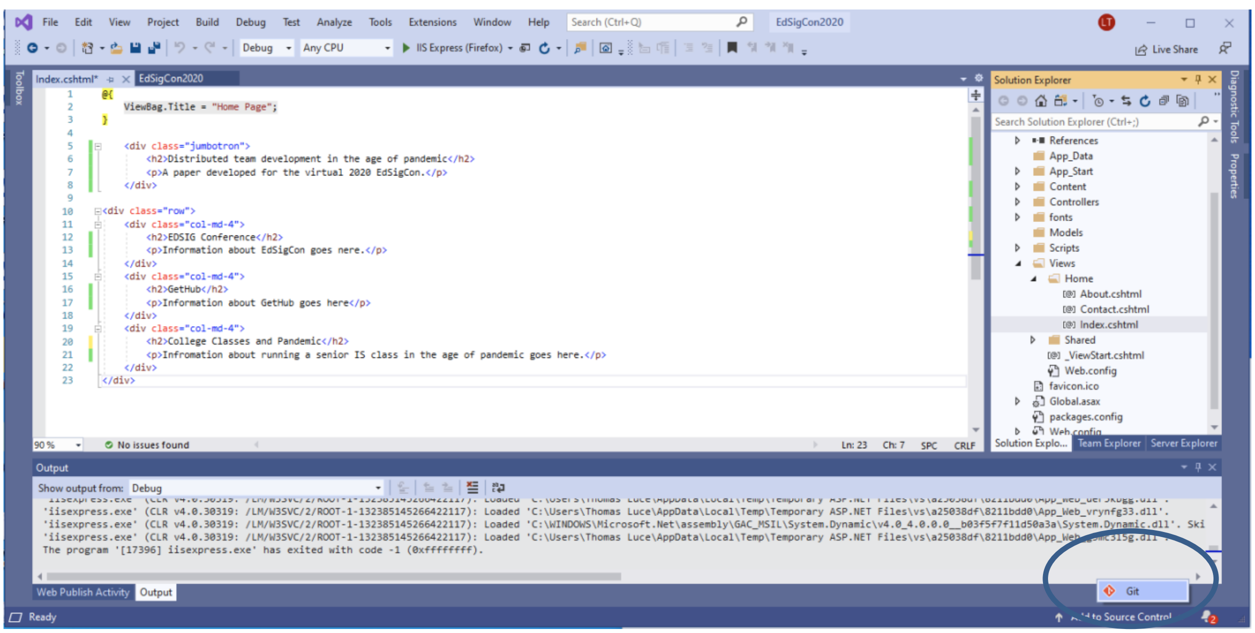
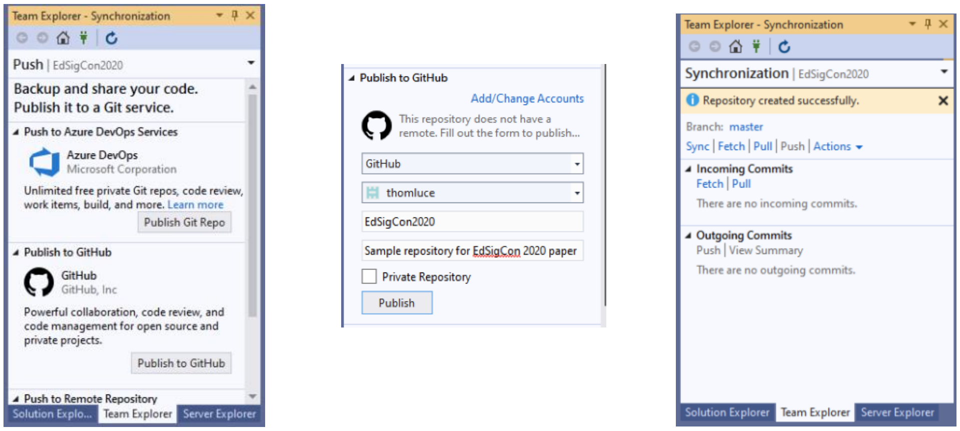
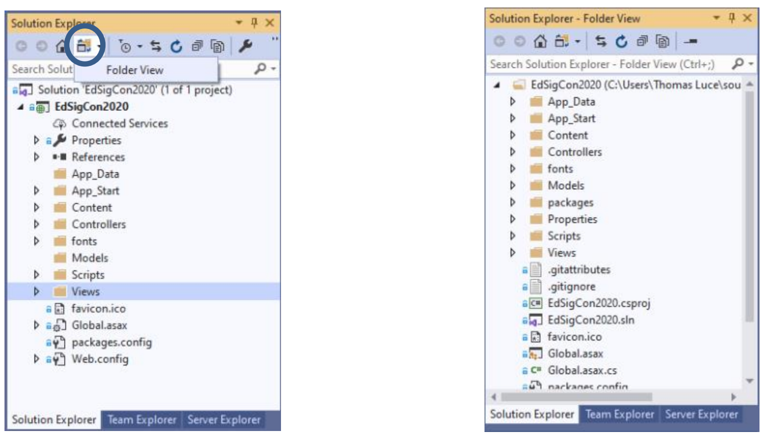


Figure 2. Visual Studio showing Add to Source Control



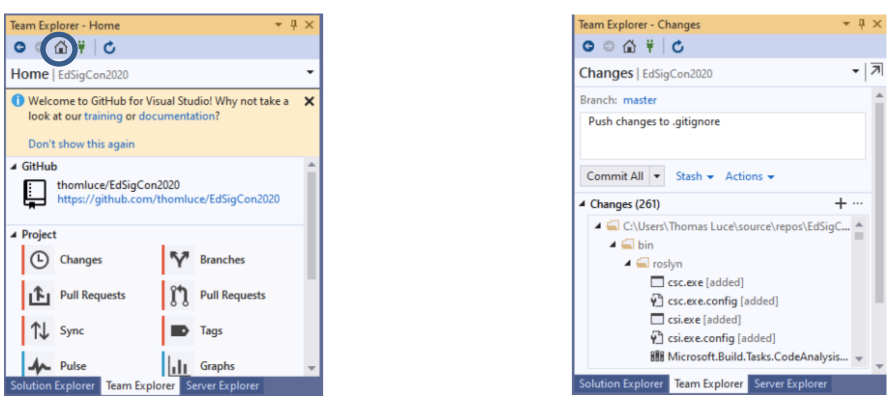
(A) (B) (C)

Figure 3. Adding GitHub for source code management.



