Communications of the Association for Information Systems

Volume 54 Paper 9

3-4-2024

Modular Design of Teaching Cases: Reducing Workload While Maximizing Reusability

Kevin Parker
Idaho State University, parkerkr@isu.edu

Chang-Tang Chiang
Chinese Culture University, Taiwan, arthurctchiang@gmail.com

Modular Design of Teaching Cases: Reducing Workload While Maximizing Reusability

Cover Page Footnote

This manuscript underwent editorial review. It was received 06/06/2023 and was with the authors for six months for one revision. Blooma John served as Associate Editor.

Communications of the Association for Information Systems

Education Article DOI: 10.17705/1CAIS.05409 ISSN: 1529-3181

Modular Design of Teaching Cases: Reducing Workload While Maximizing Reusability

Kevin R. Parker
College of Business
Idaho State University
parkerkr@isu.edu
0000-0003-0549-3687

Chang-Tang Chiang
Department of Information Management
Chinese Culture University
Taiwan
arthurctchiang@gmail.com
0000-0003-1046-547X

Abstract:

Many software development courses rely heavily on a project-based learning component to provide hands-on experience with the majority of the course objectives. Courses such as systems analysis and design, database design, software engineering, and capstone software development courses often incorporate some form of case-based project that is much more extensive and integrated than are focused exercises over individual topics. To be truly educational, case studies can require a significant investment in time and effort on the part of the instructor, an investment that seldom carries over into future academic sessions. This paper presents a modular case study development concept for better managing the development of case studies and achieving project extensibility through reusable case study modules, while at the same time helping to reduce instructor workload and solution reuse by students.

Keywords: Case-based Instruction, Teaching Case, Active Learning, Experiential Learning and Education, Software Development, Systems Analysis and Design, Database Design and Implementation, Academic Workload, Plagiarism.

This manuscript underwent editorial review. It was received 06/06/2023 and was with the authors for six months for one revision. Blooma John served as Associate Editor.

1 Introduction

The case study has been regarded as one of the most effective learning tools because students find it more nurturing and enjoyable, and students applying course concepts through case studies perform as well as or better than students learning in “traditional instruction” (Albanese & Mitchell, 1993). Another advantage of case studies is that they often keep students’ attention more effectively due to their realism, relevance, and when well-designed, direct connection to course objectives (Grimes, 2019). Many courses rely on a case study component to reinforce the established learning outcomes of the course. When a case study component is incorporated into a project-based learning component, the combination can provide experiential learning opportunities through the application of course concepts. This type of project-based learning helps students develop applied skills by having them engage in representative tasks typically encountered in a real-world context (Warlick, 1999). Case studies relevant to course content can be found or adapted from other sources, such as textbooks or journals, or even created as needed (Nilson, 2013).

Software development courses such as systems analysis and design, database design, software engineering, and capstone software development courses rely heavily on a project-based learning component to provide hands-on experience with several of the course objectives. The case-based projects used in such courses generally are much more extensive and integrated than focused exercises over individual topics.

However, there may not be an abundance of case studies that lend themselves to addressing the learning objectives of a particular course. In such situations, a new case study may need to be developed. Developing case studies requires considerable effort, an effort that is over and above the development of standard course content like lectures, demonstrations, exercises, exams, and quizzes. To be truly educational, case studies can require a substantial investment in development time and effort on the part of the instructor, an investment that seldom carries over into future academic sessions. Once used in class, the solution can become compromised if shared among students from semester to semester (McCardle, 2010). Therefore, there can be a significant downside to reusing a case study from one academic session to another.

The purpose of this paper is to present a modular approach for developing reusable cases with two primary objectives: discouraging student sharing while at the same time helping to reduce instructor workload. The pedagogical approach described here was initially motivated by the two competing goals associated with developing case studies—maximizing reusability from the standpoint of the instructor, and minimizing it from the standpoint of the students. These goals seem almost mutually exclusive. The modular case study development approach develops a case study as a baseline case along with multiple modules, which, when combined, form a coherent whole. Once the case study module descriptions and solutions are developed, they can be used in various configurations multiple times. By swapping modules in and out of a case study description, along with adding or removing modules, the instructor can sufficiently vary a case study description to refine the problem that the case study addresses. This approach addresses the problem of reusability from the instructor’s perspective by making it unnecessary to develop an entirely new case description every semester. Further, the ability to add modules to a case study description makes case studies extensible. Finally, with each new case study made up of various interchangeable modules, solutions from previous semesters are no longer entirely applicable, and the opportunities for academic dishonesty are reduced. This modular case study development approach helps instructors better manage the practical component of experiential learning. While not the intent of this study, one fortuitous byproduct of the modular case study development concept is that it can also assist instructors who prefer to give every student or student group a different case study to work on to reduce students’ temptation to cheat (Lewis & Massingill, 2006; Pérez & Rubio, 2020).

2 The Scholarly Roots of Modular Case Study Development

A variety of research areas contributed to the inception of this teaching tool, including case-based learning, project-based learning, projects in software development courses, real-world case-based projects, simulated case-based projects, component-based development, and modular design.

2.1 Case-based Learning

Case-based learning (CBL) is an active-learning approach that has been promoted for increasing student success in STEM (Rhodes et al., 2020). CBL is also an authentic learning approach because it allows students to directly apply their knowledge or skills to realistic or real-world scenarios, bridging the gap between theory and practice (Lombardi, 2007; Hamdy, 2015). CBL uses hypothetical or real-life problems that help learners develop problem-solving skills through contemplation of different viewpoints or assumptions, and exploring alternate decisions (Leland & Nicholas, 2023). CBL facilitates active learning by applying the students' theoretical knowledge to real-world scenarios (Ameta et al., 2020). CBL can assume many forms but generally relies on a case study that describes a specific situation and requires students to work through the information to generate solutions and solve problems (Savery, 2015). CBL requires students to use techniques acquired in class to help them integrate, synthesize, and apply newly learned information in a broader context, foster critical-thinking skills, develop problem-solving skills and collaborative skills, and learn how to apply what they are learning (Rhodes et al., 2020).

There is a variety of sources for case studies to be used in teaching. Teaching cases for specific disciplines can be found in textbooks, discipline-based journals, and conference proceedings.

Textbook-based cases are used to reinforce concepts in specific courses. They can appear in textbooks as cases intended to serve as course projects, with suggested solutions provided in the instructors' manual. Other textbooks provide only cases incorporated into the chapters to serve as examples and reinforce chapter content. An example of a textbook that provides case studies that can be assigned as course projects *is Modern Database Management* (McFadden & Hoffer, 1991). It includes case studies like "Mountain View Community Hospital" and "Pine Valley Furniture Company" that instructors can assign to their students. As solutions for such cases proliferate online, subsequent editions of the textbook may incorporate the cases into the textbook as examples. For example, "Pine Valley Furniture Company" is now an ongoing example throughout the textbook, and the "Forondo Artist Management Excellence Inc." now serves as a mini-case with project assignments (Hoffer et al., 2022). Other textbooks primarily provide illustrative case studies that serve as ongoing examples. For example, Spurrier and Topi (2021) provide an ongoing I2C2 Pharmacy case study and mini-cases to reinforce text content.

There are several journals dedicated to the publication of case studies, such as the *International Journal of Teaching and Case Studies* and the *Journal of Business Case Studies*. Discipline-based journals like the *Journal of Information Systems Education* and *Information Systems Education Journal* sometimes have special issues on teaching cases. In addition, there are professional associations like the North American Case Research Association, the Society for Case Research, and The CASE Association (formerly known as the Eastern Case Writers' Association) that assist in the publication of written cases. They publish teaching cases in their conference proceedings or affiliated journals, like the *CASE Journal* (the official journal of The CASE Association).

2.2 Project-based Learning

There are similarities between project-based learning and other approaches such as intentional learning (Scardamalia & Bereiter, 1991), design experiments (Brown, 1992), and problem-based learning (Gallagher et al., 1992), as well as project-focused, experiential education, and active learning (Thomas, 2000). Project-based learning organizes learning around projects (Thomas, 2000). Projects are typically complex tasks, based on challenging problems and involving students in design, problem-solving, decision-making, or investigative activities. Further, they culminate in realistic results or deliverables (Jones et al., 1997; Thomas et al., 1999). Project-based learning is based on constructivist theory (Henze & Nejdl, 1997), the focus of which is what actual people in a real-life context typically do (Bednar et al., 1992). Students learn content knowledge, skills, and dispositions while working through a realistic project modeled on a scenario that might be encountered in the real world.

Students engaged in project-based learning assume a real-life role and apply the tools of a knowledge domain in completing a project. The opportunity to learn while engaged in a real-life situation facilitates the transfer of learning (Fisher & Frey, 2007). Project-based learning allows students to address complex questions and undertake projects that require the synthesis and application of course knowledge applied to real-world issues.

2.3 Projects in Software Development Courses

Learning to apply the fundamental techniques of software development or database design requires more than simply listening to lectures about current approaches and best practices. Although traditional teaching is based on lectures focusing on theoretical fundamentals accompanied by explanations of hypothetical examples and analysis of case studies (Fertalj et al., 2013), such teaching practices fail to provide either the context or sufficient practice for students to develop the skills needed to deal with real work problems (Brodie et al., 2008). Students often have limited exposure to working on large projects (Cohen & Menzies, 1994) as well as little awareness of the needs of the businesses for whom they are expected to build software systems (Grundy, 1996). University graduates entering the workforce with little idea of what will be required of them in their industry roles is both unacceptable and unproductive (Grundy, 1996). Hence, a critical component of any software development course is a student project in which theory is put into practice (van Vliet & Pietron, 2006) by engaging students in "stimulating, practical tasks that are based upon real-world requirements" (Kerins, 2012, p. 80).

Multiple disciplines include a focus on software development, including information systems, computer science, and software engineering. The IS 2010 Information Systems curriculum guidelines (Topi et al., 2010) state that software developers must have a broad understanding of business and other domains, strong analytical and critical thinking skills, and strong interpersonal communication and team skills. Toward that end, courses should be designed to provide students with hands-on experience with management issues and the complexities of running a real-world software development project (Scott, 2006). The same holds for computer science and software engineering programs. Computer science capstone courses should involve projects characterized by a level of structured complexity that is potentially solvable, rather than an entire real-world project that would, in actuality, take far more time than is available in an academic term (Russell et al., 2005). Engaging both computer science and software engineering students in real-world projects that require a real teamwork environment is of great importance in a complete education (Fertalj et al., 2013).

