Journal of
Information
Systems
Education

Volume 35
Issue 1
Winter 2024

Teaching Tip
Leveraging Learning Strategies at Scale – Big and Small Changes in a Big IS Course

Steven Sclarow, AJ Raven, and Mart Doyle

Recommended Citation: Sclarow, S., Raven, A. J., & Doyle, M. (2024). Teaching Tip: Leveraging Learning Strategies at Scale – Big and Small Changes in a Big IS Course. *Journal of Information Systems Education*, 35(1), 1-13. https://doi.org/10.62273/FLSR7630

Article Link: https://jise.org/Volume35/n1/JISE2024v35n1pp1-13.html

Received: June 30, 2022
First Decision: August 26, 2022
Accepted: May 8, 2023
Published: March 15, 2024

ISSN: 2574-3872 (Online) 1055-3096 (Print)

Journal of Information Systems Education, 35(1), 1-13, Winter 2024
https://doi.org/10.62273/FLSR7630

Teaching Tip
Leveraging Learning Strategies at Scale – Big and Small Changes in a Big IS Course

Steven Sclarow
Fox School of Business
Temple University
Philadelphia, PA 19122, USA
sclarow@temple.edu

AJ Raven
LeBow College of Business
Drexel University
Philadelphia, PA 19104, USA
aj.raven@drexel.edu

Mart Doyle
Fox School of Business
Temple University
Philadelphia, PA 19122, USA
mdoyle@temple.edu

ABSTRACT

This paper presents field-tested improvements over an 11-year period of a large-scale *Introduction to Information Systems* core business school course and provides a framework for implementation. Engagement and learning in large-scale courses can prove challenging, especially when the class is a requirement within a business school's core curriculum where most students taking the class are not majors in the course's subject matter. The course sections typically have 100 or more students each. The non-IS major student population attends this core course because they must, rather than want to, attend which in most cases leads to indifference and disengagement. A primary need and challenge of a required core course are to create a compelling 14-week *Introduction to Information Systems* course that captures this student population's interest in an engaging manner. This paper identifies an innovative approach to create and foster an enhanced learning environment organized around active learning, dynamic content, and a narrative structure. Major changes discussed in the paper are the abandonment of textbooks, the introduction of in-class activities and JavaScript coding, and the establishment of a narrative.

Keywords: Active learning, Information systems education, Large classes, Narratives

1. INTRODUCTION

Digital Systems is a required information systems core course, taught to over 1,600 Temple University School of Business students each academic year, in many sections of about 100 students each. This course is part of the Business School's core curriculum, and acts as a gateway for students to become MIS majors and minors. Students tend to be sophomores and juniors, but some are freshmen and seniors. This introductory course to Management Information Systems (MIS) enables students to thrive in today's digital world. They learn to apply technology, utilizing course concepts in their field of study, and develop critical thinking skills to solve today's business problems while planning for problems that will exist in tomorrow's digital world. The course learning goals are:

- Learn to apply a core body of MIS-specific knowledge to business situations and problems.
- Develop critical thinking skills through the analysis of business processes and the MIS systems that support them.
- Develop quantitative reasoning skills by assessing the impact investments in MIS systems have on a business.
- Develop visual communications skills while learning to model business processes, the information required to perform these processes, and the systems that support these processes.

The course goals, overall learning outcomes, and weekly learning objectives are aligned with the Business School's Bachelor of Business Administration's (BBA) Learning Goals and Strategic plan. They are implemented utilizing our delivery framework which was conceptualized as the "Active Learning Funnel" incorporating three significant dimensions: Technology, Content, and Activity that form an Enhanced Learning Environment (Lavin et al., 2018). The typical course work-week utilizes this framework and consists of a 50-minute classroom discussion followed by two In-Class Activities (ICAs) that take up to 50 minutes. The ICAs are designed to build upon and complement the week's topics and learning objectives.

The four main learning goals of the BBA program are like those of the course: 1) Demonstrate business knowledge needed to make business decisions, 2) Apply critical thinking skills to business problems, 3) Apply quantitative reasoning skills to make recommendations and business decisions, and 4) Apply effective business communication techniques and business situations; all are integrated within our weekly discussions. Students are encouraged to demonstrate their business knowledge and engage in open discussion with the instructor and their peers, evaluating business scenarios, challenges, and opportunities.

These discussions provide students with opportunities to explore innovative ideas, make recommendations, and apply a variety of business communication skills. The ICAs enable students to apply their newfound MIS knowledge with their own discipline-specific knowledge plus any pre-existing industry experience to solve business problems. ICA topics range from the development of business process diagrams to the analysis of digital tools and culminate in the exploration of programming concepts where students utilize critical thinking skills to evaluate and solve business problems.

The Digital Systems course shares the Business School's values, too, and incorporates them throughout the course goals and objectives. Our flipped classroom environment is just one of many examples where we align with the school's mission. Students collaborate during ICAs, forming breakout groups where they apply course concepts, analyze business challenges, and develop solutions with the common goal of achieving course learning objectives. These ICAs foster an environment of diversity and inclusion, empowering students to learn how to utilize new tools and resources. They encourage innovation, as student teams generate a variety of solutions that are shared with their teammates and the class, fostering growth, learning, and evaluation of new ideas and approaches to solve today's business challenges.

Survival in today's digital world requires that every business professional understands technology; however, understanding alone is not enough. Students must also know how to leverage technology to create business value, think critically, and create tools and solutions to solve today's business problems and those that will exist in the future. Management Information Systems is a key enabler of business innovation. This Digital Systems course explores the systems that organizations use to solve their business problems including creating and/or implementing digital products, including the platforms that these systems are built upon and the API ecosystem by which systems extend their reach and capability. The course explores defining Data Analytics and Big Data, including discussion of how data creates value for organizations. It also considers cyber-security, including the risks and responses that surround digital products. Finally, in this age of increasing use of software, students are introduced to the creation of software by learning the basics of programming in JavaScript.

The initial iteration of the course was first introduced in the Fall of 2011 and was built on a traditional textbook-based framework. Doyle et al. (2015), in their discussion of studio classes in MIS education, refer to this iteration as a "*course in a box*," where all the reading materials, exam questions, assignments, and slide decks are provided by a textbook publisher. Years of large lecture-hall instruction following this course-in-a-box approach have fostered a passive learning environment. Students were observed to become worker drones, sifting through massive amounts of content, memorizing and regurgitating terms and definitions, which many times were forgotten once the lecture-hall door closed behind them.

The MIS team responsible for creating this course identified three primary challenges: first, the need to improve student's engagement (Carini et al., 2006); second, the need to keep the course content relevant; and, finally, the need for a common thread to connect concepts across the course. These needs were determined from student feedback, faculty feedback, and department/school committee reviews. Effectively addressing these three primary challenges and establishing the most effective learning environment occurred over an eleven-year period where innovative approaches were piloted, vetted, and when successful woven into a new teaching approach that incorporates three key themes: active learning, dynamic content, and narrative. The three themes are identified in the paper through icons. Active learning: ▲ Dynamic Content: ⚙ Narrative: 📖 The merging of these three themes into one cohesive approach is our innovation. This Teaching Tip provides a scaffolding that could be applied to any IS and/or business school core curriculum course.

2. THE COURSE

The course framework is discussed through the lens of the three themes: Active Learning, Dynamic Content, and Narrative. This framework is then summarized in the form of an Implementation Guide that will enable the reader to replicate the course development.

The course as designed is delivered as a thrice-weekly 50-minute class, a twice-weekly 80-minute class, or a once-a-week 115-minute class. It is offered synchronously, in-person, or online.