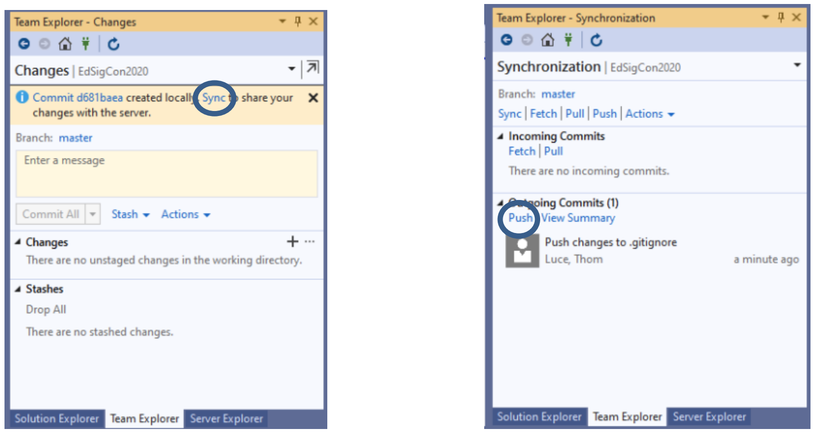
(A) (B)

Figure 4. Solution folder options.



(A) (B)

Figure 5. Team Explorer after clicking the Home icon and the window after clicking Changes



(A) (B)

Figure 6. Commit, Sync Push

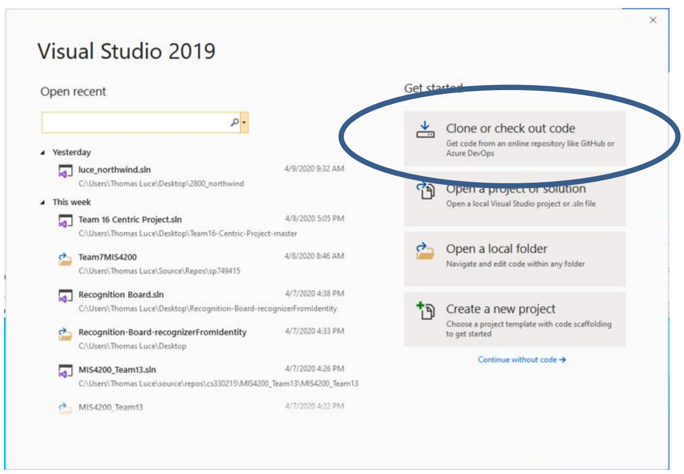


Figure 7. Visual Studio project startup

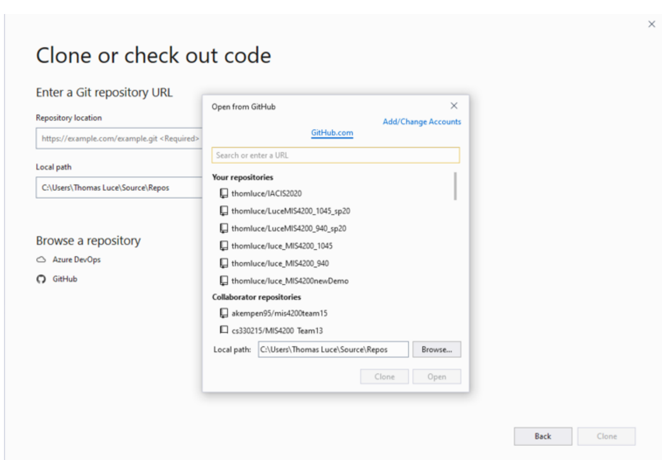
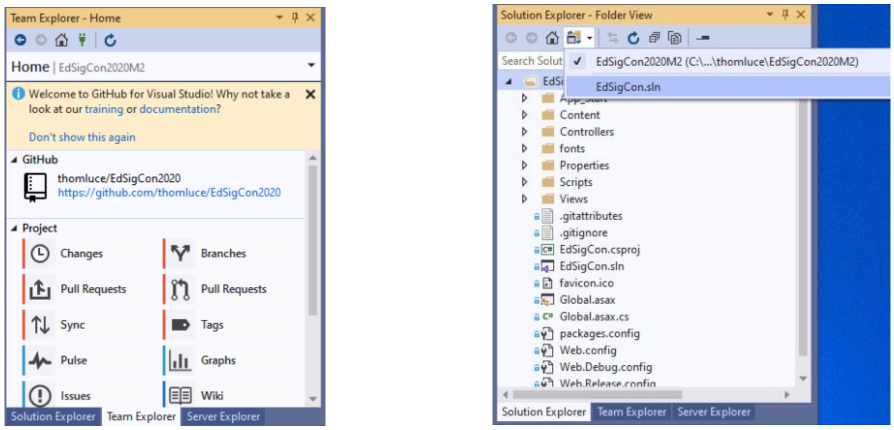
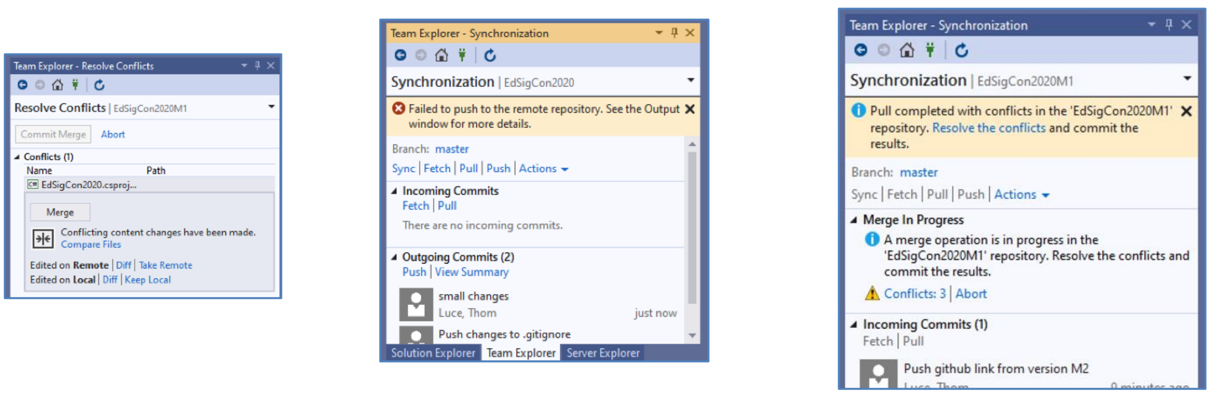


Figure 8. Clone or Check Out code window



(A) (B)

Figure 9. After cloning a project.