The type of projects that are the focus of this study reinforces the entire spectrum of concepts taught in the class or across the curriculum, rather than one single topic. Class projects may range from individual to group projects (Lenox & Woratschek, 2005), and vary from larger-scale simulated situations to actual cases to real-world projects that are based on scenarios in existing companies, for example, Cappel (2002), Scott (2006), Surendran et al., (2005), and van Vliet and Pietron (2006).

2.4 Real-world Case-based Projects

Systems development for real-world clients has several important advantages including realism, interaction with real clients, and the potential to develop a system that is useful to an organization (Cappel, 2002). Most case studies used in the classroom are based on real, actual teaching and learning experiences (Trier, 2010). In many cases, software development courses involve real-world clients with real needs, which gives students a unique and realistic experience (van Vliet & Pietron, 2006). The most obvious advantages of factual case studies are that they can provide a wealth of detail and give credibility to situations and problems (Warner, 2009). Scenarios involving real-world clients and projects provide students with experience both in interacting with actual clients and in addressing potentially "big" problems (Grundy, 1996). Real-world projects provide students with a more realistic perspective of what is relevant in the environment in which they will be working (Ellen & West, 2003).

Working with "live" projects can be plagued by a variety of challenges, including manageability, unpredictability, client demands, client expectations, additional faculty time, and increased grading complexity.

Project selection can be challenging. While projects may be encountered throughout the industry, a very limited number of them are suitable for use in the classroom. Project size, duration, and complexity can significantly reduce the range of plausible options (Fertalj et al., 2013). Universities located some distance from larger metropolitan areas may have limited access to live cases, especially cases of ideal size. Projects characterized by an extensive scope, extremely complicated processes, or ambiguous information sources might prove too complex and challenging for student developers, resulting in multiple roadblocks throughout the academic term that can overwhelm inexperienced developers (Harris, 2009). Authentic projects often need to be scaled down and adjusted to the selection of teaching topics and the schedule of classes (Fertalj et al., 2013). Scott (2006) recommends initially defining a generic business problem and then seeking businesses that match the problem and are willing to support student groups.

Some instructors assign a single system development project from an actual organization, an approach that can be driven by requests for assistance from entities on or off campus to instructors. Other instructors require that student teams locate their own clients for a development project (Cappel, 2002). Cappel (2002) points out the difficulty in finding “clients with suitable projects that are ‘doable’ in the limited time frame of a semester or quarter” (p. 233). Real-world projects can be “very large and monstrously complex” (Fertalj et al., 2013, p. 178), and making such projects feasible in academic settings may require decomposition, modularization, and reducing the scope to subsystems rather than entire systems.

The primary disadvantage of real-world projects is their unpredictability (Grundy, 1996). Involvement with real users quickly educates students about the problem of imprecise, inconsistent, and changing requirements (Fertalj et al., 2013).

Real projects require extensive interaction with real clients. Companies must be able to guarantee the availability of organizational personnel to meet with students in interviews early in the academic term and periodically throughout the course (Cappel, 2002). If company personnel do not provide adequate support for a project, project requirements will be inaccurate or incomplete, and the lack of ongoing feedback that is critical to successful software development will result in a failed project.

If a client “engages the services” of a software development class, they may have unrealistic expectations of the final outcome. Typical software development courses may be taken by third-year undergraduates, students who may know something about programming but little about developing large functional systems. Clients must be given realistic expectations regarding the deliverables that they can expect from such classes. Capstone courses will have more advanced students, but again the clients must be warned to have realistic expectations. If the project fails or ends with an expectations gap, then the disappointment of wasted time and effort could prevent further collaboration in future projects (Fertalj et al., 2013). Seldom will a fully implemented system be produced (Grundy, 1996).

Any worthwhile projects require substantially greater amounts of faculty effort and time, including the recruitment of real-world clients for project assignments (Jensen & Wee, 2000), mentoring throughout the systems development process (Fox, 2002), the management and guidance of student groups (Tan & Phillips, 2003), and a complex grading effort when students collaborate in groups, but especially if teams work on dissimilar project assignments (Sherman, 2000; Tan & Phillips, 2005; van Vliet & Pietron, 2006).

Incorporating real-world projects in software development courses leads to project deliverables that lack predetermined solutions, making them more difficult to grade (Hayes, 2002). It may also require the instructor to design a solution to the case in advance to detect obstacles that students will encounter and to make it possible for the instructor to provide students with useful and effective guidance. Hence, the use of real-world project case studies in software development courses may, in effect, involve free consulting services on the part of the instructor.

Finally, real-world projects are difficult to reuse without risking compromised solutions, unless they were poorly done and the client requests another attempt by an upcoming iteration of the class.

2.5 Simulated Case-based Projects

As an alternative to real-world project cases, inventive instructors might create a simulated scenario (Green, 2000). Simulated project case studies, also known as hypothetical cases or fictional cases, focus on hypothetical but realistic situations that make possible a more manageable application of specific methodologies and approaches. Such cases are “solely the result of the imagination of the case writer, which requires a significant amount of creativity on the part of the writer” (Buckles & Redmer, 2004, p. 8). As a simplified yet realistic depiction of the real world, a simulated case study can support authentic inquiry practices (Rutten et al., 2011). Some of the more compelling simulated cases are those that closely parallel factual situations (Warner, 2009). Percival and Ellington (1980) suggest that simulated case studies are formed by the intersection of pure case studies and pure simulations (Figure 1). Leigh and Kinder (2001) follow up by describing simulated case studies as being representative of aspects of real life and focused on highly specific skill-transferring tasks. Although simulated case studies for project-based assignments may contrive details, they are generally still based on an instructor’s practitioner experience and knowledge (Crockett & Foster, 2005).

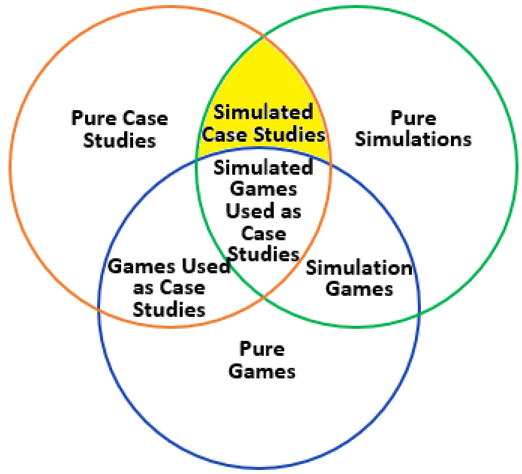

Figure 1. Intersection of Instructional Methods (Source: Percival & Ellington, 1980)

Simulated case studies are more conducive to a more disciplined and more manageable piece of reality than observing or doing real-world projects (Shulman, 1992). The development of simulated cases can help students apply specific skills by ensuring that the project case study addresses the required course objectives (Grundy, 1996). One of the greatest advantages associated with simulated case studies is that they provide an opportunity to incorporate specific learning objectives (Buckles & Redmer, 2004). Simulated projects, in effect, provide a controlled environment. Writers of simulated case studies are not constrained by the facts but can emphasize problems, issues, and situations to focus specifically on the problems they want to address (Warner, 2009). The author can control the scope and size of the project and can ensure that all required aspects are included.

Creating simulated cases alleviates the problem of limited access to live cases. Further, the project is not required to deliver a fully functional project upon completion, so the case could carry over to multiple courses. For example, a case may require a system that is analyzed and designed in one course, a database component that is developed in another course, and an implementation that could be developed in yet a third course. Developing an unfolding project in stages may be unacceptable with a live project with real clients.

A simulated project case may also be easier to administer because student questions are easier to anticipate if the instructor has developed the project details. Simulated projects are less stressful for students (Fleischmann, 2015) because no actual client is involved, eliminating concerns about making a bad first impression on a potential future employer.

To paraphrase Leigh and Collier (2015), we are not arguing that either real-world cases or simulated cases provide a better, less effective, or more effective learning experience. Both real-world cases and simulated cases have inherent advantages and disadvantages.

2.6 Case Development

Developing case studies can be challenging, but there are several useful resources to guide educators in their development. *Writing Cases* (Leenders et al., 2001) provides a step-by-step guide to assist educators in case development. The text provides an effective planning tool for case development. Cappel and Schwager (2002) provide a guide specifically for writing IS teaching cases. Their stated purpose is “to assist IS academicians and professionals who are interested in writing cases for publication.” Some of their most interesting observations are that a case should “feel ‘real,’ regardless of whether it is based on real or fictional circumstances.” Another recommendation is that a case should have a ‘hook’, which is a statement or short paragraph to grab readers’ interest and attention. Farhoomand (2004) provides another guide for writing teaching cases. He points out that the teaching value of a case can be increased by linking it to pertinent theories and concepts. Kim et al. (2006) developed a conceptual framework for developing teaching cases across disciplines. Sipior, Granger, and Farhoomand (2021) provide an excellent structure for a teaching case, along with an effective discussion of the teaching note.

2.7 Component-based Development

Developing high-quality simulated cases is not a trivial task, and once they are completed the question of reusability comes into play. Reflection on approaches for making reusability possible evoked software development techniques such as component-based development and modular design. Component-based development is a software system development methodology in which the system is developed using reusable components (Crnkovic et al., 2005). This approach originated with McIlroy (1968), based on the idea of component production such as that found in engineering fields (Vale et al., 2016). Component-based software engineering is based on constructing software systems from existing reusable components, with system evolution occurring through the customization and replacement of components (Szyperski, 2002).

2.8 Modular Design

Modular design has long been a staple of software development. A modular system is made up of smaller modules that fit together in a predefined architecture (Russell, 2012). Modularization involves the subdivision of complex processes into small tasks. Further, it enables reusability, that is, using existing components as building blocks to create new systems.

Component-based development and modular design are equally applicable to the development of simulated teaching cases. Simulated cases can be developed as a combination of multiple components or modules that, once combined, form a coherent whole. Once the case study modules are developed, they can be used in varying configurations multiple times. The modular approach for developing teaching cases is based solidly on software development concepts, including component-based development, modular design, reusability, scalability, extensibility, and componentization.