2.1 Theory Base

▲ **Active Learning:** The learning skills of Bloom's revised taxonomy (Anderson & Krathwohl, 2001; Bloom, 1956; Forehand, 2010) were used by the authors to evaluate the three needs identified above and the current course structure and contents. The evaluation showed that students were only using their lower-level learning skills: remembering and understanding. Ideally, they would also apply the higher-level skills of applying, analyzing, evaluating, and creating. Lavin et al. (2018) identified how a core course could be redesigned with a focus on using active learning principles, reinforced by a flipped classroom (Foldnes, 2016; Mok, 2014; Sandrone et al., 2020)

Bonwell and Eison (1991) define strategies to promote active learning as "…instructional activities involving students in doing things and thinking about what they are doing…" (p. iii). Furthermore, "They must read, write, discuss, or be engaged in solving problems. Most important, to be actively involved, students must engage in such higher-order tasks as analysis, synthesis, and evaluation" (p. iii). Since their work was first published, the practice of active learning has evolved and expanded to many fields, including information systems (Riordan et al., 2017; Romanow et al., 2020; Woods, 2020). Prince (2004) defined active learning as "any instructional method that engages students in the learning process. In short, active learning requires students to do meaningful learning activities and think about what they are doing" (p. 223).

Prince (2004) discussed how meta-studies on active learning can show inconsistent results, and how this apparent inconsistency may be the result of lumping diverse types of active learning together. Moreover, Freeman et al. (2014) reviewed 225 studies in a meta-analysis and noted a strong positive impact of active learning. The performance of students in active learning classes was nearly half a standard deviation higher than those in traditional lecturing classes (Freeman et al., 2014).

A key structure for active learning in the class is formed by the in-class activities. Typical weeks include two ICAs, which are designed to complement the weekly readings, videos, and class discussions and to reinforce course learning objectives. Each activity includes an instructor's PowerPoint to guide the activity and a corresponding LMS "deliverable" that the students complete in-class. ICAs are not announced prior to class meetings to improve attendance. Assignments are utilized as homework, and they are designed to enable the students to apply the concepts learned from class on their own and/or in teams and to enhance their critical thinking skills.

Active learning is often tied in with the concept of the "flipped classroom" (Mok, 2014; Sandrone et al., 2020). Students attend lectures in class and do the homework activities at home in traditional courses. In a flipped classroom, the students watch videos, read articles, and review slide decks on their own time (Bergmann & Sams, 2012). Reading articles, reviewing slide decks, and watching videos before a class does not come naturally for many students. Flipping the classroom requires more discipline from students, and more encouragement, engagement, and coaching from faculty. Students are expected to do the readings and watch videos before coming to class in the Digital Systems course.

Max Labs, created by Tim Hill at San Jose State University utilized a storified learning-model to specifically address the learning needs of iGen students who spent their entire adolescence with smart phones. Max Labs is a hands-on learning project that taps into iGen students' inherent interest in stories and social media. The project follows a fictional female student, Max, through her blog as she documents her progress in developing CRM tools for her Silicon Valley client, Riley, using *Salesforce* in a cloud computing environment. The lab has students set up a database, create a phone app, normalize the database, and set up workflow automation, while also getting experience with *Salesforce* (Hill & Nance, 2016). Students work on the Max Labs project individually. The Max Labs project is a powerful illustration of active learning since students apply concepts from customer relationship management and systems development to an actual information system. The project takes them through all the steps, but is sufficiently challenging to keep it interesting and has a working system as a deliverable.

The introduction of the cloud-based Max Labs CRM assignment coincided with the MIS department's ongoing assessment of industry trends and feedback from their IT Advisory Board; most notably, a change in approach in application design and development from "make" to "buy," building only what cannot be sourced elsewhere, and moving to server-less architectures and web APIs.

The MIS department set out to redefine its entire undergraduate curriculum based on the idea that it should teach how to use, design, architect, and manage API-centric applications in organizations that apply analytics, cloud, cyber-security, agile, and user experience. This redefining process brought about a paradigm shift in the MIS department's undergraduate curriculum map and established new scaffolding across all MIS courses. The precept helps students differentiate themselves in the marketplace. For example, if they can show that they have even basic experience with *Salesforce* or JavaScript, then this gives them a competitive advantage as they search for their first job.

Dynamic Content: A significant problem with textbooks, especially in an Information Systems (IS) course, is that the material is often outdated by the time a book is placed in the hands of students. The materials in a book must be written, edited, and then printed, a process that can take years. IS are changing at an ever-increasing pace, and books do not seem capable of keeping up. Some fundamentals of a college course may not change; however, the conception of what should be considered fundamental to an IS course must evolve to keep pace with industry changes.

A second problem with textbooks is that they provide overly simplified views of the subject matter. The authors tend to distill their knowledge into unambiguous definitions and descriptions. Many students indicate that they prefer this approach as it lends to basic memorization and regurgitation when it comes to test time. However, in academia and industry, there are many varying and evolving views regarding IS, and application approaches range from organization to organization. In our Digital Systems class, the traditional textbook was replaced with a collection of articles that are available over the Internet to address these problems. This collection of articles focuses on the core learning objective covered each week. There are typically three-to-five articles assigned each week. Most of the articles are short (two-to-four-minute reads) and are very engaging.

Students read articles from sources like *CIO Magazine* and *Wired Magazine* (Kuang, 2015; White, 2021) and come prepared to class to discuss their insights and observations. These articles are found through searches and word-of-mouth conversations and are relatable to the students regardless of their major or prior work experience. Even though students still prefer printed text over digital reading, most student reading is now performed digitally (Hargreaves et al., 2022).

Students love the idea of not having to pay for a textbook during the first week of the semester; however, by the third week of the semester, a handful of students express that their wishes for a textbook to revert to a passive learning approach. Change is challenging for many students. A textbook is written by a single author or a small group of authors with the undergraduate college student as the intended audience. Editors and reviewers make sure that all the pieces of the puzzle are covered clearly in a well-organized manner, where information from one part of the text never contradicts another part of the text. The authors, editors, and reviewers synthesize the topics, so the reader does not have to; however, this is NOT true with a collection of readings from the Internet where there are many different authors with different "voices" and a wide range of audiences. These articles are independent of one another and do not have the goal of educating undergraduate college students so that they can answer questions on a multiple-choice exam. The instructor must help the reader (i.e., the student) to "connect the dots" and synthesize knowledge from these seemingly disconnected collections of readings. The readings require the students to take a non-traditional approach to studying when preparing for multiple-choice exams.

The vision for the Digital Systems course now included the exploration of systems that organizations use to create their digital products and explore the platforms that these systems are built upon and the API ecosystem by which systems extend their reach and capability. The core course provides students with an overview of IS in industry and is the foundation of learning content for MIS majors.

Even though APIs are providing large parts of systems development and integration, there is still coding that must take place, and it was decided to also teach the students some introductory level coding. Few students understand what software really is, let alone what is involved in its development. Most companies now recognize that their workers need to have basic technology abilities. Establishing basic technology literacy benefits both employers and employees, hopefully digitally transforming the worker and workplace (Fenlon & McEneaney, 2021).

Price Waterhouse Coopers surveyed over 28,000 employees working on a contract or on a temporary basis and determined that "workers want to reskill and 77% are ready to learn new skills or completely retrain." Workers, when given the opportunity, take advantage of upskilling opportunities; however, many unskilled workers do not have the same access to new training or resources and risk being left behind in the digital economy (PricewaterhouseCoopers, 2022). By learning basic technology skills in the core IS course, students require less upskilling once they are in the workplace.