(A) (B) (C)

Figure 10. Conflict resolution and merging

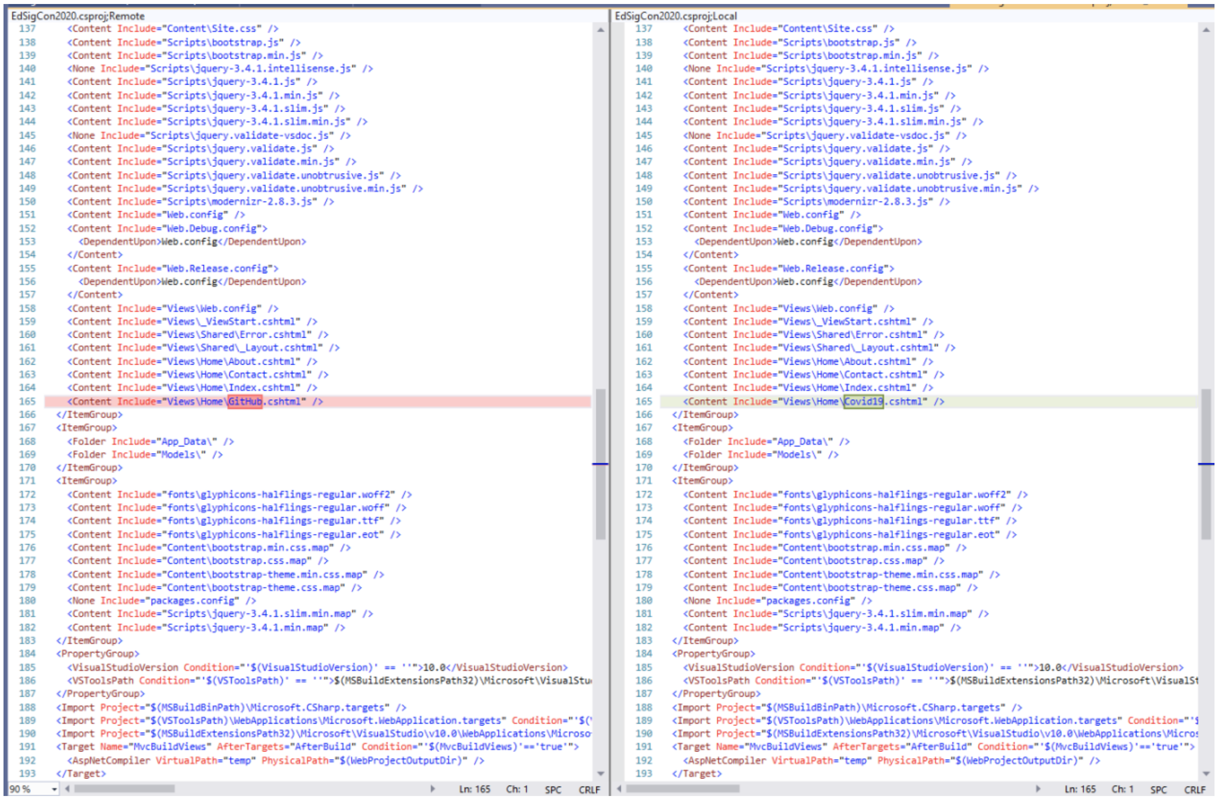


Figure 11. Comparing remote and local versions of EdSigCon.csproj

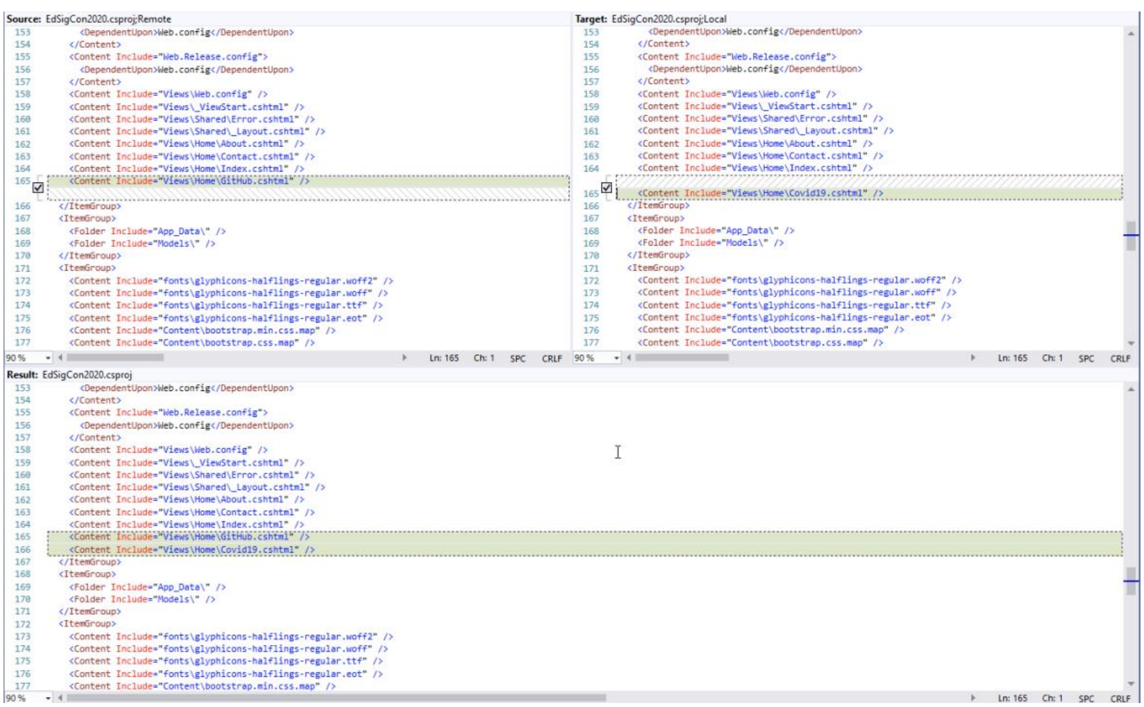


Figure 12. Merging files.