3 Research Design

This study applies action research (AR) to an educational problem of practice, carried out by educational practitioners (Baskerville & Wood-Harper, 1996; Checkland & Holwell, 1998; Davison et al., 2004, Mertler, 2021). AR is recognized as an effective tool for enhancing teaching methods (O'Connor et al., 2006; Arung, 2014; Lufungulo et al., 2021; Mertler, 2021). AR has been an accepted approach for conducting research in information systems since the 1990s (Baskerville, 1999; Baskerville & Myers, 2004; Wang & Luo, 2020) as it simultaneously links business practices and theoretical development (West et al., 1995).

The focus of AR is more on solving a problem efficiently and feasibly, rather than the intricate statistical scrutiny of quantitative research or the narrative explanations of qualitative research (Koshy, 2012; Lufungulo et al., 2021). The AR framework includes (1) the identification and investigation of problems or concerns recognized, (2) change designed and implemented in practice to address the problem or concern identified, and (3) the effect of change as observed and analyzed against pre-established measures to understand the impact of the change (Davey et al., 2016; Houghton et al., 2018).

AR is often approached as a multi-cycle process of the above sequential steps (Riel, 2007; Mertler, 2021), therefore, the observed outcome can guide the refinement of the design and implementation of change. AR is characterized by an ongoing and systematic interaction between the problem-solving cycle that addresses the difficulties of reusing case-based projects and the research cycle that develops new knowledge (Chiasson et al., 2008; Mckay & Marshall, 2001). Relying on this interaction and its longitudinal outcomes, along with a variety of studies concerned with lack of academic integrity (Araujo & Kyrilov, 2020; Brommer, 2015; Evans, 2016; Fraser, 2014; Lee et al., 2016; McCardle, 2010; Sorea & Repanovici, 2020; Swift & Nonis, 1998), two primary problems were identified:

1. Excessive time that it takes for instructors to develop or modify case-study descriptions.
2. Temptation for students to seek and use compromised case solutions.

The study began in 2012 and the most recent cycle took place in Fall 2022. There have been twelve cycles of action research into the problems under consideration. Each cycle resulted in reduced preparation and made it obvious when students were "recycling" case solutions from previous semesters. According to Mertler (2021), "Rigor can be enhanced by engaging in a number of cycles of action research into the same problem or question, where the earlier cycles help to inform how to conduct later cycles, as well as specific sources of data that should be considered" (p, 7). Mertler (2021) goes on to state that "with each subsequent cycle of action research, more is learned, and greater credibility is added to the findings" (p. 7). Over many cycles of refining this approach, a solution intervention was devised and fine-tuned to address both problems. These outcomes are not rigorous to the extent that they are generalizable, but they constitute ideas tested over twelve iterations from 2012 to 2022.

4 Modular Case Study Approach

This research describes a pragmatic approach to developing teaching case studies to maximize reusability and scalability. As the name indicates, a case study is assembled from a collection of modules. When developing a case study, the author writes a baseline case, along with a collection of modules, each of which describes an aspect of a business.

Modules can be added to or removed from the teaching case to reconfigure the case content each time it is used to tailor the case to reinforce specific course objectives, or even scale the case up or down per the desired time frame of the project or the desired level of difficulty (Figure 2). This approach improves the reusability of teaching cases over time, allows cases to be scalable, and provides extensibility, given that the modular approach is designed to allow the addition of new features. The myriad permutations of the swappable modules help to reduce student temptation to share solutions from previous sessions, compared to reusing static case studies with solutions that may have been compromised.

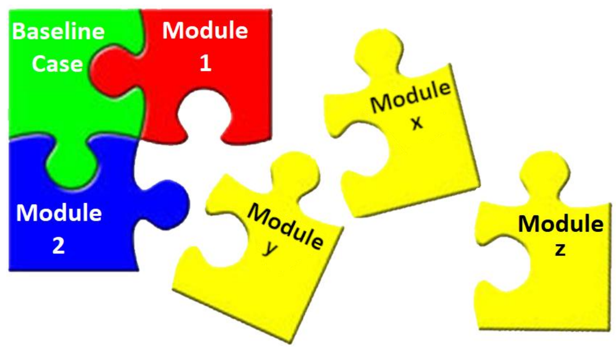

Figure 2. Baseline Case Description with Interchangeable Modules

With project-based cases, there are two primary components: the case study itself, which describes the case scenario, and the project deliverables that are linked to the teaching objectives of the course. The focus of this study is on developing the case study description, although an example of course-specific project deliverables will be discussed in the accompanying example.

Cappel (2002) describes some desirable qualities of the ideal written case:

> It should be interesting, realistic, and challenging to students, without being overly complex. A case should concern a type of organization and subject matter with which most students have some level of experience. This helps students in conceptualizing the problem and possible solutions.

Modules are designed for two primary purposes: to be compelling and to provide an additional business aspect to be analyzed. Each module has been researched in depth and describes a scenario designed to intrigue students. This is in line with the recommendation made by Sipior, Granger, and Farhoomand (2021) that "using a teaching case in the classroom should be compelling and fun" (p. 660). While the overall case presents a topic designed to captivate students, many modules focus on a scenario selected to increase student interest. For example, the author researched the sport of heli-skiing, which is off-trail, downhill skiing or snowboarding where the skier reaches the top of the mountain by helicopter, instead of a ski lift. Realizing that a smaller company like WRLRR would be unable to afford the expense of a helicopter, the author researched more affordable options and came up with the concept of gyrocopter skiing, as presented in one of the modules. Other module themes like wingsuits and ice surfing should be equally appealing. This approach conforms to guidance from Cappel and Schwager (2002) that a case involves a topic of interest.

Other module topics are less esoteric and instead focus on more practical matters, matters that may be encountered in many business environments. Some modules deal with repairs and maintenance, the sale of parts, and rentals. These are representative activities that can be found in other businesses. Cappel and Schwager (2002) point out that even in a case targeted to an information systems publication, addressing business functions is appropriate because "business functions today are more integrated than ever before" (p. 289). Each module provides details of the business processes required by the new case feature being addressed, which emphasizes business integration.

Regardless of whether the module is business-focused or is designed to add more captivating elements, every plugin module describes a business function that can be analyzed and synthesized by the students. A case should "promote both analysis and synthesis skills" (Sipior et al., 2021, p. 662). They define synthesis as "how to put related facts into a coherent, articulated plan of action" (p. 662). The supplemental modules reinforce and expand upon the core case's purpose of providing experiential learning opportunities for students. Each module provides an elaborate narration, with both relevant and irrelevant details.

Further, the plugin modules make it possible for the instructor to customize the length of the case. Cappel and Schwager (2002) explain that a case must be an appropriate length, but "there is no 'magic formula' conceptualizing how to determine how long a case should be" (p. 290). Cappel and Schwager (2002) also note that the length of a case is dictated by its objectives. With the modular approach, the instructor can expand or curtail both objectives and case length as appropriate.

Each plugin module has the following structure:

- One or more paragraphs providing a thorough description of a new sales item, service, or activity that will be offered.
- A bullet list of all customer-facing business processes required by the additional items to be sold, additional service, or additional activity that will be offered.
- A detailed description of all customer-facing business processes required by the new feature that will be offered.
- A bullet list of all back-office business processes required by the supplemental options that will be offered.
- A detailed description of all back-office business processes required by the additional features that will be offered.
- A bullet list of all additional business rules associated with the new sales item, service, or activity that can be offered.

This structure helps students understand the general purpose of the new sales item, service, or activity that will be offered. The module structure also provides "details about the specific issue(s) that the company faces" (Sipior et al., 2021, p.661) needed for analysis, design, and implementation. This level of detail is required because a case must provide adequate information to make it possible for students to break down the relevant facts (Farhoomand, 2004). "The use of specific details … tends to enhance realism. The inclusion of both relevant and irrelevant details also promotes analytical skills by requiring students to distinguish what is important from what is not" (Cappel & Schwager, 2002 p. 290).

Figure 3 demonstrates the outline of a case study that has been used in implementations of the modular case description concept in database design courses and systems analysis and design courses—two core courses in many information systems curricula (Davey & Kelly, 2005; Lenox & Woratschek, 2005). The structure was designed specifically for those courses but can be altered in any way desired by instructors to best serve their courses. The modular case description is made up of both core components and plugin modules. The plugin modules are annotated in Figure 3.

The following section will describe how the modules can be used to supplement the case. Two scenarios will be discussed: the initial time that the case is used, and subsequent uses of the case.

The initial use of the case is the most straightforward because the core business remains unchanged. This means that no additional sales items, services, or activities have been added to the core case in previous semesters. This discussion will first address that scenario.

The "Business Background" section begins with an "Overview" that describes the core business. This is followed by "Background Details." This subsection includes both "Essential Business Details" and "Supplemental Business Details." "Essential Business Details" provides details about what the business does. These are aspects of the business that are not intended to be changed. The "Supplemental Business Details" subsection appears next. However, if the core case has not been altered or supplemented, then the "Supplemental Business Details" subsection is not required.

Business Background

Overview
- Concise Core Business Scenario

Background Details
- Essential Business Details
- Supplemental Business Details (Plugin Module)

System Description

Listing of Business Processes
- Customer-Facing Business Processes
  - List of essential customer-facing processes
  - List of supplemental customer-facing processes (Plugin Module)
- Back-Office Business Processes
  - List of essential back-office processes
  - List of supplemental back-office processes (Plugin Module)

Descriptions of Business Processes
- Customer-Facing Business Processes
  - Description of essential customer-facing processes
  - Description of supplemental customer-facing processes (Plugin Module)
- Back-Office Business Processes
  - Description of essential back-office processes
  - Description of supplemental back-office processes (Plugin Module)

Drivers of System Modifications

Problems Encountered (Plugin Module)

Strategies Under Consideration (Plugin Module)

Requested System Description

Strategies Adopted (Plugin Module)

Listing of Requested Business Processes
- List of requested customer-facing processes (Plugin Module)
- List of requested back-office processes (Plugin Module)

Descriptions of Requested Business Processes
- Description of requested customer-facing processes (Plugin Module)
- Description of requested back-office processes (Plugin Module)

Business Rules

Rules Associated with Existing System
- Business rules for essential business system
- Business rules added for modules supplementing existing system (Plugin Module)

Rules Associated with Supplemental Modules for Requested System
- New rules associated with modules added to requested system (Plugin Module)

Figure 3. Basic Structure of an Example Modular Case

The “System Description” section provides details of the business processes that are featured in the current system. It includes both a “Listing of Business Processes” and “Descriptions of Business Processes.” Both of these sections include details about both customer-facing processes and back-office processes. This section provides details about all processes associated with the core business.