The MIS department's advisory board indicated technical skills and capabilities were expected of recent graduates entering the workforce. The board felt that every business school student should have a basic understanding of coding. An added benefit of this code integration was that it would prepare those students who become MIS majors and minors for future MIS course content.

This led to the inclusion of material on APIs and an introduction to coding in JavaScript. This content was first piloted in two sections for honors students. The implementation condensed existing content and removed less relevant materials, making space for five weeks of coding content (including introductory JavaScript coding, HTML and CSS) at the end of the semester.

During the five weeks of coding, one third of the weekly class time was used for instruction, discussion, and an initial example or two. The other two thirds were used for in-class activities structured around "Paired Programming." The class would be divided into groups of two, with one student self-designated as the "driver" and the second as the "navigator." The instructor would introduce a coding challenge to the section and then instruct the paired teams to begin coding with the driver doing the writing and the navigator directing the approach and reviewing the syntax. The paired programming roles would switch after each challenge was completed. Instructors and information technology assistants would "walk the room" and engage with the various paired teams.

Coding challenges consisted of html files with the framework of the program provided and a commented-out section that included the description of a problem, indicating the challenge with a few hints and sometimes the formulas required to solve the challenge. An example is provided in Appendix A. Paired programming teams that completed a coding challenge were asked to share their html files with the instructor who then could share them on the projection screen. The instructor asked for volunteers to share/present their coding solution to the group, and they would run the program. The flexibility of JavaScript allows for variance in the coding and a few different solutions would be shared. Teams whose program would not run would open the "developer tools" and check the console for errors. The instructor could share the program on the classroom display screen, too, and engage the entire class in debugging the code.

Weekly homework assignments consisted of three sets of three coding challenges each. Every assignment followed the same framework as the in-class challenges and incrementally built upon the skills and tools discussed and experimented with during the in-class challenges. The final exam consisted of multiple-choice questions concerning general knowledge and coding case studies. Student feedback forms and in-class reflections indicated that the students enjoyed the coding content and appreciated the value of what they learned. The engaged class sessions pointed to positive results, too.

Narrative: An advantage of Max Labs is that the lab provides students with a narrative structure that can be used throughout the course. Szurmak and Thuna (2013) describe the role that narration can have in education. They state that "A narrative creates the scope for embedding details while simultaneously serving as the vehicle for establishing the large-scale guiding structure. Thus, working through a narrative allows the students to have both elements present in their learning and for later recall" (p. 546). With Max Labs, students see how a fictional student who takes initiative can gain valuable experience, but can also make some money at the same time. "Max" has many of the same experiences that the students have. Her instructions and explanations enable students to do complex work in *Salesforce*. The narrative gives the students a framework to help them understand the work that they are doing, and helps them remember what they have done. Max is in many ways a model student: hard working, independent, fun, someone who takes the initiative and tries things out without fear of making a mistake.

The team utilized insights from the pilot's roll-out, and the course curriculum was revisited to determine how to scale the coding pilot content across ten sections and 1,200 students. Working with honors students provided especially useful insights; however, when fully implemented, the class consisted of a much broader subset of students ranging from low to medium and high performers.

Additional inspiration was provided from a book about product management, “Project to Product” (Kersten, 2018). Specifically, the concept of the Digital Product Manager (DPM) was of special interest. The DPM concept provided a unifying element that the course coordinator utilized to connect-the-dots across each of the course’s 14-week sections.

IT Professionals are seen as Digital Product Managers (DPMs). They envision, plan for, deploy, and support digital products and services that solve problems for organizations. These DPMs must have strong technical skills, understand the users of their products and how they interact with their systems, and have an intimate understanding of their business and the problems that their business needs to solve.

The course framework took a narrative approach that was different from the past and was built around the exploration of the systems that organizations use to create their digital products. The DPM Narrative became the connective tissue that unified all the course concepts together, establishing a dynamic DPM framework for the course. Discussions are designed to explore the platforms on which these systems are built and the API ecosystem by which systems extend their reach and capability.

The first two thirds of the semester still did not require a book; however, the last third now required a book on JavaScript for beginners (Chinnathambi, 2019). The book introduces students to the basics of programming and is easy to read. Each week’s content could now be connected back to the overarching concept of the Digital Product Manager and the student’s journey towards becoming a DPM themselves.

Students were now expected to install a few tools on their laptops and bring their laptops to class every day during this part of the course. The Fox School of Business had a minimum standard set of laptop requirements that established a baseline for all students. This ensured faculty that students would have the appropriate digital tools to complete the new course work.

Course components were moved to a departmental LMS and the university LMS at the same time. A virtual coding Help Desk was established and was run by the course Information Technology Assistants (ITAs are successful former students of the course who are selected for their ability to help other students). The ITAs established a weekly schedule of *Zoom* sessions for students to attend at their discretion. ITAs were required to attend all coding classes with their respective section from weeks 9-14. Students tended to not attend the *Zoom* sessions unless a major deadline was coming up.

2.2 Success and Challenges of the Initial Rollout

A structure was put in place to promote continuous improvement of the course. Prior to the first refresh rollout, the course coordinator hosted two instructor coordination meetings to review the latest content and answer any questions/concerns of the team. Team feedback was critical to the iterative implementation of the refresh and the course coordinator established a series of three mandatory team meetings during the fall semester to discuss in real time what the team observed and experienced. Engaging the department chair was another key to establishing team buy-in to this new initiative. The department chair attended the first meeting and was kept informed about the efforts. The faculty team was part of the department’s larger initiative and the chair’s direct involvement in early discussions provided additional support and feedback to the team.

The course coordinator took a bottom-up approach of gaining input and buy-in for the refresh. Establishing a semester series of feedback meetings was a vital component of this rollout process. The sessions included a discussion of what worked and what did not, generated steps for improvement, empowered the team, and promoted buy-in. The course coordinator reviewed the team’s feedback, determined what was feasible for the next iteration, and developed an updated plan for the team to collaborate on developing changes. For example, one outcome was re-evaluating the assignment file upload process and moving it to the Google Form platform for uploading for the following semester. Students were familiar with the university’s Google platform tools and the successful rollout of the ICAs (that used the same tools) was a natural fit.

Another change was to use two free coding platforms, Notepad++ (Windows users only) or BBEdit (Mac users only). The School of Business’ laptop requirements establish a minimum functionality standpoint; however, they do not mandate using an iOS or Windows system. The amount of variance across the two platforms was not manageable. The course faculty decided to move to one other free platform, Visual Studio Code, for the following spring semester. This change was well received and while there were still challenges, all students, instructors, and ITAs were now on the same platform. This enhanced comprehension and communication as the content looked and worked similarly for all. Another benefit was the ease of troubleshooting, and a positive unintended consequence was that students helped one another in breakout rooms.

Historically, MIS student exam scores showed improvement with each exam within a semester. Scores increased 5-10 points on average across each of the three exams, showing the students’ ability to improve across the semester. The exams were not cumulative, and the newly established third exam covered only the new coding content. Students expressed anxiety about answering coding questions in an exam. However, average final scores ranged from 81-86, and feedback after the exam indicated their understanding and comfort with the coding content.

Active Learning: The introduction of In-class activities significantly increased classroom interaction, student participation, and overall attendance. Engagement was positive for each specific activity (as indicated by the student feedback submitted with each returned activity); however, students noted that they wanted more swimlane diagrams and ERDs to practice for exams. Another challenge students indicated is that they want more direct correlation between the In-Class Activities and the specific readings and discussion slide decks. These concerns were addressed by providing additional case studies and using in-class activity slide decks with content from the discussion decks for instructors to connect course concepts prior to starting each activity.