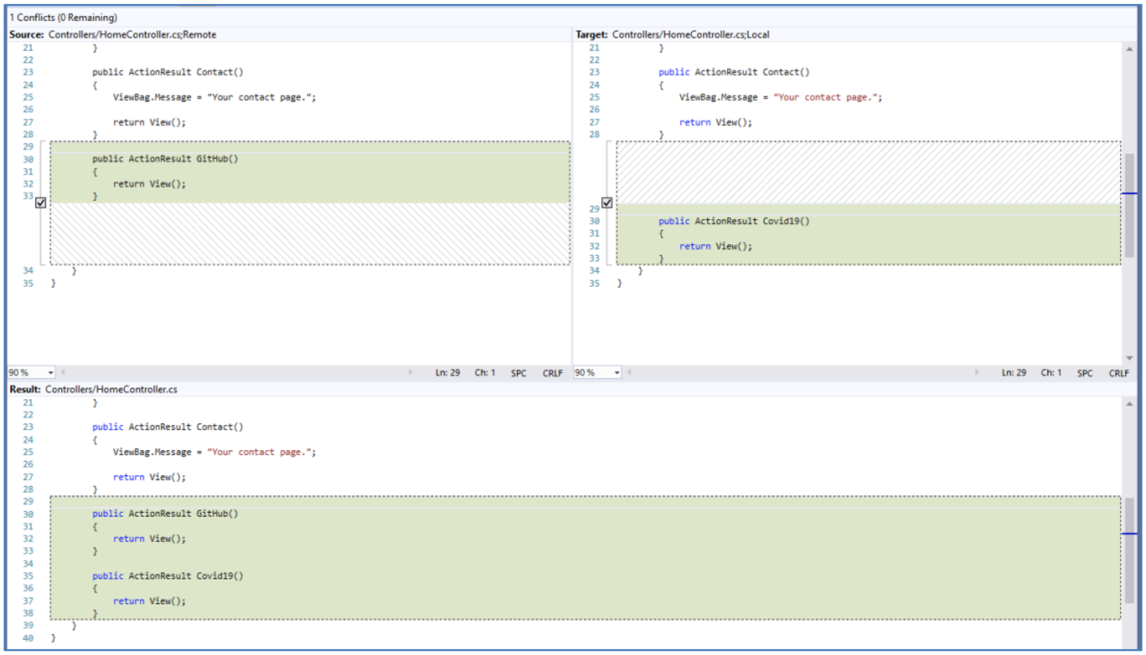


Figure 13. Merging changes in HomeController.cs

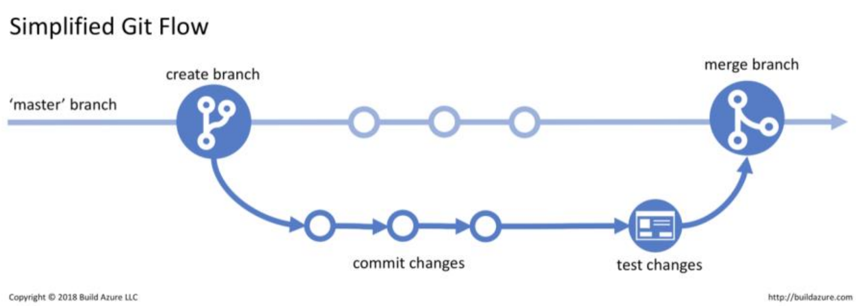
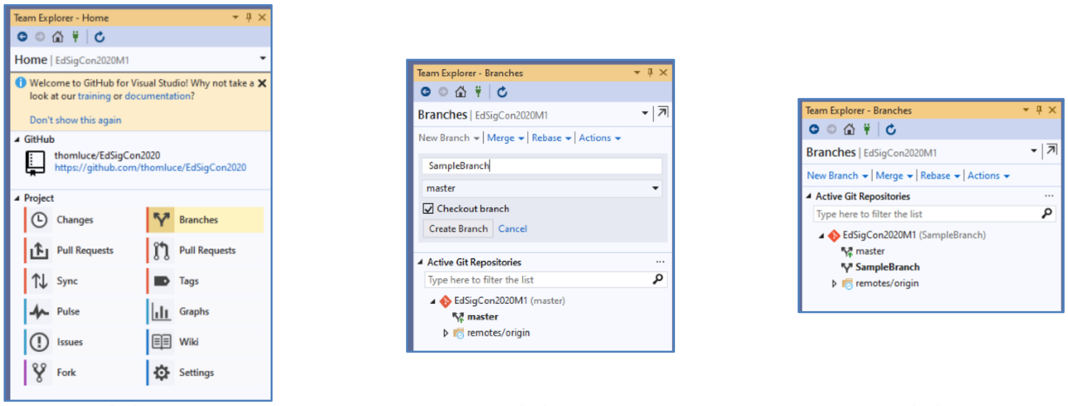


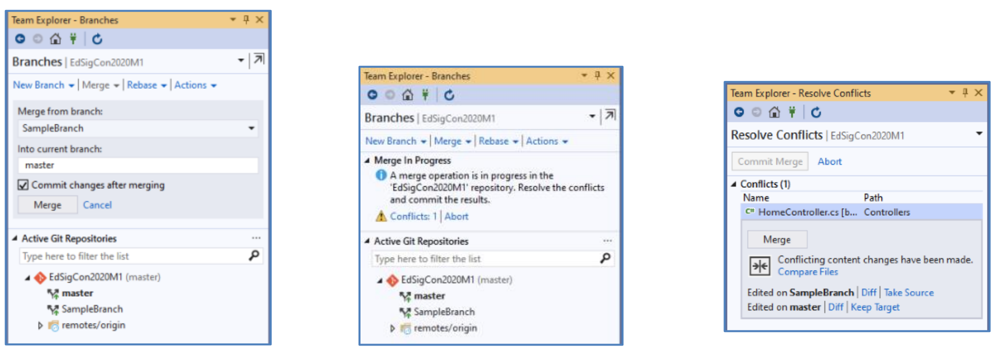
Figure 14. Simplified Git Flow (https://build5nines.com/introduction-to-git-version-control-workflow/)



(A) (B) (C)

Figure 15. Creating a new branch

Figure 16. Visual Studio with SampleBranch selected



(A) (B) (C)

Figure 17. Steps in the Merge Process

Re-Engineering General Education and the Impact on Undergraduate Technology Students

Stuart L. Wolthuis
stuart.wolthuis@byuh.edu
Faculty of Mathematics and Computing

Patricia H. Campbell
patricia.campbell@byuh.edu
Faculty of Education and Social Work

Brigham Young University-Hawaii
Laie, HI 96762

Abstract

This research shares insights into the results of a three-year process of re-engineering general education at Brigham Young University–Hawaii (BYU–Hawaii), a private university, and the impact this change has conveyed to our students, curriculum, and faculty. The authors first describe the background of traditional general education, then describe the impetus for change. This is followed by a detailed explanation of the new general education program and the resulting impact on students, specifically those perusing degrees in computer science, information systems, and information technology.

Keywords: General Education, Modular GE, Re-engineering, Technology Majors

1. INTRODUCTION

The call for innovation in general education (GE) has resounded in the halls of academia many times over the past several decades. The roots of GE, however, go back several centuries. The common core content for Harvard, founded in 1636, used Plato's classical liberal arts outline from the trivium and quadrivium published in Plato's *Republic* (O'Banion, 2016). Reengineering GE is much like the process of creating a new software program. In software engineering the existing system is identified as the "as-is" system and the new or improved system is referred to as the "to-be" system (Dennis, Wixom, & Tegarden, 2015). "To-be" GE systems are constantly emerging and since Harvard's inception Plato's outline has served as the foundation for American colleges until democratized education splintered liberal education into multiple disciplines and specialties (O'Banion, 2016).