The third section in Figure 3 is “Drivers of System Modifications.” The first subsection is “Problems Encountered,” which provides an overview of problems encountered in the existing system. The “Strategies Under Consideration” subsection appears next, and it discusses strategies being considered to address those problems or to improve the strategic position of the company.

The following section is "Requested System Description." This description of the requested, or future, system is made up almost exclusively of plugin modules. The section first addresses "Strategies Adopted" by explaining the strategy to add new sales items, services, or activities in response to concerns brought up in the "Problems Encountered" subsection. This will involve the insertion of one or more module segments to provide details about the new feature to be added by the future system to achieve strategic objectives. This is followed by a subsection titled "Listing of Requested Business Processes" that lists both customer-facing and back-office business processes associated with each new feature requested for the future system. Finally, the "Descriptions of Requested Business Processes" subsection describes both customer-facing and back-office business processes associated with each new feature requested for the future system. The source of the details, listing, and description for each new feature is one of the available plugin modules.

Finally, in the "Business Rules" section, the "Business rules for essential business system" sub-subsection of the "Rules Associated with the Existing System" subsection as well as the "Rules Associated with Supplemental Modules for Requested System" will reflect the business rules associated with the core business as well as the business rules associated with additional modules required by the future system.

In the second scenario, the core business has evolved through the addition of sales items, services, or activities that have supplemented the business in previous semesters. Of course, once an instructor develops a case in one semester that adds new features to the business via material from plugin modules, he or she may logically want to consider the new core case to be an amalgam of the original core case plus features that were added in prior semesters as if it were a real-world company that adapted and evolved. That is how this approach has been used by the authors. This will typically necessitate changes only in the first two sections, with a possibility of slight changes required in the third and last sections.

In this scenario, the "Supplemental Business Details" subsection of the "Business Background" section provides an opportunity to insert or substitute modules. This presents module content that describes one or more additional sales items, services, or activities that were added to the core business in a prior semester that would now be listed as part of the evolving core. These can also be removed as the core business evolves.

In the "System Description" section, if any additional sales items, services, or activities supplemented the core business background in prior semesters via plugin modules, they must be explained, listed, and detailed in this section.

In the "Drivers of System Modifications" section, if any problems arose due to the addition of any sales items, services, or activities, then those must be explained in the "Problems Encountered" subsection.

The "Requested System Description" section is handled exactly as described in the simpler scenario.

In the "Business Rules" section, the "Business rules added for modules supplementing existing system" sub-subsection of the "Rules Associated with the Existing System" subsection may require the addition of any business rules associated with additional modules required by the future system. These, like all related material, are provided in each module describing a specific business feature.

Optimally, a set of modules is pre-written, making it relatively easy to swap modules in and out of the case structure. The reconfiguration of the case through the exchange of modules makes it apparent to students that while the case appears similar to those used in previous sessions, the cases are not identical. Those differences make it rather complex, if not impossible, to reuse solutions developed by students who preceded them.

5 A Working Example of a Module

Project-based cases, often encountered in courses such as database design, systems analysis and design, software development, networking, and programming, forego the discussion component found in non-project-based cases in favor of a project-focused element that applies students' theoretical knowledge to real-world scenarios to provide experiential and active learning. Requiring a more extended and sustained effort, the project solution is significantly broader than what is commonly encompassed in a non-project-based case, often addressing system requirements through the development of diagrams, charts, models, documentation, screen prototypes, or an entire system (Cappel & Schwager, 2002). The actual project deliverables, or "action items", that make up the project-focused element of the case are linked to the teaching objectives of the course.

5.1 Experiential Element

Regardless of the case study selected or composed, the experiential portion of a project-based case is often made up of several units designed to help address and assess course objectives. This section will describe examples of course-specific projects based on modular case descriptions.

The case study text provides sufficient details to support the experiential element, which in turn must be designed to meet course objectives and assessment goals. If project requirements are extensive and complex, the design can be made up of a series of deliverables (i.e., the measurable results of intermediate phases that make up a project). A series of deliverables gives students manageable “chunks” to process, permits those chunks to be assigned in close conjunction with relevant course content, and enables students to develop a more realistic system. The deliverables can be aligned with specific course objectives or even an integrative section of a capstone course.

For example, the deliverables for a database design course may include an entity-relationship model, a normalized relational schema, implementation of database tables, SQL query design, stored procedures and triggers, etc. Deliverables for a systems analysis and design course may include use cases and use case narratives, a class diagram, decision trees, activity diagrams, and sequence diagrams. Each of the deliverables should be associated with one or more course objectives. In educational projects, a deliverable is instrumental in learning how to apply concepts and methodological approaches related to a particular task (i.e., an educational objective is reinforced by the corresponding deliverable) (Piccinini & Scollo, 2006).

5.2 Case Study Element

The case study text provides an in-depth, detailed description of a particular case in a realistic context. Previous papers have focused on various simulated case studies, including *Teton Whitewater Kayak* (Parker, 2003), set in Jackson Hole, Wyoming, *Lost River Wind Riders* (Parker, 2005), set in the Lost River Range of Central Idaho, *Winter Wildlife Safaris* (Parker, 2016), set in Sun Valley, Idaho, and *Wind Riders of the Lost River Range* (Parker, 2024), also set in a high mountain range of the Rocky Mountains. These scenarios were selected to spark the interest of students due to the variety of exhilarating recreational activities in dramatic and beautiful destinations (Figure 4). This discussion will focus on applying the modular case development approach to the *Wind Riders of the Lost River Range* case (Parker, 2024).

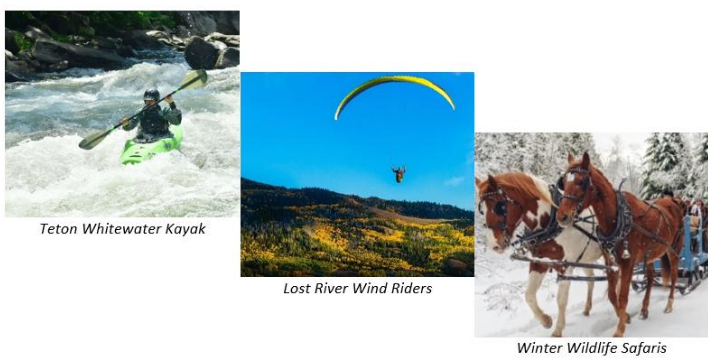

Figure 4. Previous Papers with Simulated Case Studies

The *Wind Riders of the Lost River Range* (Parker, 2024) was originally developed for use in both database courses and system analysis design courses, but it can easily be adapted for use in additional case-based project courses. Each segment of the case explained below corresponds to a section or subsection in Figure 3. A narrative explains the correspondence and is accompanied by a table that provides more detail about the structure shown in Figure 3, including the purpose of each element along with an example from the core case or a supplemental module.

The case sets the scene in the “Business Background” section by explaining “Wind Riders of the Lost River Range is a growing enterprise in central Idaho. The business sells wind-propelled sports equipment such as hang gliding, paragliding, and snowkiting equipment.” More details of this section from Figure 3 appear in Table 1.

Table 1. Section: Business Background

Subsection	Sub-Subsection	Purpose	Example
Overview		Describe the core business.	Wind Riders of the Lost River Range (WRLRR) is a specialty sports shop in central Idaho.
Background Details	Essential Business Details	Describe what the business does (Aspects that seldom change).	The company's primary focus is on the sale of hang gliding, paragliding, and snowkiting equipment
	Supplemental Business Details	Insert the segment of the corresponding module to describe one or more additional sales items, services, or activities that were selected to supplement the core business in a prior semester that would now be listed as part of the evolving core. These can also be removed as the core business evolves.	WRLRR recently expanded its product line by selling various lines of wingsuits. Wingsuit flying (or wingsuiting) is the sport of gliding through the air using a wingsuit which adds surface area to the human body to enable a significant increase in lift. WRLRR also offers training for novice wind riders, as well as advanced classes for those who wish to improve their skills before tackling greater challenges.

The core case begins the "System Description" section with a list of core customer-facing business processes followed by a list of core back-office business processes. These can be seen in Table 2, along with additional details about the "System Description" section. These items make up the current system of the core case and typically do not change, although they can be supplemented by additional customer-facing and back-office modules for additional reconfiguration.

For example, in one classroom-tested iteration of the case, the existing system was extended by plugging in the module for repair and maintenance services as well as the module for the sale of ultralight aircraft. The modules were added to make the case more challenging and to make alterations to the appearance of the case. In a later iteration, those two modules were removed, accompanied by an explanation that the owner had decided to discontinue sales of ultralight aircraft for a variety of reasons, as well as to terminate the repair and maintenance services over liability concerns.

Table 2. Section: System Description

Subsection	Sub-Subsection	Definition	Example
Listing of Business Processes	Customer-Facing Business Processes	Insert the segment of the corresponding module to list all of the customer-facing business processes associated with the core case as well as additional sales items, services, or activities selected to supplement the existing system.	• Customer purchases item(s) • Generate purchase agreement • Customer returns item(s) • Customer requests equipment service or maintenance • Generate service agreement • Generate service detail report • Generate service summary report
	Back-Office Business Processes	Insert the corresponding module segment to list back-office processes associated with the core case as well as additional sales items, services, or activities selected to supplement the existing system.	• Process returned inventory • Order new inventory • Process new inventory • Generate sales report • Generate current inventory report • Generate salesperson sales report • Generate technician service report
Descriptions of Business Processes	Customer-Facing Business Processes	Insert the corresponding module segment to describe customer-facing processes associated with the core case as well as additional sales items, services, or activities selected to supplement the existing system.	Generate Purchase Agreement A purchase agreement must be generated for every sale. As stated earlier, the employee prints two copies of the purchase agreement, which the customer signs. One copy is retained in a paper file and one copy is given to the customer as a receipt. The purchase agreement contains the following information: • Header Section: Current date/time, salesperson number, customer name, customer address, customer city/state/zip, customer phone, customer e-mail • Detail Section (for each item): Item

			number, description, serial number (if available), sales price • Footer Section: Sales subtotal, sales tax (see business rules), grand total, payment method, masked credit card number.
	Back-Office Business Processes	Insert the corresponding module segment to describe back-office processes associated with the core case as well as additional sales items, services, or activities selected to supplement the existing system.	Process New Inventory Assume that all orders placed to a given supplier are received in full. Further, also assume there is no need to make an allowance for partial shipments and backorders. 1. When a shipment is received, an employee enters the following information into the system for each item: • Serial number (if available) • Item number • Item category (selected from a product category list) • Cost • Purchase date 2. The system creates a new record in the inventory table for each item. Then it initializes additional fields as follows: • Item number is automatically generated by the system in sequential order and is used as an inventory tag. Generate Technician Service Report The technician service report can be generated for one or all technicians as needed. It includes a listing of all service jobs performed, including: • Header Section: Begin date, end date • Detail Section: Service performed, quantity performed, hours labor, labor cost, parts cost, total cost • Footer Section: Total performed, total hours, total labor, total cost, grand total

The future system allows even greater inclusion of interchangeable modules that detail strategic decisions that the company has decided to implement. For example, at the same time that maintenance services and ultralight sales were eliminated, the cable hang gliding module was selected for inclusion.