Dynamic Content: The introduction of the online reading collections affords the course coordinator the opportunity to iterate more effectively, reviewing and replacing readings and corresponding discussion-deck slides each semester to reflect current industry trends. Many students still indicate that they do complete the readings which was reflected on their first exam scores. Instructors are piloting a variety of classroom approaches including weekly quizzes, group questions, and random generators to call on students and encourage the completion of readings prior to class and improve exam scores.

Narrative: The development of Riley's Rankings, a JavaScript coding challenge, was the final large iterative change. A storyline from the Max Labs project was combined with the Digital Product Manager narrative into the JavaScript component of the course. The intent was to connect the IS and Max Labs concepts with building a digital tool using JavaScript. Max Labs has an established narrative and the course coordinator decided to expand on the Max Labs universe and develop an epilogue based on Riley's investor needs.

Riley's Rankings consisted of a narrative describing a business case, and a challenge for students to code the ranking program three JavaScript functions through a *for* loop, a *while* loop, and an *if else* statement. Students accessed a starter file via the LMS and developed the three functions as part of this final coding assignment. This new coding challenge took the place of the three previous coding assignments. One concern expressed by the team was that students would not be able to begin working on Riley's Ranking coding assignment until week 11, when students learn about conditional statements. This concern was addressed by discussing each assignment as it related to the corresponding weekly discussion, enabling the students to begin decomposing the assignment and thinking about how they would start developing their own solution.

The Course Coordinator identified a concern about students not being able to begin coding challenges on their own without help. Many students initially indicated that they did not know where to start. One approach identified in the MIS faculty team's feedback meeting was the creation of coding assignment starter videos. The Course Coordinator had a group of three information technology assistants to create a series of coding kickoff videos, walking students step by step through each line of code of the first few in-class challenges. This additional resource proved useful for many students.

The Riley's Ranking rollout was successful from a student score perspective; the average score was 90 out of 100, which indicates that most students were able to get their code running, many with just minor syntax issues.

3. COURSE FRAMEWORK

3.1 Required Readings

The first two-thirds of the semester do not require a textbook. For the coding part of the semester, a relatively inexpensive book on JavaScript is required (Chinnathambi, 2019). The Max Lab readings and materials are part of the core course content. The fee associated with the Max Lab assignments is $19.99 per student per semester. Course readings consist of a collection of free articles that are available over the Internet. These articles focus on the core learning objective covered each week.

3.2 Class Site/LMS Standardization

The course is designed by the course coordinator, with help from other faculty. For each section, a copy of the course structure is placed in the LMS. Individual instructors can then add additional content as they see fit.

3.3 Graded Components

There are three core components to the calculation of a student's final grade: In-Class Activities, Assignments, and Exams. The course grading components and their weights are shown in Table 1.

Grading Component	Percentage of Total
In-Class Activities	10%
Assignments	30%
Exam #1	20%
Exam #2	20%
Exam #3	20%

Table 1. Grading Components

3.4 In-Class Activities

All In-Class Activities (ICAs) are provided for each of the weeks of the semester by the course coordinator and are designed to be completed in 10 to 50 minutes. PowerPoint slide decks (with notes) are provided to help instructors facilitate these discussions. Announcements are not made about when the class will or will not be having an in-class activity, so students are encouraged to attend the class regularly.

The completion of ICAs represents a sizable portion of a student's final grade. Appendix B shows an example of an ICA.

3.5 Assignments

The course includes eleven assignments that are integrated with the weekly course discussions. The instructor's job is to "connect the dots" between the assignments and the course content using the digital product manager framework. Instructors are provided PowerPoint slide decks (with notes) to help facilitate these discussions.

3.6 Exams

The course includes three multiple choice exams: two midterms and a final exam. These are standardized exams across all sections to provide consistent data for the school's assurance of learning initiatives. The midterm exams partly cover concepts and partly cover "swimlane" process flow diagrams and entity relationship diagrams. The final covers coding in JavaScript.

3.7 Week-by-Week Class Overview

The semester is divided into nine major "Discussion" units. A road map (see Appendix C) was developed to enable the students to follow and reflect on their learning journey. This roadmap is used in all lecture notes, ICAs, and assignments to show the students where they are at any point in the semester.

4. EVIDENCE OF IMPACT

The university collects feedback from the students about their courses through formal Student Feedback Forms. This data was used to assess student perceptions of the course from fall 2012 through fall 2021. Only the full semester classes were used to compare the course from semester to semester. The official course evaluations collect data on the instructor, the course, and on student effort. Our focus here was on the evaluations of the course and of student effort.

The evaluations were compared for both regular courses, and honors sections. The honors section covers the same material, plus additional assignments and content designed for high-performing honors students. The material tends to be delivered more quickly and more efficiently, with less repetition.

No specific demographics were collected for the undergraduate business students in the course. Demographic data for the university is shown for the eleven-year period with the current data in parentheses: 8% (3%) non-resident alien; 60% (54%) white; 13% (15%) black or African American; 12% (9%) Hispanic or Latino; 8% (13%) Asian; 3% two or more races; 0% American Indian or Alaska Native; 0% Native Hawaiian or Other Pacific Islanders. Currently 56% of the undergraduates are women (Data USA, n.d.).

Four items were used to evaluate the course in the Fall 2012 – Spring 2020 period:

1. I came well prepared for class (1-5).
2. The course content was consistent with the educational objectives of this course (1-5).
3. The course increased my ability to analyze and critically evaluate ideas, arguments, and points of view (1-5).
4. I learned a great deal in this course (1-5).

The university's survey items were changed in the fall of 2020. The first question was dropped, and the wording of the three others was changed. However, the wordings are close enough that they can be matched with the earlier evaluations.

1. Removed.
2. In general, course materials helped me understand the topic better (1-5).
3. The course improved my critical-thinking skills (1-5).
4. Overall, I learned a great deal from this course (1-5).

Figure 1 shows course evaluations for spring and fall regular 14-week classes. The data shown is for all sections taught in a particular semester, by all instructors, typically 4 to 10 classes.

Figure 2 shows course evaluations for honors students spring and fall 14-week classes. No honors classes were offered in the fall of 2021.

There are several conclusions that can be drawn from the data.

1. Variation in the data over the 11 years is limited. The largest variation is in honors classes, for "course contents consistent with objectives," which varies from 3.0 to 4.8. Interpretation: Honors students tend to be more critical and expressive. They will let you know when they are happy or unhappy. Honors students are willing to work extremely hard, but they also want their efforts to be rewarded.
2. The measures tend to go up and down together. Interpretation: When students are happy or unhappy about elements of the course, it appears to color their entire perception of the work.
3. Since the spring of 2019, the honors courses have started to receive higher evaluations than the regular courses.
4. At the time of the introduction of coding in the fall of 2019, there was a clear drop in course evaluations in the regular classes, but not in the honors course. Interpretation: The honors students appreciated the challenge and usefulness of learning to code more than the regular students. The regular students were more frustrated by any of the technological problems they encountered.
5. Temporary drops in evaluations appear to take place whenever a major change is made in the course, for example as discussed in point 4 above. Interpretation: The dips disappear as instructors learn how to better deliver the new material, and as students become more aware of the changes, and as the rationale for the new material is better explained to them.