The goal of this paper is to outline a simple approach for reengineering GE to give students a higher level of autonomy, more choices on areas of study, and more combinations with those choices.

2. IMPETUS FOR CHANGE

Change in GE has been influenced by many factors, including a 1994 U.S. Presidential report on education that emphasized a set of college graduate skills 26 years ago (Adelman, 1994). A current evaluation of the same subject reveals that employers today want skills that are completely different (Bortz, 2020), see Table 1.

Adelman also reviews the 1979 U.S. Presidential report from the President's Commission on Foreign Language and International Studies, *Strength Through Wisdom,* and emphasizes the findings, namely that the most important areas of study for a college graduate in 1994 was, in order: a global perspective, fluency in a second language, and finally, multicultural knowledge and experiences.

Bear in mind, that in 1979 the global economy and globalization was on the rise. Bortz attributes his information to the *Job Outlook 2020* survey performed by the National Association of Colleges and Employers (NACE), Table 1.

Ranking/%	College Graduate Skill
1/91.2%	Problem-solving skills
2/86.3%	Ability to work in a team
3/80.4%	Strong work ethic
4/79.4%	Analytical skills, critical thinkers
5/77.5%	Written communication skills
6/72.5%	Leadership skills
7/69.6%	Verbal communication skills
8/69.6%	Initiative
9/67.6%	Detail-oriented
10/65.7%	Technical skills
20/2.9%	Fluency in a foreign language

Table 1. Skills Employers Look for in New Grads, 2020

Employer expectations have felt a seismic shift. The first 10 skills were not on Adelman's radar in 1994, and the number 1 skill 26 years ago now appears as number 20 with a current percentage of 2.9% (included in Table 1 for comparison).

Other skills, typically on the top 10 list in the past, have also fallen: interpersonal skills, organizational ability, tactfulness, and creativity used to be seen as workplace skills that would be taught in GE classes. Why? Because academic leadership typically does not want to give up precious time in their disciplinary majors and minors to teach interpersonal skills (Koritz, 2019).

The top 10 skills, sought after by employers, are no doubt a product of our global economy. Our ability to connect with each other, whether personally, or commercially in C2B, B2B, or B2G venues has vastly improved as broadband replaced antiquated dial-up speeds in systems employing fiber optic cable, satellites, microwave and more capable switches, routers, and servers.

Background in General

This remarkable acceleration of technology has outstripped the traditional education system. The overall accessibility of information precludes the need to memorize information (Rusetskii, 2014), the standard of past education systems. Memorizing the U.S. Presidents, the capitals of each state in the U.S. or your own home country was a rite of passage for children. One of the authors of this work enjoyed an undergraduate humanities class in music and memorized the progress of Gregorian Chant to the early classics, but can't remember the sequence now or other details of non-STEM GE course material.

Foundation courses essential to STEM majors begin with basic GE courses in mathematics and science (Stieha & Shadle, 2017). Although these basic courses make perfect sense to STEM educators and hopefully students, they may not make sense to other disciplines. The GE repertoire is obviously much broader than mathematics and science; Rusetskii also notes that the crises in GE is that the body of knowledge for any specialization has increased in size and complexity and may be obsolete next semester, or even tomorrow. His third point is one that educators grapple with every semester, what should be taught now for students to be successful in the 21st century? He also notes the following points:

- Education is losing its ability to empower students in providing upward social mobility.

- The three "R's" no longer set up a person for success for the rest of their life as the world's body of knowledge is rapidly changing daily.

- And finally, students at most levels, and especially the college level, expect autonomy in the educational process to follow their interests and personal goals.

The Environment

BYU–Hawaii has over 3,000 students, 50% of them international. Approximately 29% of students are in Science Programs (including Biochemistry, Biology, Chemistry, Exercise and Sport Science, Psychology, Physical Science Education, and Physics Education); 4% are in Computing and Mathematics (Computer Science, Information Systems, Information Technology, and Mathematics, applied, pure and educational). The remainder of students are in business, arts, and humanities. BYU-Hawaii doesn't have the infrastructure for an engineering program, but does have robust programs in science, technology, and mathematics.

Benchmark GE Programs
Rusetskii's last point, autonomy, is the core driver of GE change at BYU–Hawaii. After many years of trying to tweak our GE program, the authors' university became aware of two new GE programs at other universities that provided a benchmark. Both the University of Rochester and The College of Idaho created GE curricula that simplified their GE programs as follows and let students make most of their own academic choices:

The University of Rochester created a simplified GE program with three basic parts (University of Rochester GE URL, 2020):

1. Primary writing requirement, mastery of written communication, with dozens of choices.

2. Course requirements for the major, three broad areas to choose: Humanities, Social Sciences, or Natural Sciences and Engineering.

3. Cluster in two other areas:
   - Politics and math
   - Sonic arts and technology
   - Power and inequality

The College of Idaho fashioned a similar approach requiring students to graduate with 1 major and 3 minors, in 4 years.

1. This program is called PEAK and offers 24 majors and 40 minors allowing students to pursue their "passion".

2. The categories, or "Peaks" include the humanities, social sciences, natural sciences, and a professional field.

3. GE MODULAR DEVELOPMENT

In June 2016 a Modular GE committee was led by a dean and formed of all department chairs, 17 educators, who had all experienced GE firsthand as students and teachers. Our charge was to "craft a proposal for a GE approach that can offer students both breadth and depth of education to help them be both more prepared and more marketable as they move beyond the University."

As this diverse group met each month for over a year, we discussed and crafted a new modular GE program that now has the name Holokai and was implemented Fall of 2018. Holokai, is a Hawaiian name meaning voyage, specifically: "Pacific Islanders sailed to new lands in voyaging canoes using the stars and waves for navigation. The Hawaiians call this voyage Holokai. (kai = ocean, holo = to go, to move, to travel)." (Holokai website, 2020).

Two specific charges the committee was asked to address were student autonomy and allowing students to follow their passion.

Early in this process the authors were reminded that many of our students come to our university with a directive from their parents on what major they should pursue.

The new modular GE program, Holokai, allows a student the autonomy to select their own journey; they could still major in computer science for the family business, but they could follow their passion and select a minor in film and a minor in entrepreneurship, see Figure 1.