The inclusion of the cable hang gliding module begins in the "Drivers of System Modifications" section, as can be seen in Table 3. The "Strategies Under Consideration" subsection explains the reasoning behind the strategy being considered. The "Drivers of System Modifications" are described further in Table 3.

Table 3. Section: Drivers of System Modifications

Subsection	Sub-Subsection	Definition	Example
Problems Encountered		Insert the corresponding module segment to provide an overview of problems encountered in the existing system.	WRLRR has recognized an increasing demand for maintenance and repair services.
Strategies Under Consideration		Insert the corresponding module segment to discuss strategies under consideration to address those problems or to improve the strategic position of the company.	Nick considered several other services, including training lessons, adventure packages, renting wind sports equipment, or offering wind sports activities.

The background for this new activity, as seen below, is inserted in the "Requested System Description" section, more specifically the "Strategies Adopted" subsection in Table 4. This is followed by a list of the associated customer-facing and back-office business processes in the "Listing of Requested Business Processes" subsection. Next, the "Descriptions of Requested Business Processes" subsection in Table 4 provides details about the associated customer-facing and back-office business processes. More details about the "Requested System Description" section can be found in Table 4.

Table 4. Section: Requested System Description

Subsection	Sub-Subsection	Definition	Example
Strategies Adopted		Insert the corresponding module segment to provide an overview of a new sales item, service, or activity selected to be added by the future system to achieve strategic objectives.	During a childhood trip in his native Australia, Nick visited Tasmania and experienced cable hang gliding at Launceston's Trevallyn Dam and Geeveston's Eagle Hang Glider at Tahune Adventures. That experience got him hooked on gliding, and he wants to include it in the activities offered by WRLRR. Cable hang gliding is similar to zip-lining, but rather than a steel cable with a trolley from which a harness is suspended, you have a steel cable with a trolley from which a hang glider is suspended, along with a harness for the passenger. The entire flight is cable-controlled. The advantage of cable hang gliding is that it allows everyone to experience the thrill of hang gliding, even if they are unwilling to leap off a mountain in a regular hang glider. The securely cable-controlled hang glider allows anyone to experience the thrill of a sport normally enjoyed by only the most daring adventure lovers. Passengers are fitted with a harness by their instructor before experiencing the thrill of hang gliding under controlled conditions, soaring from a take-off platform across a valley to a landing site over a quarter of a mile away. At some points, the hang glider is around 200 feet off the ground. Soaring from the craggy peaks of the Lost River range, passengers will view the wilderness from the perspective of the golden eagle and experience the freedom of flight, with the safety of cable hang gliding. Guests will soar past talus slopes, over forests of Douglas fir, Engelman spruce, and subalpine fir, until dropping to a gentle landing among foothills covered in shrubs and grasses.
Listing of Requested Business Processes	Requested Customer-Facing Business Processes	Insert the corresponding module segment to list customer-facing business processes associated with the adopted strategies.	• Customer schedules cable hang gliding session
	Requested Back-Office Business Processes	Insert the corresponding module segment to list back-office business processes associated with the adopted strategies.	Generate upcoming cable hang gliding reservations report.
Descriptions of Requested Business Processes	Requested Customer-Facing Business Processes	Insert the corresponding module segment to describe customer-facing business processes associated with the adopted strategies.	Customer Schedules Cable Hang Gliding Session A customer may reserve a cable hang gliding session by telephone or in person. 1. Employee requests the customer's phone number to determine if the customer is on the customer list. a. If the phone number is not found, then the Cable Hang Gliding Reservation Form is displayed, a customer number is generated and assigned to the customer, and their personal information is entered and added to the customer file, including: ○ Name

			○ Address (include city, state, zip) ○ Phone number ○ E-mail address b. If the phone number is found, personal information is verified and the Cable Hang Gliding Reservation Form is displayed. 2. The customer specifies the following information, which is entered into the Cable Hang Gliding Reservation Form: a. Requested date b. Size of party c. Lodging Name d. Lodging Address (include city, state, zip) 3. The customer settles up. a. If the reservation is being made by telephone, the customer provides the following information to pay for the deposit as well as for any prior damages, and the information is entered into the Cable Hang Gliding Reservation Form: ○ Card type (if applicable) ○ Card number (if applicable) ○ Expiration date (if applicable) ○ CVV (if applicable) b. If the reservation is made in person, the employee requests payment for current charges. The customer provides cash or a credit or debit card. Credit or debit cards have a chip containing their ID, and when read by the card reader a data record is sent to the point-of-sale (POS) system. The POS transmits it and the amount due to the credit card processing system. If a credit or debit card is used then the following fields on the Hot Air Balloon Reservation Form are populated: ○ Card type (if applicable) ○ Card number (if applicable) ○ Expiration date (if applicable) ○ CVV (if applicable) 4. The employee taking the reservation saves a digital copy of the Cable Hang Gliding Reservation Form and the data is stored in the cable hang gliding reservations file. A Cable Hang Gliding Passenger Waiver is emailed to the customer, to be completed and submitted before accessing the take-off platform.
	Requested Back-Office Business Processes	Insert the corresponding module segment to describe back-office business processes associated with the adopted strategies.	Generate Upcoming Cable Hang Gliding Reservations Report The cable hang gliding instructor can generate an upcoming cable hang gliding reservations report for any given date. The report contains the following: ○ Header Section: trip date. ○ Detail Section (for each passenger): participant name, address (with city/state/zip), phone number, e-mail address, lodging name, lodging address. Footer Section: Total number in party.

Finally, the “Rules Associated with Supplemental Modules for Requested System” subsection of the “Business Rules” section appearing in Table 5 provides the associated business rules. Table 5 includes more details about the “Business Rules” section.

Table 5. Section: Business Rules

Subsection	Sub-Subsection	Definition	Example

Rules Associated with Existing System	Business rules for essential business system	Specify the business rules associated with the core business.	A partial list of business rules associated with WRLRR includes the following:
			• A general list of product categories includes:
			  ○ hang glider
			  ○ paraglider
			  ○ snowkite
			• Each inventory category can have multiples of each item in stock or none at all.
			• When a purchased item is not in stock, an order for that item must be placed within 3 hours.
			• Each specific inventory item, even those from the same inventory category, can have a unique price. These prices differ due to certain circumstances, such as price reductions, etc.
			• WRLRR does not sell parts directly.
			• The return period for purchases is ninety days.
			• Each non-return transaction will be charged a 7.5% sales tax.
			• The Payment Card Industry Data Security Standard (PCI DSS) guides organizations that handle branded credit cards from major card schemes. PCI DSS does not prohibit the collection of card verification codes/values before authorization of a specific transaction. However, card verification codes/values cannot be retained after the authorization of the transaction for which it was collected. Anyone who accepts credit cards in the United States agrees to comply with PCI DSS. While not a law, this industry standard is agreed to when a business contracts to accept credit cards. If a business accepts credit cards, it agrees to comply with PCI as part of the registration process. Some states have laws that support, to various degrees, PCI as a security and privacy standard.
			• An employee is assigned an employee number, which may be referred to as a salesperson number for salespersons. Designating employee numbers occurs beyond the scope of this system. Assume employees have one and use it when necessary.

	Business rules added for modules supplementing the existing system	Insert the corresponding module segment to specify the business rules associated with additional modules added to supplement the core business.	If gyrocopter skiing had been added to the core business in a previous semester, its associated business rules would appear here. • A reservation is typically required to ensure staff availability. • The waiver must be completely filled out and signed by each participant.

			• Gyrocopter skiing pricing: ○ Adults $1,500 per day ○ Children are not permitted.
Rules Associated with Supplemental Modules for Requested System		Insert the corresponding module segment to specify the business rules associated with additional modules required by the future system.	• A reservation is typically required to ensure instructor availability. • Cable hang gliding is suitable for all ages and abilities. • There is a weight limit of 350 pounds for safety reasons. • The waiver must be completely filled out and signed by each passenger or guardian before accessing the take-off platform. • Closed-toe shoes are required. • Cable hang gliding pricing: ○ Adults $25 ○ Children $20 A partial list of product categories includes: ○ hang glider ○ paraglider ○ snowkite ○ wingsuit

5.3 Module Design and Availability

The dynamic interchangeable strategic modules can be varied from class to class. Each module has been researched in depth and describes a scenario designed to appeal to students. Sufficient details are provided to allow both data and processes to be analyzed.

Sixteen modules have been developed over time to address a variety of business decisions impacting WRLRR, including:

- Sales: Ultralights
- Sales: Wingsuits
- Sales: Parts and accessories
- Service: Maintain equipment
- Service: Rent equipment
- Activity: Offer wind rider training
- Activity: Offer tandem flights
- Activity: Offer cable hang gliding
- Activity: Offer hot air balloon rides
- Activity: Offer gyro-skiing

The availability of multiple modules makes a large number of permutations and combinations possible. For example, 16 modules have been developed (in addition to the baseline module), so if four modules were regularly selected for each project, there would be 1,820 variations of the case. To make the process of inserting module content into the case study more efficient, macros have been developed and made available to automate the process.

6 Advantages and Challenges

The advantages of simulated cases were discussed earlier, and Table 1 recaps those advantages obtained from simulated cases developed using the modular case study development approach.

Advantages and challenges can be classified based on the design of the case study, resources required, from an academic perspective or a student perspective, and assessments and learning outcomes.