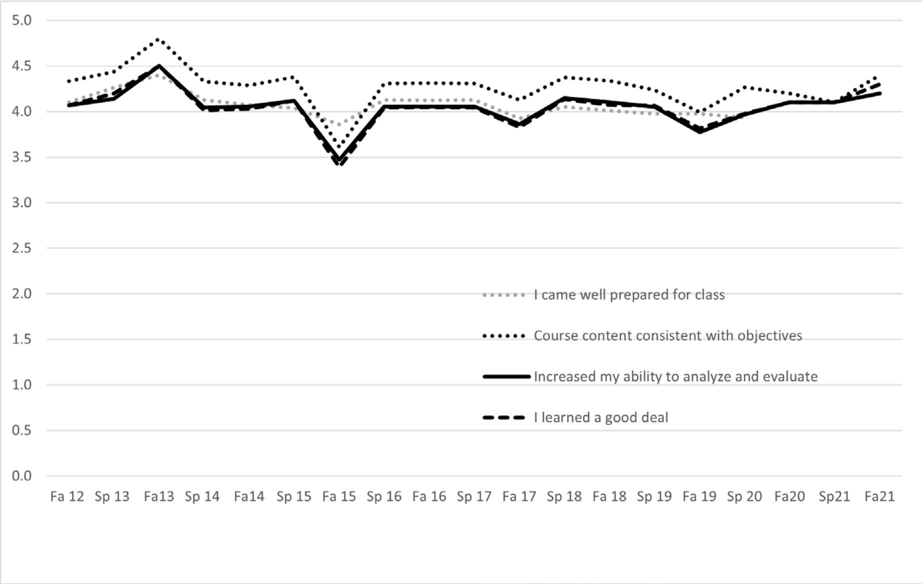


Figure 1. Course Evaluations Regular Classes (5-Point Scale)

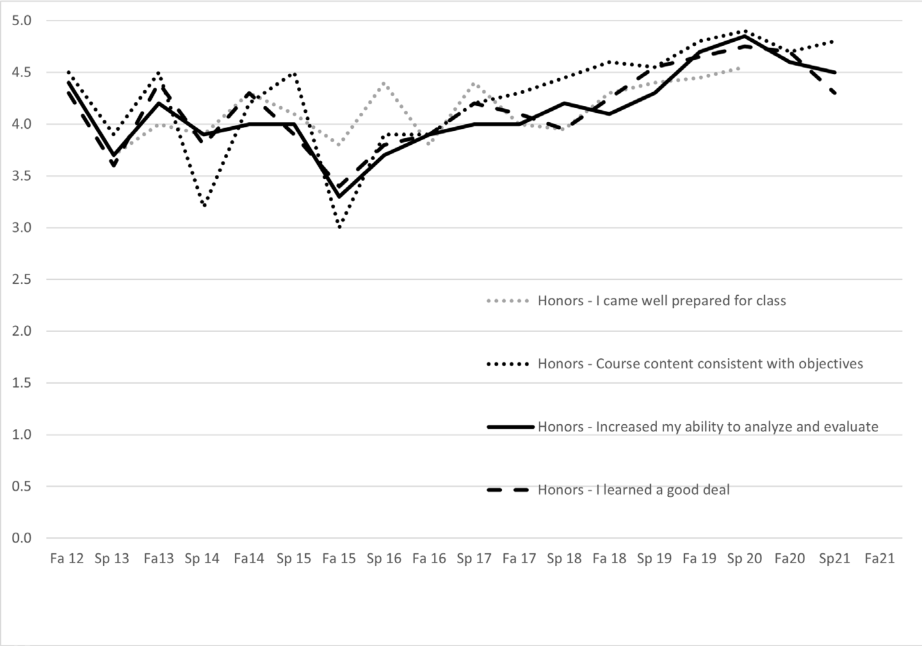


Figure 2. Course Evaluations Regular Classes (5-Point Scale)

5. DISCUSSION AND CONCLUSION

Effectively iterating a large-scale IS course's content when it is a requirement within a business school's core curriculum is challenging and necessary to stay relevant and engaging. This teaching tip provides an effective, time-tested approach to help IS instructors create a scaffolding for continuous improvement which corresponds to course and school learning outcomes. To our knowledge, this is the first teaching tip that combines three themes (i.e., Active Learning, Dynamic Content, and Narrative) to create a framework for building, managing, and maintaining a large scale IS course.

Moving the *Introduction to IS* class from what were disparate parts to a unified narrative and a true holistic approach capped with coding was quite radical. These changes occurred over an eleven-year period providing a new framework for the course and culminating in the evolution of students from traditional learners to burgeoning Digital Product Managers. The three themes — Active Learning, Dynamic Content, and Narrative — each had their own impact.

▲ **Active learning:** Many new activities have been added over time. These changes raised the levels of learning in Bloom's taxonomy, and the students are learning many more applied technologies. With the available data, it is not possible to determine if the students achieve higher levels of learning; however, qualitative observation indicates they are more engaged during the classes.

⚙ **Dynamic content:** The rate of technological change increases each year. Information systems faculty have a vital role to explain what these changes are, and how they are likely to impact students and the industries they will be joining.

📖 **Narrative:** In the experience of the authors, it is more pleasant to work with a narrative. The course is more integrated because of the narrative, and it is easier to show students the connections. In a potential future study, it would be especially useful to collect data from students and other faculty to determine if they value the narrative approach, and to assess if it helps them.

The incorporation of the three themes creates a mutually reinforcing structure and drives these improvements. The active learning activities can make the students more excited about the content, and with iterative, up-to-date content, the activities are more relevant to the students. The evolution and mindful merging of three traditionally independent themes, Active Learning: ▲ Dynamic Content: ⚙ and Narrative: 📖 led to the formation of one cohesive approach. The combination of these three themes works not only in a core information systems course but could work in elective courses and for other subjects.

IS education, with inherent technical aspects and ever-changing contents, is challenging to teach. The authors believe that to keep the courses relevant, understandable, and engaging, the course must constantly evolve. As this Teaching Tip suggests, this evolution can be driven by designing and improving the course around a narrative, with active learning activities and dynamic content. The core MIS course discussed here will continue to evolve. The next iteration has already started, and a major review has been called for the spring and summer of 2023.

6. REFERENCES

Anderson, L. W., & Krathwohl, D. R. (2001). *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's*

Taxonomy of Educational Objectives. New York, NY: Longman.

Bergmann, J., & Sams, A. (2012). *Flip Your Classroom: Reach Every Student in Every Class Every Day*. Eugene, OR: International Society for Technology in Education.

Bloom, B. S. (1956). *Taxonomy of Educational Objectives. Vol. 1: Cognitive Domain.* New York, NY: McKay.

Bonwell, C. C., & Eison, J. A. (1991). *Active Learning: Creating Excitement in the Classroom.* Washington, DC: School of Education and Human Development, George Washington University.

Carini, R. M., Kuh, G. D., & Klein, S. P. (2006). Student Engagement and Student Learning: Testing the Linkages. *Research in Higher Education*, 47(1), 1-32. https://doi.org/10.1007/s11162-005-8150-9

Chinnathambi, K. (2019). *JavaScript Absolute Beginner's Guide*. Indianapolis, IN: Que.

Data USA. (n.d.). *Temple University*. https://datausa.io/profile/university/temple-university

Doyle, M., Flanagan, R., & Moustafellos, J. (2015). Learning by Doing: Studio Classes in MIS Education. *Americas Conference on Information Systems 2015 Proceedings.* 14. https://aisel.aisnet.org/amcis2015/ISEdu/GeneralPresentations/14

Fenlon, M., & McEneaney, S. (2021). How We Teach Digital Skills at PwC. *Harvard Business Review*, Special Issue, Summer 2021, 144-145.