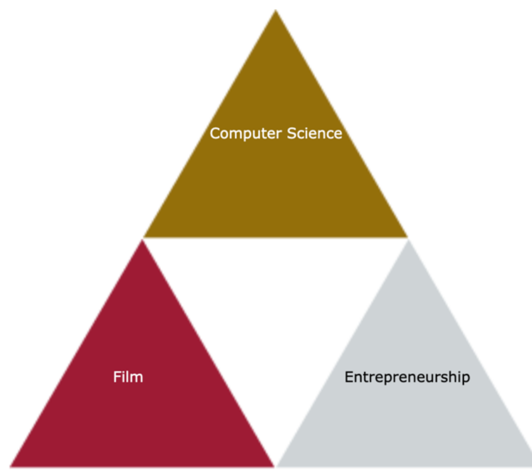


Figure 1. Sample Holokai

This graphic represents the majors, minors, and certificates in three categories:
- Crimson represents Arts & Humanities
- Gold signifies Math & Sciences
- Silver stands for Professional Studies
- The White triangle stands for the Religion, Math and English Core.

Details for each of these categories appear on the Holokai website, this also acts as an application that allows a student to graphically setup their plan (Figure 2 and 3 in the appendices shows the student website for creating a Holokai).

Before this significant GE change, the number of minors was very small in the technology area at BYU–Hawaii. Each major (CS, IS, and IT) had an *existing minor.

To meet the needs of many more students seeking options for minors under the Holokai, the Computer & Information Sciences Department created 10 new minors/certificates in the "Professional" and "Math and Sciences" categories. Our excitement to create these new options has not matched the expected traction, and those marked with a "^" may be suspended, Table 2.

Professional	Cr Hrs
- Agile Project Mgt Cert	15
- Digital Business Cert	14
- Digital Security Cert	18^
- Digital Security Minor	12^
- Digital Tech. Minor	18
- Enterprise Business Systems Minor	15
- Information Sys. Minor	18*
- Intro. to Digital Tech. Minor	12^
- Intro. to Mobile App Development Minor	12
- Intro. Web Des. Minor	12
Arts and Humanities	
- none	
Math and Sciences	
- Comp. Science Minor	18*
- Information Tech. Minor	18*
- Web Development Minor	18^

Table 2. Minors and Certificates Created for New GE Program

Another consideration that was a common discussion by the Modular GE Committee was what will happen to GE service courses currently part of the as-is GE Program? Time has proven that these courses were replaced or redesigned to meet the needs of the to-be GE Program with a zero-sum gain.

Another programmatic area of concern was the need to assure the students received both a depth and breadth of experience that was the impetus of previous traditional GE programs. This is inherently solved with the categories designed in the Holokai and the requirement that a student's major and two minors must be selected from different categories: Professional, Arts & Humanities, Math & Sciences, see Table 3 in the appendices for details.

All students are required to also complete 14 credit hours of religious education, selecting from 23 courses; students also are required to complete core classes in Math and English. The remainder of a student's academic program is their choice, or their Holokai with the following core Math and English choices:

Mathematics-Quantitative and Logical Reasoning (3-5 credit hours required) allow the following choices:

- Statistics: MATH 121, BIOL 340, PSYC 205, POSC 300 & POSC 304, or EXS 339
- Calculus: MATH 212, MATH 213, or MATH 119.
- Trigonometry: MATH 111
- Quantitative Reasoning: MATH 107
- Discrete Mathematics: CIS 205

For Reading/Writing/Speaking, the following choices are provided to students:

- Reading/Writing/Speaking (6 credit hours required, 3 basic, 3 advanced).
- ENGL 101 Communication in Writing, Speaking, and Reading (3) and,
- Advanced Writing (3 credit hours).

The advanced writing requirement can be completed by taking one of the following: ENGL 314 Exposition and Analysis in the Humanities (3) or ENGL 315 Topics for Advanced Writing and Analysis (3) or ENGL 316 Technical Writing (3) or an Approved Senior Seminar in the major: CHEM 494, HIST 490, CS 491-493, IT 491-493

4. MOVING FORWARD

The University GE Program is in a transition period at this time. A resounding majority of students have chosen the new to-be Modular GE Program (Holokai) over the old as-is GE Program, 2,492 of 3,176 = 78% of our students.

Guiding Students on Their Holokai

BYU–Hawaii is staffed with an excellent cadre of 7 advisors who assist our students in all things related to their academic progress. This includes initial new student orientation and matriculation when the advisors help new students to create their academic plan. One advisor summarized the reasons why our students receive the Holokai with enthusiasm.

Why Students Choose the Holokai

The majority of students have embraced the Holokai Program due to the following reasons:

- They graduate with a minor or certificate that is beneficial for post-graduation and career goals and opportunities.

- Traditionally students had to take a prescribed set of general education requirements, now with the Holokai they have options with minors and certificates that allow them the breadth and depth to explore those minors and certificates to complement their learning and other ways of knowing based on their interest and personality.

- Students have a choice and a variety of different options for their education in the major they choose, and the minor/certificates span three disciplines: arts/humanities, math and sciences, and professional.

– This allows students a diverse set of choices in their chosen fields and to expand their vision and perspective of different areas and epistemologies.

- Additionally, students feel that when they graduate, they have the additional skill sets and knowledge which will potentially help them in the future workforce.

- Many of them will also complete additional minors and or certificates to increase their marketability, additional skills that will help them to successfully move beyond the university.

5. CONCLUSIONS

The new Holokai GE program empowers students with the autonomy they seek to control their academic studies which supports Rusetskii's premise that students prefer the ability to plan and pursue their own interests and personal goals.

The Holokai GE program also allows students to build a personal study program that provides breadth with a major and two minors in different categories: arts/humanities, math and sciences, and professional.

Students have shared with their academic advisors that they are very satisfied with the Holokai program. Each semester, lessons are learned and applied to improve the system, the courses, and the course offerings.

Concerns about losing programs, FTEs, or enrollment have been unfounded. In fact, in the technology area we have seen a steady enrollment increase of 7% per year for the past three years and are on mark to hire a new faculty FTE. Three adjuncts currently assist in teaching technology classes to meet the enrollment increases.

Planned future research and development activities include:

1. Formalize the process of gathering information from students about their Holokai experience, graduate surveys will likely be modified as our first group of students, who have experienced the Holokai from the beginning of their undergraduate program, graduate in 2020.

2. Continue to review and fine-tune minors and other programs to meet our ever-changing world of technology, information, and employer requirements.

6. REFERENCES

Adelman, C. (1994). What Employers Expect of College Graduates: International Knowledge and Second Language Skills. Office of Research, Education Research Report. Retrieved 29 June 2020 from https://www2.ed.gov/pubs/OR/ResearchRpts/employ.html

Brigham Young University–Hawaii (2020). GE Curriculum. Retrieved 29 June 2020 from https://Holokai.byuh.edu/

Bortz, D. (2020). Skills Employers Look for in College Graduates. Retrieved 29 June 2020 from https://www.monster.com/career-advice/article/5-skills-employers-want-in-new-grads-and-arent-finding

Dennis, A., Wixom, B. H., & Tegarden, D. (2015). Systems analysis and design: An object-oriented approach with UML. John Wiley & sons.