Table 6. Advantages of Simulated Cases Developed Using the Modular Case Study Development Approach

Advantage	**Description**
Engaging	The topic of the case study can be tailored to students' learning interests.
Directed	Can be designed to target particular experiential learning objectives.
Focused	Easier to elaborate on problems, issues, situations, and people to focus specifically on the problems that should be addressed.
Completeness	Makes it possible to ensure that all required aspects are included.
Tractability	Provides a more structured yet tractable scenario.
Adaptability	Enables more control over the scope and size of the project, depending on the duration of the course, student classification, and students' abilities and maturity.
Microcosmic	Presents a realistic scenario while ensuring a manageable scope; not required to deliver a fully functional project to a real client.
Accessibility	Alleviates limited access to appropriate live cases.
Achievability	May be less stressful for students due to enhanced manageability and no potential employer to possibly disappoint.
Predictability	Facilitates administration, due to the structured nature of the case; also allows student questions to be anticipated if the instructor has developed the case study.
Interrelatedness	The course objectives among related classes can be tightly integrated.
Additional advantages derived from the modular case study development approach	
Phased implementation	Various modules can be implemented in phases as needed, in an evolutionary manner.
Scalability	Modular design allows the instructor to scale size and complexity.
Extensibility	Modular design allows the addition of new capabilities or functionality via the addition of modules.
Reusability	Case modules can be interchanged, allowing case reuse while confounding solution recycling.
Modularity	Modular architecture contributes to clarity, simplicity, greater maintainability and modifiability, and divide-and-conquer benefits.
Componentization	Componentization involves breaking a system down into identifiable and reusable pieces that can be built and deployed independently, as well as recombined.
Customizability	Modules can be selected to emphasize different topics.
Efficiency	Can reduce instructor effort and time each semester.
Academic integrity	Alterations of case details lower incentives and opportunities for students to use solutions developed in previous semesters.
Targeted learning	New modules can be tailored to meet learning goals.

As noted, the modular case development approach came about in part as an attempt to provide case study reusability from the standpoint of the instructor, and to reduce the propagation of compromised solutions. Compromised solutions are a serious problem since some students see nothing wrong with either networking with former students to obtain solutions or searching online for past solutions. An Internet search for solutions associated with commonly used cases revealed easily accessible information such as instructor lecture slides providing tips for specific cases, case solution write-ups, student presentations, student blog posts discussing various cases, and case information posted on web communities such as Scribd and SlideShare (McCardle, 2010). Another study related that plagiarismdetection software revealed plagiarism in multiple instances of case study responses (Egleston, 2013). The "generation of own, written cases provides [...] the potential to reduce plagiarism through the use of unique cases in assessments" (Evans, 2016, p. 8).

The modular case development approach makes it possible to develop cases that both deter plagiarism and facilitate reuse. While academic dishonesty can never be eliminated, it can be greatly reduced by making it much more difficult to accomplish. Rotating multiple case modules each semester or quarter changes the content of a case enough to reduce student confidence in the usefulness of relying on previous solutions. As noted in Table 1, modularized case studies are also scalable, with size and complexity related to the number of modules that are included.

The modular simulated case development approach does not come without challenges. From the student perspective, providing students with a fully developed case description in software development courses does not present any opportunities for students to develop their skills in the crucial area of requirements gathering. Graduates entering the workforce often lack experience in requirements gathering, and thorough case descriptions do not serve to hone their skills in this area. A paper by Davey and Kelly (2005) explains one approach to address this shortcoming, as explained in the "Future Research" section.

From the instructor's perspective, developing detailed and realistic modules requires a significant up-front investment in time and effort. While dynamic modules can be developed over time, the "core case" must be developed well in advance of project assignment. Effective writing skills are essential when writing case studies because they must explicitly communicate what is intended for the case study to be successful in achieving the identified learning goals.

The case study descriptions must also be engaging and capture the imagination of the student. Student interest can be piqued by creating a storyline they can relate to, as well as individuals they might be able to identify with (Buckles & Redmer, 2004). Developing a simulated case that appears realistic presents a challenge, but provides the opportunity to create an effective teaching tool to enhance student learning.

7 Contributions

This study describes a modular case-study development process for designing realistic learning scenarios. The goal of formalizing this approach is to make a practical contribution to the practice of developing simulated cases with the benefits of helping to reduce an instructor's effort as well as making it less tempting for students to plagiarize. Some theoretical contributions have also been realized. The modular case-study development process addresses a need pointed out by Grimes (2019), who observes that "instructional strategies employing realistic scenarios and real-world learning opportunities lead to positive outcomes in student learning, [but] why are there so few evidence-based procedures for designing such activities?" (p. 139). This paper provides a specific design technique.

In addition, recent years have seen an increased demand from industry for universities to produce "ready-to-work" graduates (Baker, 2020). Employers from all areas of business are clamoring for graduates who are 'work-ready' (Spanjaard et al., 2018). Universities face an unreasonable demand to produce graduates with the profile needed by the industry (Tamrat, 2019). Universities worldwide are investing significant resources to address industry calls for work-ready graduates (Jackson et al., 2013). The term "work-ready" means that prospective employees' skills are aligned with employer skill requirements, including critical thinking, time management, teamwork, and interaction skills (ACT!, 2013). The most common institutional response to the demand for work-ready graduates is the off-campus internship (Hora, 2023). However, while some surveys report that 50-60% of all college students have taken an internship, others indicate a much lower participation rate of 21.5% (Hora et al., 2021). Fortunately, preparing work-ready graduates does not "have to be done through work or a job; long-term projects that take a semester or more to complete produce desirable workplace outcomes for graduates, too" (Busteed, 2020, para. 6). Carefully constructed experiential learning activities can equip students with real-world experiences (Spanjaard et al., 2018). While authentic learning environments are "not particularly easy to implement because of the time commitment and the difficulty in distilling a problem into an appropriate case or authentic environment in which to anchor instruction" (Jennings et al., 2005. p. 369), our study contributes to the literature by showing how complex teaching cases can be developed using reusable modules, thus helping to reduce instructor workload, addressing one of the major barriers to incorporating active learning/experiential learning in the classroom.

8 Lessons Learned

The modular approach for developing reusable cases emerged over time. It began with a core case and was followed up by modifications consisting of additions or deletions of business functions to vary the case from one semester to the next. The narratives of each of those changes were collected as part of each new version of the case that was saved. The idea of extracting details regarding each new feature and saving those additions as plugin modules evolved over several project iterations. It took years to recognize the opportunity presented by creating an archive of reusable plugin modules that encapsulated each of those case additions.

The following is a list of recommendations based on decades of teaching with project-based cases:

- Whether you develop a new case or adopt a case like WRLRR, treat the case as your own. Make modifications if errors are encountered, change things to better fit your course needs, and develop modules that you think would be useful. Case writers do not (or should not) develop a case with the expectation that it will be used “as is” for all eternity. If you adopt a case, make it work for your course needs.
- Keep an archive of all case revisions as well as their solutions.
- Develop a format for each module that integrates smoothly into the core case.
- Separate the case study and the assigned project deliverables into two files to make any necessary corrections easier.
- Discipline yourself to develop a solution as soon as revisions to the case are made. Be sure to develop the solution before assigning the case to the class.
- Immediately make modifications to the case to reflect errors or inconsistencies that students discover. Above all, do not be defensive when students discover errors. Instead, acknowledge their attention to detail in class.
- Once students have developed their initial model for a project-based case, I have found that providing students with a solution for the deliverables up to that point serves to level the playing field and makes consultations with the instructor run a bit more smoothly due to standardization.
- If a hypothetical business is used, such as WRLRR, think of it as if it were an actual business. Allow the core business to adapt and evolve through the addition of sales items, services, or activities that have supplemented the business in previous semesters. Consider the unfolding core case to be a fusion of the original core case along with the business functions that were added in prior semesters. This approach makes it easier for you to recall the current state of the business and has the added benefit of making the business more real to the instructor and to the students.
- Use the case in consecutive courses. It can be used in an analysis and design course, followed by a database course, followed by a software development course or a front-end development course. One drawback to this approach is encountered if students take a semester off and get out of sync with their cohort, but if the core case evolves each semester, the modified case, although further changed, will still make sense to those students who rejoin the course sequence.

9 Future Research

The inclusion of video components for requirements gathering can enliven a case and capture students’ imaginations. Davey and Kelly (2005) recommend the use of such tools in a systems analysis project to increase student engagement, particularly in the requirements-gathering phase. Because a fully developed case description is presented in simulated cases, there are limited opportunities for students to develop their skills in the crucial area of requirements gathering. Presenting fewer details in the case study and replacing them with video components would alleviate this shortcoming.

A related future research area focuses on the development of cognitive teaching aids based on artificial intelligence, machine learning, and natural language processing to train students to ask the necessary questions for requirements gathering and to ask those questions correctly. Tools such as IBM’s Engineering Requirements Quality Assistant (Lawton, 2020) and Alice, a cognitive business analysis assistant (Sharif, 2019), are available to identify problems in user stories and requirements. The concepts on which these tools are based could be applied to develop a tool that poses as a virtual client from whom students can elicit user requirements.

Additional research will be required to evaluate this approach from different perspectives. We would like to evaluate how the modular case study development approach is perceived by instructors from two perspectives. The first area of interest is developing a means of assessing the impact on instructors' preparation time regarding project planning over multiple semesters. The second is to test whether there is a perceived decrease in plagiarism when using modular cases compared to reusing non-modular cases. Instructors who tend to focus on academic honesty may be reluctant to test this because it would require that they first reuse the same non-modular case over multiple semesters and try to detect plagiarism. Increasing the likelihood of plagiarism to detect plagiarism would be a major impediment for many instructors unless they have no concerns about reusing cases. Other empirical tests can be designed to measure learners' satisfaction or to assess learners' career readiness as a means of assessing the effectiveness of this approach.

Further research should investigate whether the modular case study development approach can be applied to case-based research projects in other disciplines, such as management, marketing, health, law, engineering, etc. In addition to application to other disciplines, the applicability to the development of real-world cases should also be explored. Developing visual tools to manage modular case study development is being contemplated, as is a process model to provide step-by-step guidelines for developing modular case studies.

10 Summary and Conclusion

While project-based teaching cases provide an authentic learning component to many software development courses, finding real-world cases with needs that match course and project objectives is often not possible. Real-world cases may provide hands-on experience with only a subset of the topics and skills that a course project needs to reinforce. Simulated cases offer a more customizable alternative. However, simulated cases are difficult to conceive and develop, and compromised solutions due to student propensity for "sharing" can make it necessary to invest huge amounts of time each academic session developing new simulated cases from scratch.