Foldnes, N. (2016). The Flipped Classroom and Cooperative Learning: Evidence From a Randomised Experiment. *Active Learning in Higher Education*, 17(1), 39-49. https://doi.org/10.1177/1469787415616726

Forehand, M. (2010). Bloom's Taxonomy. *Emerging Perspectives on Learning, Teaching, and Technology*, 41(4), 47-56.

Freeman, S., Eddy, S. L., McDonough, M., Smith, M. K., Okoroafor, N., Jordt, H., & Wenderoth, M. P. (2014). Active Learning Increases Student Performance in Science, Engineering, and Mathematics. *Proceedings of the National Academy of Sciences*, 111(23), 8410-8415. https://doi.org/10.1073/pnas.1319030111

Hargreaves, H., Robin, S. & Caldwell, E. (2022) Student Perceptions of Reading Digital Texts for University Study. *Journal of Learning Development in Higher Education*, 2022(24), 1-21. https://doi.org/10.47408/jldhe.vi24.817

Hill, T., & Nance, W. (2016). Innovating Business Systems Labs for Engaging iGeneration Students. *Americas Conference on Information Systems 2016 Proceedings*. https://aisel.aisnet.org/amcis2016/ISEdu/Presentations/29

Kersten, M. (2018). *Project to Product: How to Survive and Thrive in the Age of Digital Disruption With the Flow Framework*. Portland, OR: IT Revolution.

Kuang, C. (2015, March 10). Disney's $1 Billion Bet on a Magical Wristband. *Wired Magazine* https://www.wired.com/2015/03/disney-magicband/

Lavin, A., Martin, M.C., & Sclarow, S. (2018). Radically Redesigning Introductory MIS Large-Scale Lectures: Creating Enhanced Learning Environments. *Proceedings of the 2018 AIS SIGED International Conference on Information Systems Education and Research.* https://aisel.aisnet.org/siged2018/28

Mok, H. N. (2014). Teaching Tip: The Flipped Classroom. *Journal of Information Systems Education*, 25(1), 7-12.

PricewaterhouseCoopers. (2022, December 8). *Upskilling Hopes and Fears 2021*. PwC. https://www.pwc.com/gx/en/issues/upskilling/hopes-and-fears.html

Prince, M. (2004). Does Active Learning Work? A Review of the Research. *Journal of Engineering Education*, 93(3), 223-231. https://doi.org/10.1002/j.2168-9830.2004.tb00809.x

Riordan, R. J., Hine, M. J., & Smith, T. C. (2017). An Integrated Learning Approach to Teaching an Undergraduate Information Systems Course. *Journal of Information Systems Education*, 28(1), 59-70.

Romanow, D., Napier, N. P., & Cline, M. K. (2020). Using Active Learning, Group Formation, and Discussion to Increase Student Learning: A Business Intelligence Skills Analysis. *Journal of Information Systems Education*, 31(3), 218-231.

Sandrone, S., Berthaud, J. V., Carlson, C., Cios, J., Dixit, N., Farheen, A., Kraker, J., Owens, J. W., Patino, G., & Sarva, H. (2020). Active Learning in Psychiatry Education: Current Practices and Future Perspectives. *Frontiers in Psychiatry*, 11(211), 1-7. https://doi.org/10.3389/fpsyt.2020.00211

Szurmak, J., & Thuna, M. (2013). Tell Me a Story: The Use of Narrative as Tool for Instruction. *Proceedings of the ACRL 2013 Conference*.

White, K. (2021, August 18). What Is a Product Manager? A Complex, Cross-Functional Role in IT. *CIO Digital Magazine*. https://www.cio.com/article/221885/what-is-a-product-manager-a-complex-cross-functional-role-in-it.html

Woods, D. M. (2020). Active Learning Using Debates in an IT Strategy Course. *Journal of Information Systems Education*, 31(1), 40-50.

AUTHOR BIOGRAPHIES

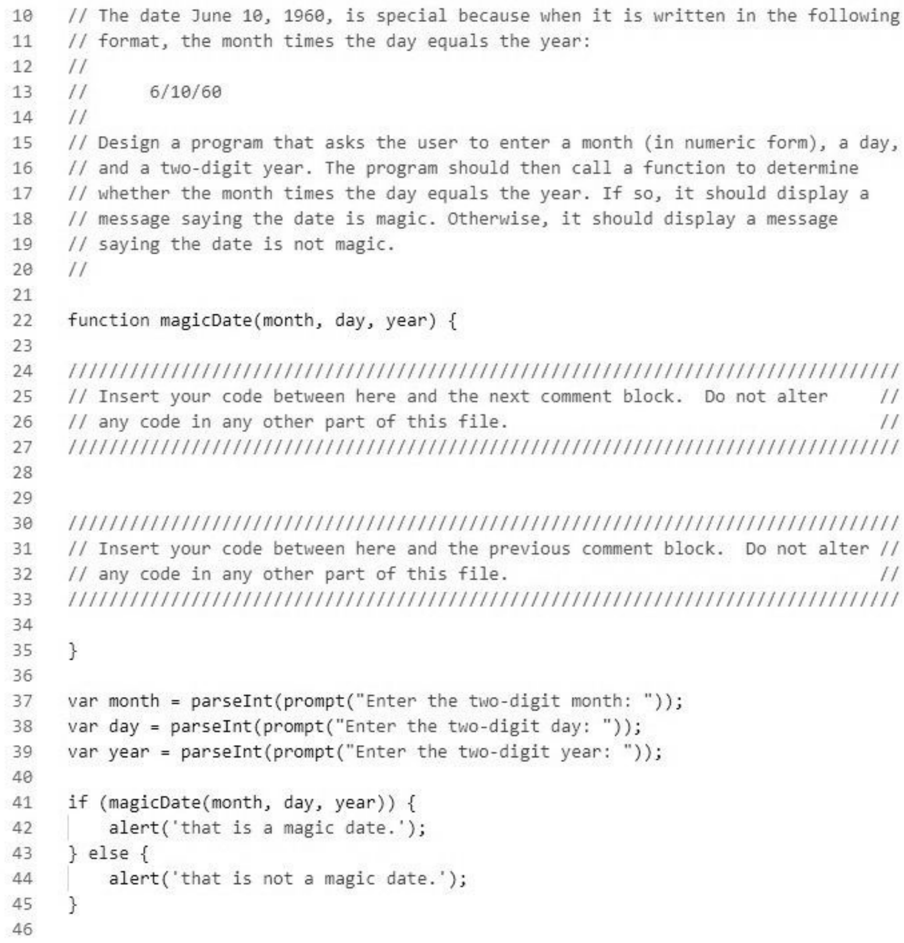
Steven Sclarow joined Temple University's Fox School of Business as a Management Information Systems Adjunct in 2016 and became a full time Professor in 2017. Steve has taught multiple sections of Digital Systems, Digital Design & Innovation, Design Thinking and Process Improvement & Innovation. Steve spent the previous 20 years of his professional experience in the Architectural and Construction Industry. He is a licensed architect with an established reputation for excellence, high project profitability and investor ROI. Steve keeps his skills and architectural licenses current with an active consulting practice in addition to teaching undergraduate, graduate and MBA programs.