Job Outlook (2020). Key Attributes Employers Want to see on Student's Resumes. Retrieved 29 June 2020 from https://www.naceweb.org/talent-acquisition/candidate-selection/key-attributes-employers-want-to-see-on-students-resumes/

Koritz, A. (2019). Introduction: The Public Purposes of General Education. The Journal of

General Education, 67(3-4), 173-177. doi:10.5325/jgeneeduc.67.3-4.0173

O'Banion, T. (2016). A brief history of general education. Community College Journal of Research and Practice, 40(4), 327-334.

Rusetskii, V.F. (2014) Russian Education & Society. Nov2014, Vol. 56 Issue 11, p41-54. 14p. doi: 10.2753/RES1060-9393561104.

Stieha, V., Shadle, S. E., & Paterson, S. (2017). Stirring the Pot: Supporting and Challenging General Education Science, Technology, Engineering, and Mathematics Faculty to Change Teaching and Assessment Practice. The Journal of General Education, 65(2), 85–109.

The College of Idaho (2020). GE Curriculum. Retrieved 29 June 2020 from https://www.collegeofidaho.edu/academics/peak/curriculum

University of Rochester (2020) GE Curriculum. Retrieved 29 June 2020 from https://www.rochester.edu/about/curriculum.html

Appendices

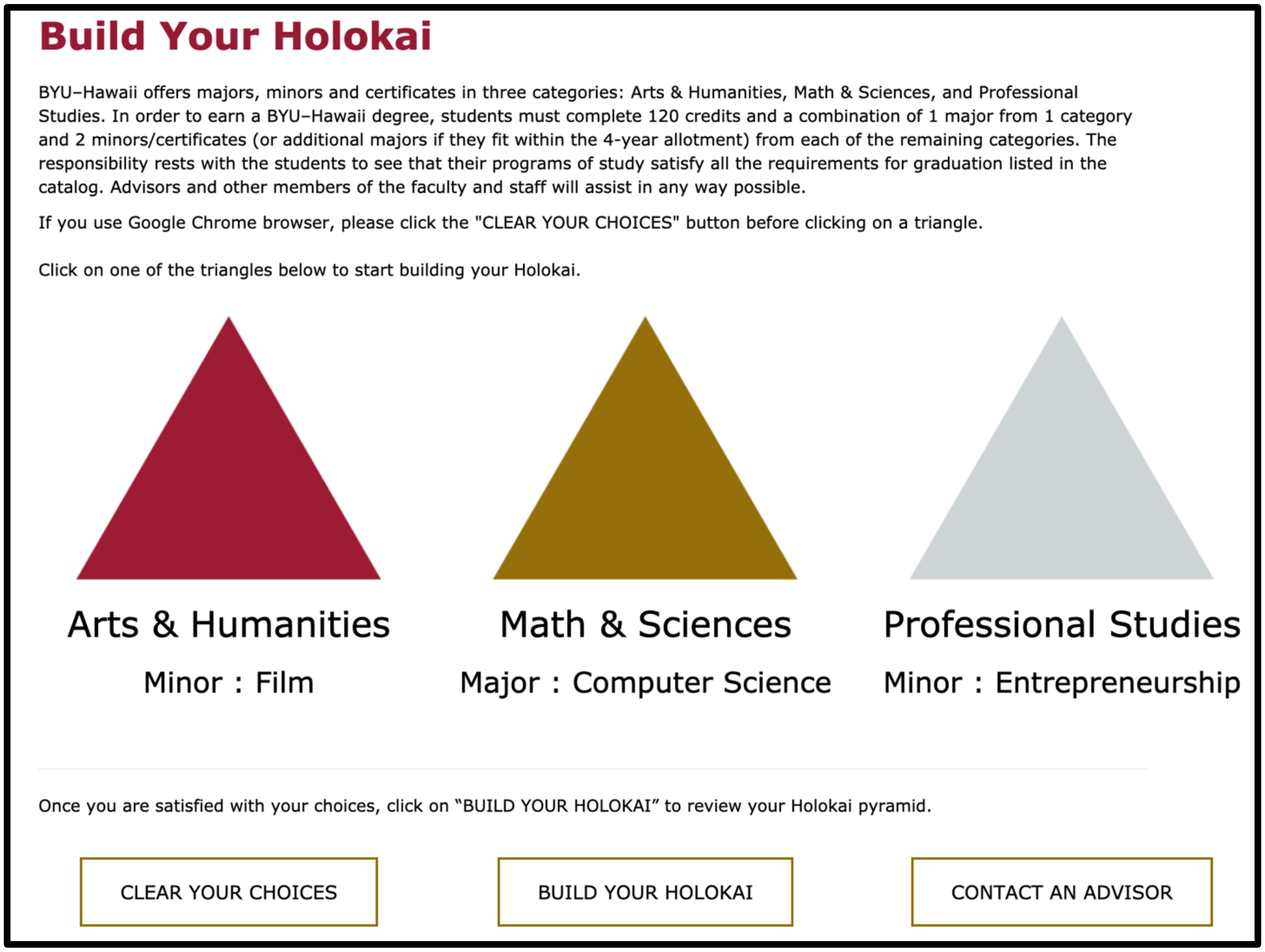


Figure 2: Student Tool, Build Your Holokai (Holokai Website)