Modular teaching case development offers a variety of advantages that make it possible to develop cases in phases, yet change them up adequately to confound student cheating. Further, such cases can be developed in such a way that student interest and engagement are maximized, as is the likelihood of realizing teaching and learning goals and ultimately better preparing our graduates to succeed.

References

ACT! (2013). *Work readiness standards and benchmarks*. Act.com. Retrieved from https://forms.act.org/research/policymakers/pdf/Work-Readiness-Standards-and-Benchmarks.pdf

Albanese, M., & Mitchell, S. (1993). Problem-based learning: A review of literature on its outcomes and implementation issues. *Academic Medicine, 68*(1), 52-81.

Ameta, D., Tiwari, S., & Singh, P. (2020). A preliminary study on case-based learning teaching pedagogy: Scope in SE education. In *Proceedings of the 13th Innovations in Software Engineering Conference*.

Araujo, G. G. & Kyrilov, A. (2020). Plagiarism prevention through project based learning with GitLab. *Journal of Computing Sciences in Colleges, 35*(10), 53-58.

Arung, F. (2014). The conceptual framework of classroom action research. *Journal of English Education, 24*, 3-15.

Baker, S. (2020). *Firms shift towards wanting 'work-ready' graduates*. Times Higher Education. Retrieved from https://www.timeshighereducation.com/news/firms-shift-towards-wanting-work-ready-graduates

Baskerville, R. L. (1999). Investigating information systems with action research. *Communications of the Association for Information Systems, 2*, 1-32.

Baskerville, R.L., & Myers, M. D. (2004). Special issue on action research in information systems: Making IS research relevant to practice-foreword. *MIS Quarterly, 28*(3), 329-335.

Baskerville, R. L., & Wood-Harper, A. T. (1996). A critical perspective on action research as a method for information systems research. *Journal of information Technology, 11*(3), 235-246.

Bednar, A., Cunningham, D., Duffy, T., & Perry, J. (1992). Theory into practice: How do we link? In T. Duffy & D. Jonassen (Eds.), *Constructivism and the technology of instruction* (pp. 17-34). Lawrence Erlbaum Associates.

Brodie, L., Zhou, H., & Gibbons, A. (2008). Steps in developing an advanced software engineering course using problem based learning. *Engineering Education, 3*(1), 2-12.

Brommer, S. J. (2015). Pedagogical strategies and training to avoid plagiarism. In *Advances in exemplary instruction: Proven practices in higher education* (pp.47-55). CreateSpace.

Brown, A. L. (1992). Design experiments: Theoretical and methodological challenges in creating complex interventions in classroom settings. *Journal of the Learning Sciences, 2*(2), 141-178.

Buckles, T. A., & Redmer, T. (2004). Case writing & teaching workshop: Comments and thoughts on the art of case writing. In *2004 Christian Business Faculty Association Conference*.

Busteed, B. (2020). *University academic leaders are losing confidence in student work readiness—and that's good news*. Forbes. Retrieved from https://www.forbes.com/sites/brandonbusteed/2020/01/23/university-academic-leaders-are-losing-confidence-in-student-work-readiness--and-thats-good-news/?sh=6e3e6a2828c7

Cappel, J. J. (2002). A systems analysis and design case: ABC church. *Journal of Information Systems Education, 12*(4), 233-243.

Cappel, J. J., & Schwager, P. H. (2002). Writing IS teaching cases: Guidelines for JISE submission. *Journal of Information Systems Education, 13*(4), 287-294.

Checkland, P. B., & Holwell, S. (1998). Action research: Its nature and validity. *Systemic Practice and Action Research, 11*(1), 9-21.

Chiasson, M. W., Germonprez, M., & Mathiassen, L. (2008). Pluralist action research: A review of the information systems literature. *Information Systems Journal, 19*(1), 31-54.

Cohen, R. F. & Menzies, T. (1994). Providing software engineering students with an experience in 'big computing'. In *Proceedings of the 1994 Software Engineering Education Conference*.

Crockett, M., & Foster, J. (2005). *Training the trainer resource pack*. The Archives-Skills Consultancy on Behalf of the ICA Section on Archival Education and Training. Retrieved from http://www.archive-skills.com/trainer-pack/p9.htm

Crnkovic, I., Larsson, S., & Chaudron, M. (2005). Component-based development process and component lifecycle. *Journal of Computing and Information Technology, 13*(4), 321-327.

Davey, B., & Kelly, D. (2005). System analysis education using simulated case studies. *Issues in Informing Science and Information Technology, 2*, 431-439.

Davey, B., Bozan, K., Houghton, R., & Parker, K. R. (2016). Alternatives for pragmatic responses to group work problems. *Informing Science: The International Journal of an Emerging Transdiscipline, 19*, 89-102.

Davison, R., Martinsons, M. G., & Kock, N. (2004). Principles of canonical action research. *Information Systems Journal, 14*(1), 65-86.

Egleston, D. O. (2013). The interactive, progressive case study. *Business Education Innovation Journal, 5*(1), 101-104.

Ellen, N. & West, J. (2003). Classroom management of project management: A review of approaches to managing a student's information system project development. *Journal of American Academy of Business*, *3*(1), 93-97.

Evans, C. (2016). Re-thinking case-based assessments in business management education. *The International Journal of Management Education, 14*(2), 161-166.

Farhoomand, A. (2004). Writing teaching cases: A reference guide. *Communications of the Association for Information Systems, 13*, 103-107.

Fertalj, K., Milasinovic, B., & Kosovic, I. N. (2013). Problems and experiences with student projects based on real-world problems: A case study. *Technics Technologies Education Management-TTEM, 8*(1), 176-186

Fisher, D., & Frey, N. (2007). Using projects and performances to check for understanding. In *Checking for understanding: Formative assessment techniques for your classroom* (pp. 79-97). Association for Supervision and Curriculum Development.

Fleischmann, K. (2015). Developing on-campus work-integrated learning activities: The value of integrating community and industry partners into the creative arts curriculum. *Asia-Pacific Journal of Cooperative Education, 16*(1), 25-38.

Fox, T. L. (2002). A case analysis of real-world systems development experiences of CIS students. *Journal of Information Systems Education, 13*(4), 343-350.

Fraser, R. (2014). Collaboration, collusion and plagiarism in computer science coursework. *Informatics in Education, 13*(2), 179-195.

Gallagher, S. A., Stepien, W. J., & Rosenthal, H. (1992). The effects of problem-based learning on problem solving. *Gifted Child Quarterly, 36*(4), 195-200.

Green, B. A. (2000). There but for fortune: Real-life vs. fictional case studies in legal ethics. *Fordham Law Review, 69*(3), 977-996.

Grimes, M. (2019). The continuous case study: Designing a unique assessment of student learning. *International Journal of Teaching and Learning in Higher Education, 31*(1), 139-146.

Grundy, J. C. (1996). Experiences with facilitating student learning in a group information systems project course. In *Proceedings 1996 International Conference Software Engineering: Education and Practice*.

Hamdy, H. (2015). Authentic learning in health professions education. In K. Abdulrahman, S. Mennin, R. Harden, C. Kennedy (Eds.) *Routledge international handbook of medical education* (pp. 128-143). Routledge.

Harris, R. (2009). A systems analysis and design semester project: A stand-alone project vs. a competitive project. *Information Systems Education Journal, 7*(11), 1-9.

Hayes, J.H. (2002). Energizing software engineering education through real-world projects as experimental studies. *Proceedings of the IEEE 15th Conference on Software Engineering Education and Training (CSEET.02)*.

Henze, N., & Nejdl, W. (1997). A web-based learning environment: Applying constructivist teaching concepts in virtual learning environments. In F. Verdejo & G. Davies (Eds.). *The virtual campus: Trends for higher education and training* (pp. 63-77). Chapman & Hall.

Hoffer, J. A., Ramesh, V., Topi, H. (2021). Modern database management (13th ed.). Pearson Education, Inc.

Hora, M. T. (2023). Career-readiness initiatives are missing the mark. *Inside Higher Ed*. Retrieved from https://www.insidehighered.com/views/2023/01/11/career-readiness-initiatives-are-missing-mark-opinion

Hora, M. T., Colston, J., Chen, Z., & Pasqualone, A. (2021). *National survey of college internships 2021 report*. University of Wisconsin-Madison Center for Research on College-Workforce Transitions. Retrieved from https://ccwt.wisc.edu/wp-content/uploads/2022/04/CCWT_NSCI-2021-Report.pdf

Houghton, R. F., Parker, K. R., Davey, B., & Bozan, K. (2018). Approaches for addressing student barriers to collaborative learning success. In J.S. Keengwe (Ed.), *Handbook of research on mobile technology, constructivism, and meaningful learning* (pp. 23-43). IGI Global.

Jackson, D., Sibson, R. and Riebe, L. (2013). Delivering work-ready business graduates—Keeping our promises and evaluating our performance. *Journal of Teaching and Learning for Graduate Employability, 4*(1), 2-22.

Jennings, M., Mawhinney, C. H., & Fustos, J. (2005) Case-based learning in computer information systems. Encyclopedia of Information Science and Technology. In M. Khosrow-Pour (Ed.), *Encyclopedia of Information Science and Technology* (pp. 368-372). Idea Group Reference.

Jones, B. F., Rasmussen, C. M., & Moffitt, M. C. (1997). *Real-life problem solving: A collaborative approach to interdisciplinary learning*. American Psychological Association.

Kerins, J. (2012). Assessment in systems analysis and design: Simulation, constraints and new foundations. *Innovation in Teaching and Learning in Information and Computer Sciences, 11*(1), 80-88.

Kim, S., Phillips, W. R., Pinsky, L., Brock, D., Phillips, K., & Keary, J. (2006). A conceptual framework for developing teaching cases: A review and synthesis of the literature across disciplines. *Medical education, 40*(9), 867-876.

Koshy, V. (2012). *Action research for improving practice: A practical guide*. Paul Chapman.

Lawton, G. (2020). *How AI can help with requirements analysis tools*. TheServerSide.com. Retrieved from https://www.theserverside.com/blog/Coffee-Talk-Java-News-Stories-and-Opinions/How-AI-can-help-with-requirements-analysis-tools

Lee, C. W. Y., & Chu, S. K. W., & Cheng, J. O. Y., & Reynolds, R. (2016). Plagiarism-free inquiry project-based learning with UPCC pedagogy. In *Proceedings of the Association for Information Science and Technology 2016*.