AJ Raven is an Associate Clinical Professor of Management Information Systems in the LeBow College of Business at Drexel University. In addition to having 18 years of academic experience, he has spent five years in industry as the general manager and a member of the board of directors of Sica America, and as an IT/strategy consultant. His research interests are in the pedagogy of management information systems, especially the use of active learning techniques. He has published in nine journals.

Mart Doyle joined the full-time faculty of the MIS department at Temple University in January 2005. Before joining Temple, he spent the previous 20 years working in the IT industry planning, deploying, and supporting technology. Mart worked for the consulting giants CSC and IBM before starting his own independent consulting practice. He also played traditional roles in corporate America, including working as the Chief Technology Architect for a Philadelphia based global, Fortune 500 Company. In addition to teaching in the undergraduate, graduate, Executive MBA and International MBA programs, Mart keeps his skills current with an active consulting practice.

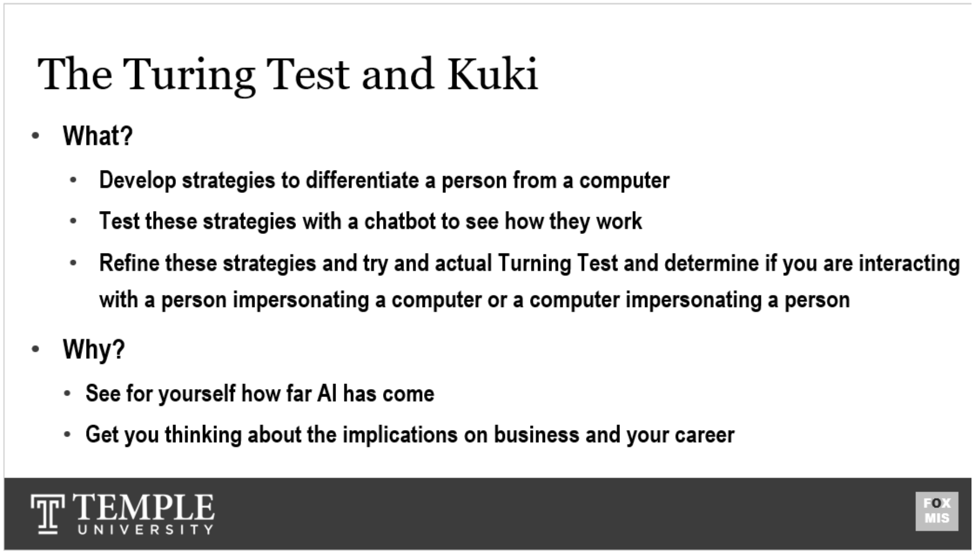
APPENDICES

Appendix A. Sample of JavaScript assignment in VS Code



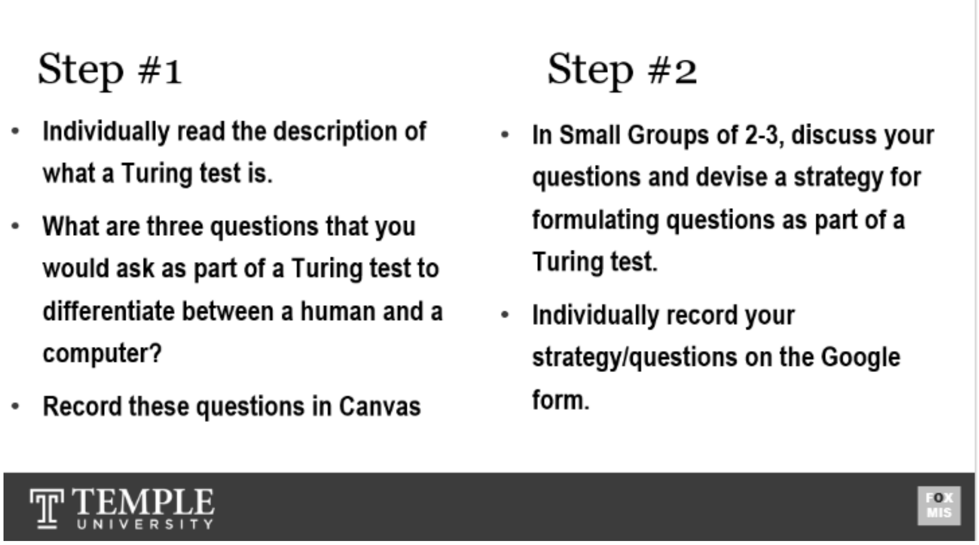
```
// The date June 10, 1960, is special because when it is written in the following
// format, the month times the day equals the year:
//
//      6/10/60
//
// Design a program that asks the user to enter a month (in numeric form), a day,
// and a two-digit year. The program should then call a function to determine
// whether the month times the day equals the year. If so, it should display a
// message saying the date is magic. Otherwise, it should display a message
// saying the date is not magic.
//

function magicDate(month, day, year) {

/////////////////////////////////////////////////////////////////////////////////
// Insert your code between here and the next comment block.  Do not alter     //
// any code in any other part of this file.                                    //
/////////////////////////////////////////////////////////////////////////////////


/////////////////////////////////////////////////////////////////////////////////
// Insert your code between here and the previous comment block.  Do not alter //
// any code in any other part of this file.                                    //
/////////////////////////////////////////////////////////////////////////////////

}

var month = parseInt(prompt("Enter the two-digit month: "));
var day = parseInt(prompt("Enter the two-digit day: "));
var year = parseInt(prompt("Enter the two-digit year: "));

if (magicDate(month, day, year)) {
    alert('that is a magic date.');
} else {
    alert('that is not a magic date.');
}

```

Appendix B. Sample of In-Class Activity



The Turing Test and Kuki

- What?
  - Develop strategies to differentiate a person from a computer
  - Test these strategies with a chatbot to see how they work
  - Refine these strategies and try and actual Turning Test and determine if you are interacting with a person impersonating a computer or a computer impersonating a person
- Why?
  - See for yourself how far AI has come
  - Get you thinking about the implications on business and your career

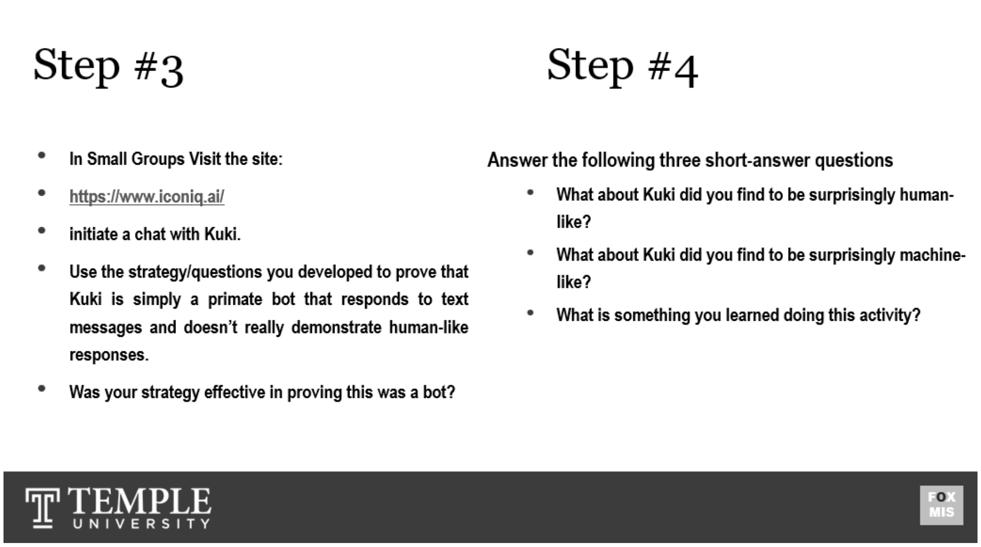


Step #1

- Individually read the description of what a Turing test is.
- What are three questions that you would ask as part of a Turing test to differentiate between a human and a computer?
- Record these questions in Canvas

Step #2

- In Small Groups of 2-3, discuss your questions and devise a strategy for formulating questions as part of a Turing test.
- Individually record your strategy/questions on the Google form.