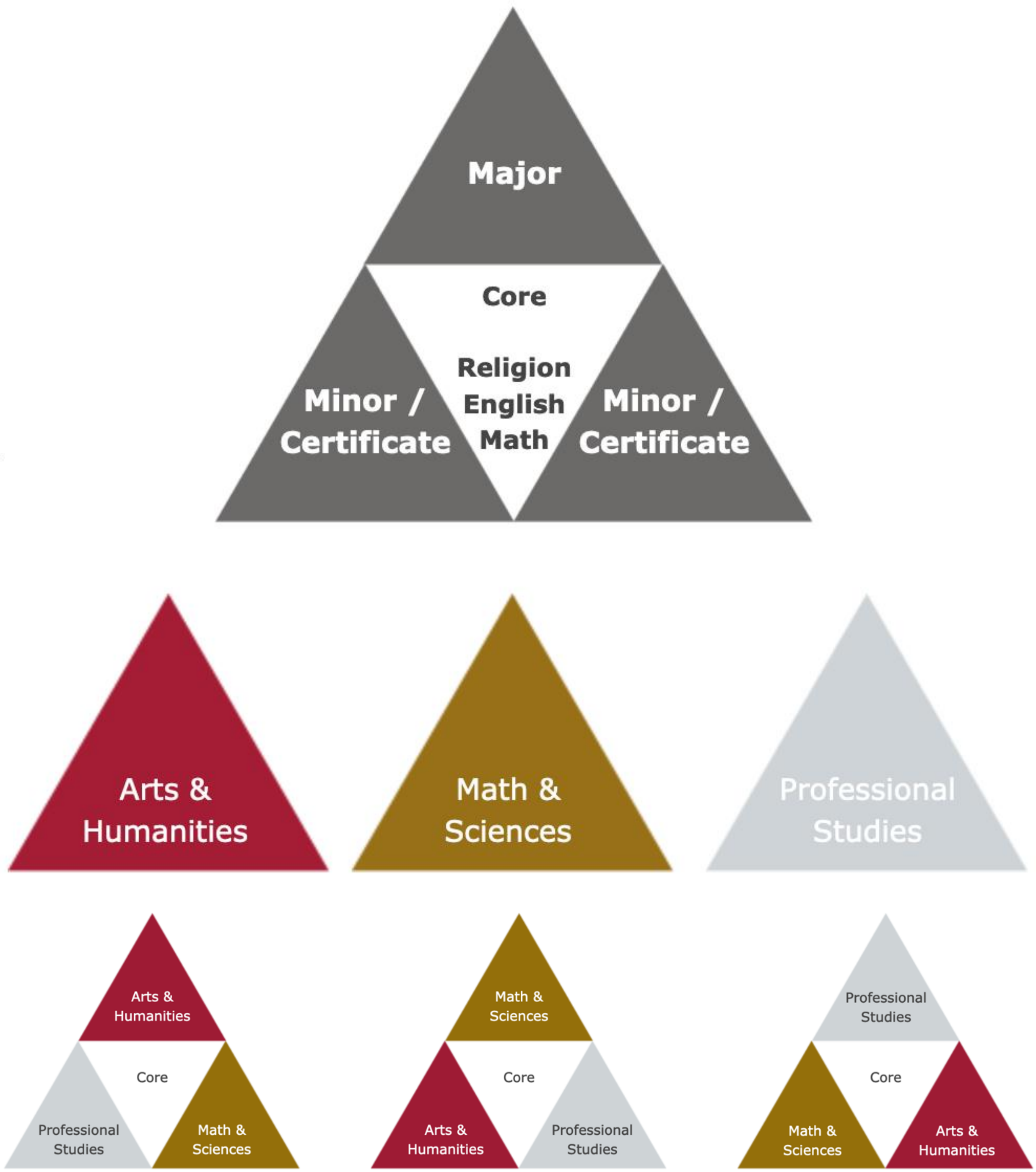


Figure 3: Holokai Category & Combinations (Holokai Website)

Arts & Humanities	Math & Sciences	Professional Studies
Art Education Major	Biochemistry Major	Accounting Major
Asian Studies Minor	Biochemistry Minor	Accounting Minor
Chinese Minor	Biochemistry Major – Environmental Science Emphasis	Agile Project Management Certificate
Communication Major	Biochemistry – Neuroscience Emphasis	Business Management Major – Economics
Communication Media & Culture Major	Biology Major	Business Management Major – Finance
Creative Writing Major	Biology Minor	Business Management Major – Human Resources & Org.
Cultural Anthropology Major	Biology Major – Ecol, Evol, & Pop	Business Management Major – Marketing
Cultural Anthropology Minor	Biology – Marine	Business Management Major – Supply Chain, Operations, and Analytics Concentration
English Minor	Biology – Molecular Biology	Coaching Minor
English Major – Creative Writing	Biology Education Major	Criminal Justice Certificate
English Major – Literature	Chemistry Minor	Digital Business Certificate
English Major – World Literature	Chemistry Education Major – US Teaching Certificate	Digital Security Certificate
English Major – Writing	Computer Science Major	Digital Security Minor
English Education Major	Computer Science Minor	Digital Technology Minor
Film Minor	Exercise & Sport Science Major – Biomedical Science	Economics Minor
Foundational Language Study Minor	Exercise & Sport Science Major – Health & Human Performance	Education Minor
Hawaiian Language Certificate	Exercise & Sport Science Major – Sports & Wellness Management	Elementary Education Major
Hawaiian Language Minor	Exercise & Sport Science Major – US Certification	Emergency Management Certificate
Hawaiian Studies Major	Health & Human Performance Minor	Enterprise Business Systems Minor
Hawaiian Studies Minor	Information Technology Major	Entrepreneurship Certificate
History Major	Information Technology Minor	Entrepreneurship Minor
History Minor	Intro Conservation Biology Minor	Governance Certificate
History Education Major	Intro to Chemistry Minor	Hospitality & Tourism Mgt Major
Integrated Humanities Major	Intro to Marine Biology Minor	Hospitality & Tourism Minor
Integrated Humanities Minor	Intro to Mathematics Minor	Human Resources Mgt Minor
Intercultural Peacebuilding Major	Intro to Natural Science Minor	Information Systems Major

Arts & Humanities	Math & Sciences	Professional Studies
Intro to Linguistics Minor	Intro to Nutritional Science Minor	Information Systems Minor
Japanese Minor	Intro to Physics Minor	Intercultural Peacebuilding Certificate
Linguistics Minor	Mathematics Minor	International Development Certificate
Music Minor	Mathematics Major – Applied Mathematics	Intro to Digital Technology Minor
Music Major – General Music	Mathematics Major – Pure Mathematics	Intro to Mobil App Development Minor
Music Major – Instrumental Performance	Mathematics Education Major	Intro to Social Work Minor
Music Major – Piano	Physical Science Education Major	Intro to Web Design Minor
Music Major – Vocal Performance	Physics Education Major	Intro to TESOL Minor
Music Major – World Music Studies	Political Science Major	Leadership Minor
Pacific Islands Studies Minor	Political Science Minor	Legal Studies Certificate
Pacific Islands Studies Major	Psychology Minor	Marketing Minor
Painting Minor	Psychology Major – Clinical/Counseling	Organizational Behavior Minor
Piano Performance Minor	Psychology Major – General/Experimental	Professional Writing Minor
Sculpture Minor		Public Management Minor
Spanish Minor		Social Work Major
TESOL Major		Social Work Minor
TESOL Education Major		Sports & Wellness Mgt Minor
Theatre Minor		Supply Chain, Operations & Analytics Minor
Visual Arts Minor		TESOL Certificate
Visual Arts Major – Visual Arts – Painting		
Visual Arts Major – Visual Arts Graphic Design		
Visual Arts BFA Major – VA Graphic Design BFA		
Visual Arts BFA Major – Visual Arts – Paint Fine Arts		

Table 3: Categories with majors/minors/certifications offered for the Holokai GE Program (Holokai Website)