Leenders, M. R., Mauffette-Leenders, L. A., Erskine, J. A., (2001), *Writing cases*, (Fourth Edition). The University of Western Ontario.

Leigh, E., & Collier, K. (2015). Case study as - and within - simulation: A Mobius Loop for analysis and learning. In N. Courtney, C. Poulsen, & C. Stylios (Eds.) *Case based teaching and learning for the 21st Century*. Libri Publishing.

Leigh, E., & Kinder, J. (2001). *Fun & games for workplace learning*. McGraw-Hill.

Leland, N., & Nicholas, L. (2023). Increasing student engagement: Case-based teaching and learning. In *UNLV Best Teaching Practices Expo*.

Lenox, T. L., & Woratschek, C. R. (2005). The pros and cons of using a comprehensive final case project in a database management systems course: Marvin's magnificent magazine publishing house. *Information Systems Education Journal, 3*(24), 1-12.

Lewis, M. C., & Massingill, B. (2006). Graphical game development in CS2: A flexible infrastructure for a semester long project. In *Proceedings of the 37th SIGCSE Technical Symposium on Computer Science Education*.

Lombardi, M. (2007). Authentic learning for the 21st century: An overview. In D. Oblinger, (Ed.), *EDUCAUSE learning initiative*, *1*, 1-12.

Lufungulo, E. S., Mambwe, R., & Kalinde, B. (2021). The meaning and role of action research in education. *Multidisciplinary Journal of Language and Social Science Education, 4*(2), 115-128.

McCardle, M. (2010). The case of the compromised case: How the online posting of case solutions killed off Black and Decker™ and other great marketing cases. In *Marketing Management Association 2010 Fall Educators' Conference Proceedings*.

McFadden, F. R. & Hoffer, J. A. (1983). *Modern database management* (1st Ed.). The Benjamin/Cummings Publishing Company, Inc.

McIlroy, D. (1968). *Mass-produced software components, software engineering concepts and techniques* (1968 NATO Conference on Software Engineering). Van Nostrand Reinhold.

Mckay, J., & Marshall, P., (2001). The dual imperatives of action research. *Information Technology and People, 14*(1), 46-59.

Mertler, C. A. (2021). Action research as teacher inquiry: A viable strategy for resolving problems of practice. *Practical Assessment, Research, and Evaluation, 26*, 19.

Nilson, L. B. (2013). *Creating self-regulated learners: Strategies to strengthen students' self-awareness and learning skills*. Stylus Publishing, LLC.

O'Connor, K. A., Greene, H. C., & Anderson, P. J. (2006). *Action research: A tool for improving teacher quality and classroom practice*. East Carolina University.

Parker, K. R. (2003). A database design case: Teton whitewater kayak. *Journal of Information Systems Education (Special Issue on IS Teaching Cases), 14*(3), 271-274.

Parker, K. R. (2005). Lost River Wind Riders: A project for teaching database design. *Communications of the Association for Information Systems, 16*, 475-494.

Parker, K. R. (2016). Winter wildlife safaris. [Unpublished manuscript].

Parker, K. R. (2024). Winds Riders of the Lost River Range. *Communications of the Association for Information Systems*, *54*.

Percival, F., & Ellington, H. I. (1980). The place of case studies in the simulation/gaming field. In P. Race, & D. Brook (Eds.), *Perspectives on academic gaming & simulation* (pp. 21-30). Kogan Pages Limited.

Pérez, B., & Rubio, A. (2020). A project-based learning approach for enhancing learning skills and motivation in software engineering. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*.

Piccinini, N., & Scollo, G. (2006). Cooperative project-based learning in a web-based software engineering course. *Educational Technology & Society, 9*(4), 54-62.

Rhodes, A., Wilson, A., & Rozell, T. (2020). Value of case-based learning within STEM courses: Is it the method or is it the student? *CBE Life Sciences Education, 19*(3).

Riel, M. (2007). *Understanding action research*. Center for Collaborative Action Research. Retrieved from https://www.ccarweb.org/what-is-action-research

Russell, A.L. (2012). Modularity: An interdisciplinary history of an ordering concept. *Information & Culture, 47*(3).

Russell, J., Russell, B., & Tastle, W. J. (2005). Teaching soft skills in a systems development capstone class. *Information Systems Education Journal, 3*(19).

Rutten, N., van Joolingen, W.R., & van der Veen, J.T. (2011). The learning effects of computer simulations in science education. *Computers & Science Education, 58*(1), 136-153.

Savery, J. R. (2015). Overview of problem-based learning: Definitions and distinctions. In A. Walker, H. Leary, & C. E. Hmelo-Silver (Eds). *Essential readings in problem-based learning* (pp. 5-15). Purdue University Press.

Scardamalia, M., & Bereiter, C. (1991). Higher levels of agency for children in knowledge building: A challenge for the design of new knowledge media. *Journal of the Learning Sciences, 1*(1), 37-68.

Scott, E. (2006). Systems development group project: A real-world experience. *Information Systems Education Journal, 4*(23).

Sharif, A. (2019). *Meet Alice: Your cognitive assistant for business analysis*. Modern Requirements. Retrieved from https://www.modernrequirements.com/blogs/meet-alice-your-cognitive-assistant-for-business-analysis/

Sherman, C. A. (2000). The perfect systems analysis project. In *Proceedings of ISECON 2000*.

Shulman, L. (1992). Toward a pedagogy of cases. In J. Shulman (Ed.), *Case method in teacher education* (pp. 1-30). Teachers College Press.

Sipior, J. C., Granger, M., & Farhoomand, A. (2021). Writing a teaching case and teaching note: A reference guide. *Communications of the Association for Information Systems, 49*, 659- 667

Sorea, D., & Repanovici, A. (2020). Project-based learning and its contribution to avoid plagiarism of university students. *Investigación Bibliotecológica: Índice Acumulativo, 34*(85), 155-178.

Spanjaard, D., Hall, T., & Stegemann, N. (2018). Experiential learning: Helping students to become 'career-ready'. *Australasian Marketing Journal, 26*(2), 163-171.

Spurrier, G. & Topi, H. (2021). Systems analysis & design in an age of options (1st ed.). Prospect Press.

Surendran, K., Ehie, I. C., & Somarajan, C. (2005). Enhancing student learning across disciplines: A case example using a systems analysis and design course for MIS and ACS majors. *Journal of Information Technology Education, 4*, 257-274.

Swift, C., & S. Nonis, S. (1998). When no one is watching: Cheating behaviors on projects and assignments. *Marketing Education Review, 8*(1), 27-36.

Szyperski, C. (2002). *Component software: Beyond object-oriented programming* (2nd ed.). Addison-Wesley Longman Publishing Co. Inc.

Tamrat, W. (2019). *Work-ready graduates require strong partnerships*. Inside Higher Ed. Retrieved from https://www.insidehighered.com/blogs/world-view/work-ready-graduates-require-strong-partnerships

Tan, J., & Phillips, J. (2003). Challenges of real-world projects in team-based courses. *Journal of Computing Sciences in Colleges, 19*(2), 265-277.

Tan, J., & Phillips, J. (2005). Real-world project management in the academic environment. *Journal of Computing Sciences in Colleges, 20*(5), 200-213.

Thomas, J. W., Mergendoller, J. R., & Michaelson, A. (1999). *Project-based learning: A handbook for middle and high school teachers*. The Buck Institute for Education.

Thomas, J.W. (2000). A review of research on project-based learning. *The Autodesk Foundation*.

Topi, H., Valacich, J. S., Wright, R. T., Kaiser, K., Nunamaker, Jr., J. F., Sipior, J. C., & de Vreede, G. J. (2010). IS 2010: Curriculum guidelines for undergraduate degree programs in information systems. *Communications of the Association for Information Systems, 26*, Article 18.

Trier, J. (2010). Designing a case study. *Multicultural Education, 17*(4), 49-56.

Vale, T., Crnkovic, I., de Almeida, E. S., Silveira Neto, P. A. D. M., Cavalcanti, Y. C., & Meira, S. R. D. L. (2016). Twenty-eight years of component-based software engineering. *Journal of Systems and Software, 111*, 128-148.

van Vliet, P. J. A., & Pietron, L.R. (2006). Information systems development education in the real world—A project methodology and assessment. *Journal of Information Systems Education, 17*(3), 285-293.

Wang, W., & Luo, L. (2020). Systematic review of action research in management information systems field. *International Journal of Academic Research in Business and Social Sciences, 10*(6), 104-117.

Warlick, D. (1999). *Raw materials for the mind*. The Landmark Project.

Warner, C. (2009). *How to write a case study*. Retrieved from http://www.charleswarner.us/howwrite.html

West, D., Stowell, F. A., & Stansfield, M. H. (1995). Action research and information systems research. In Ellis, K., Gregory, A., Mears-Young, B., and Ragsdell, G. (Eds.), *Critical issues in systems theory and practice.* Plenum Press.

About the Authors

Kevin R. Parker. Dr. Parker is a Professor Emeritus of Informatics and the Founding Chair of the Department of Informatics and Computer Science at Idaho State University. He has an extensive professional and academic background that spans both Information Systems and Computer Science. He has both a Bachelor's and Master's degree in Computer Science, and a Ph.D. in Information Systems. He has taught courses in Information Systems, Computer Science, Information Technology, and Geographical Information Systems. Parker's research interests include business intelligence, improved teaching of core IT courses, impact of developments in information systems on curriculum, and emerging systems development issues. Dr. Parker holds a B.A. in Computer Science from the University of Texas at Austin (1982), an M.S. in Computer Science from Texas Tech University (1991), and a Ph.D. in Information Systems from Texas Tech University (1995).

Chang-Tang Chiang. Dr. Chiang is an Associate Professor in Information Management Department at Chinese Culture University in Taiwan. He earned his Ph.D. degree from Fu Jen Catholic University, Taiwan and a Master of Science degree from St. Louis University in USA. His primary research interests mainly focus on the impact of IT on sustainability and IT education. His papers have been published in International Journal of Educational Technology in Higher Education, Business Strategy and the Environment, Sustainability, Journal of Marketing Management, Journal of Electronic Commerce Research, Journal of Business & Industrial Marketing, and Electronic Commerce Research and Applications.