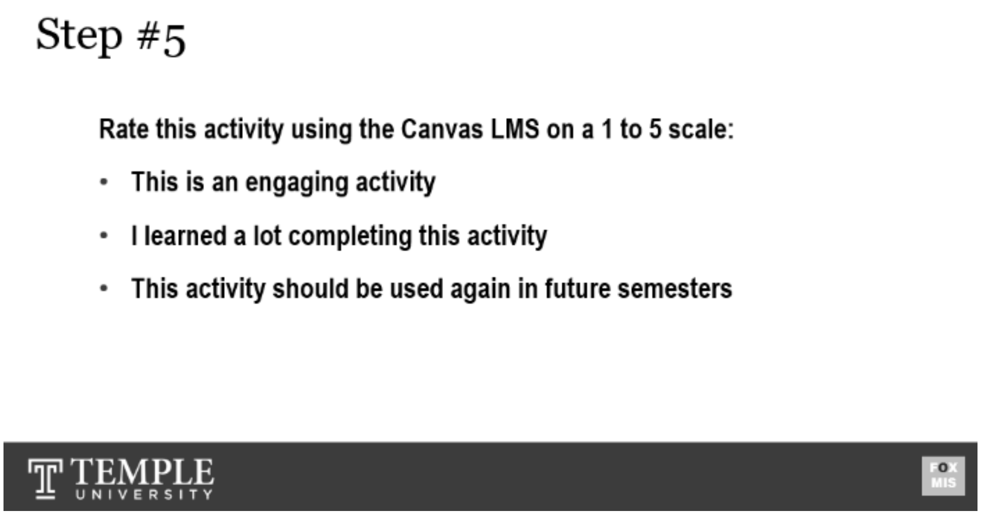
Step #3

- In Small Groups Visit the site:
- https://www.iconiq.ai/
- initiate a chat with Kuki.
- Use the strategy/questions you developed to prove that Kuki is simply a primate bot that responds to text messages and doesn't really demonstrate human-like responses.
- Was your strategy effective in proving this was a bot?

Step #4

Answer the following three short-answer questions

- What about Kuki did you find to be surprisingly human-like?
- What about Kuki did you find to be surprisingly machine-like?
- What is something you learned doing this activity?



Step #5

Rate this activity using the Canvas LMS on a 1 to 5 scale:

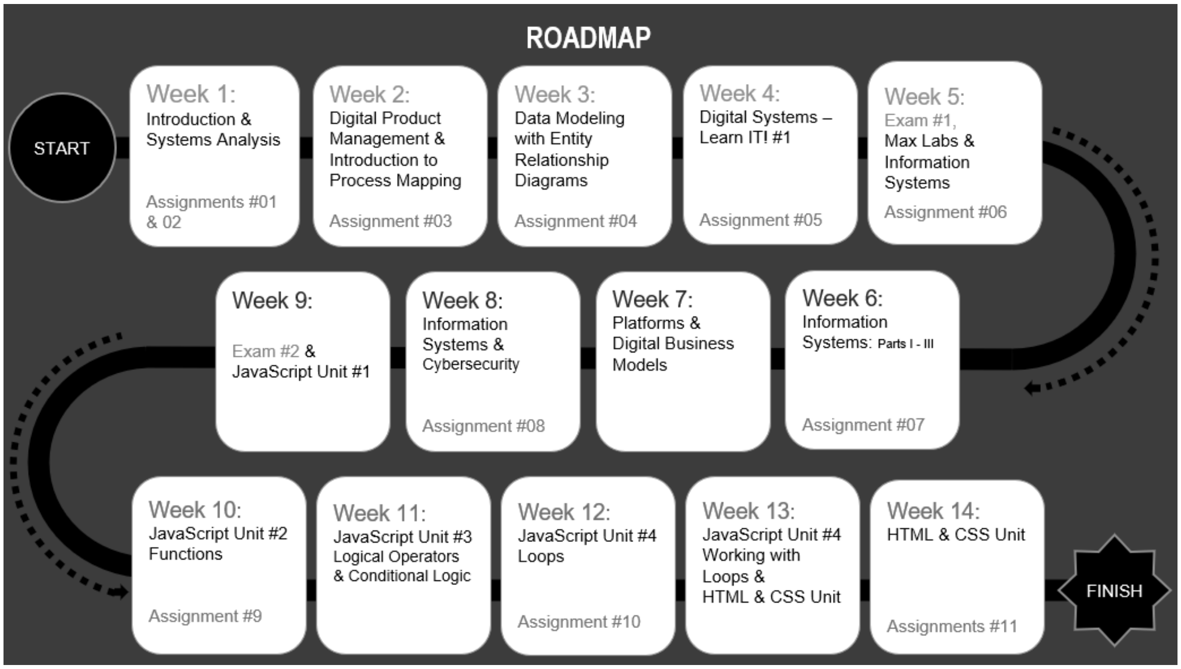
- **This is an engaging activity**
- **I learned a lot completing this activity**
- **This activity should be used again in future semesters**

TEMPLE UNIVERSITY

FOX MIS

Appendix C. Sample of Discussion Units and Weekly RoadMap
The semester is divided into ten major "Discussion" units:

1. Systems Analysis
2. Digital Product Management, Max Labs 1 & Process Mapping/Swim Lanes Intro Part 1
3. Swim Lanes Part 2 & Data Modeling Intro - Entity Relationship Diagrams (ERD) – Part 1
4. Entity Relationship Diagrams (ERD) – Part 2 & Digital Brand Management
5. Information Systems Part I & II - CRM & ERP
6. Information Systems Part III & IV – Data Analytics & SCM
7. Platforms & Digital Business models, including API's
8. Cybersecurity and the Enterprise plus AI
9. JavaScript: Units 1 – 4, plus HTML & CSS



ROADMAP

START

Week 1: Introduction & Systems Analysis. Assignments #01 & 02

Week 2: Digital Product Management & Introduction to Process Mapping. Assignment #03

Week 3: Data Modeling with Entity Relationship Diagrams. Assignment #04

Week 4: Digital Systems – Learn IT! #1. Assignment #05

Week 5: Exam #1, Max Labs & Information Systems. Assignment #06

Week 6: Information Systems: Parts I - III. Assignment #07

Week 7: Platforms & Digital Business Models

Week 8: Information Systems & Cybersecurity. Assignment #08

Week 9: Exam #2 & JavaScript Unit #1

Week 10: JavaScript Unit #2 Functions. Assignment #9

Week 11: JavaScript Unit #3 Logical Operators & Conditional Logic

Week 12: JavaScript Unit #4 Loops. Assignment #10

Week 13: JavaScript Unit #4 Working with Loops & HTML & CSS Unit

Week 14: HTML & CSS Unit. Assignments #11

FINISH

INFORMATION SYSTEMS & COMPUTING ACADEMIC PROFESSIONALS

STATEMENT OF PEER REVIEW INTEGRITY

All papers published in the ***Journal of Information Systems Education*** have undergone rigorous peer review. This includes an initial editor screening and double-blind refereeing by three or more expert referees.

ISSN: 2574-3872 (Online) 1055-3096 (Print)