Feedback at Scale: Designing for Accurate and Timely Practical Digital Skills Evaluation

Gabriele Piccoli[ab]*, Joaquin Rodriguez[a], Biagio Palese [c] and Marcin Lukasz Bartosiak[b]

[a]*E. J. Ourso College of Business, Louisiana State University, Baton Rouge, LA, U.S.A.;*
[b]*Department of Economics and Management, University of Pavia, Pavia, Italy;*
[c]*Operations Management and Information Systems, Northern Illinois University, DeKalb, IL, U.S.A.*

* Stephenson Entrepreneurship Institute, 2219 Business Education Complex South, 501 South Quad Drive, Baton Rouge, LA 70803, USA;
Email: gpiccoli@lsu.edu

This is an Accepted Manuscript version of the following article, accepted for publication in EUROPEAN JOURNAL OF INFORMATION SYSTEMS. Piccoli, G., Rodriguez, J., Palese, B., & Bartosiak, M. L. (2020). Feedback at scale: designing for accurate and timely practical digital skills evaluation. European Journal of Information Systems, 29(2), 114-133.

Feedback at Scale: Designing for Accurate and Timely Practical Digital Skills Evaluation

Abstract

The global demand for digital proficiency has resulted in increasing pressure to "massify" education. As practical digital skills development becomes more important, there is a need to design accurate and timely performance feedback systems that can scale to a large number of learners. This paper contributes meta-requirements and design principles for designing a socio-technical artefact that offers a solution to the general problem of providing performance feedback at scale. The artefact evaluation provides interesting results for achieving the three objectives of a) scalability to a large number of learners, b) validity and reliability of the feedback, and c) positive impact on learners' behaviour and engagement with the feedback system. These results are obtained through the synergistic contribution of pedagogical prioritization (i.e., what skills to cover), assignment design (i.e., what tasks to use to evaluate mastery) and automated measurement (i.e., grading engine functionalities for error detection).

Keywords: design science research, feedback systems, digital skills evaluation

Introduction

In 2001 Peter Drucker claimed that the most striking growth in the ranks of knowledge workers would be in what he called "knowledge technologists." For these individuals, regardless of whether they perform manual or knowledge work, computer proficiency would be paramount (Drucker, 2001). As the trend toward the increasing digital mediation of everyday activities continues (Yoo, 2010), individuals are challenged to acquire an increasing number of digital skills. Digital skills are "the abilities of operating digital media, handle the structures of new media, search, select, process, and evaluate information in digital media and use digital media as a means to reach a particular goal." (Dijk & Deursen, 2009, p. 291). The importance of digital skills is evident in the ranks of knowledge technologists. More importantly, digital skills are

also critical for that portion of the workforce that does not have or need a college degree – so-called middle-skill jobs. Middle-skill jobs, those requiring specialized training above the high school level but short of a college degree, are increasingly "digitally intensive" and in advanced economies, they account for about two-thirds of the labour pool. An analysis of 27 million jobs posted in the US in 2016 shows that 82% of those jobs required digital skills, with high proficiency in spreadsheets and word processing being the baseline requirement for 78% of them (Burning Glass Technologies, 2017). More generally, recent research suggests that the acquisition of new skills through learning is perhaps the only answer to the looming massive displacement of workers through automation (McAfee & Brynjolfsson, 2016).

The global demand for formal learning has resulted in increasing pressure to "massify" education. In traditional institutions, because of financial pressure and calls to increase the efficiency of education, this trend results in increasing class sizes (Chambliss & Takacs, 2014). New alternatives, such as Massive Open Online Courses (MOOC) and Virtual Learning Environments (VLE) have emerged. While greater access to education is certainly a positive global trend, recent research has recognized the need to create efficiencies in teaching (Dean, Lee-Post, & Hapke, 2017) as there is widespread consensus on the fact that both the learning and teaching experience degrade as the student-instructor ratio increases (Bandiera, Larcinese, & Rasul, 2010). Reflecting this consensus, the student-teacher ratio is a key measure in national and international university accreditations and rankings. A comprehensive analysis of 760,000 college students across disciplines and levels shows that larger classes are associated with lower subject matter mastery (Benton & Pallet, 2013; Kokkelenberg, Dillon, & Christy, 2008; Sapelli & Illanes, 2016). Among the reasons for those results is the fact that students in large classes remain mainly anonymous, leading to a

degradation of interaction and engagement as well as a decrease in the quality of the feedback students receive (Chambliss & Takacs, 2014; Cuseo, 2007). Feedback describes the "information provided by an agent (e.g., teacher, peer, book, parent, experience) regarding aspects of one's performance or understanding" (Hattie & Timperley, 2007, p. 81). Feedback systems are designed to reduce the discrepancy between current and expected performance and behaviour, with timely and precise feedback being a critical element of any learning system (Hattie & Timperley, 2007; Kluger & DeNisi, 1996). However, as the student-teacher ratio increases, it becomes harder for instructors to provide frequent and timely quality feedback to learners (Cuseo, 2007). In other words, the search for efficient delivery of a college education leads to a decrease in student-teacher interaction (Piccoli, Rodriguez, Palese, & Bartosiak, 2017), feedback frequency, and the type of feedback provided (Cuseo, 2007).

This paper reports on the design, implementation, and evaluation of a socio-technical (ST) artefact (Gregor & Hevner, 2013) intended to address the problem of providing performance feedback for digital skills mastery at scale. While set in the context of an introductory Information Systems college course for residential students, our research provides insight for the more general class of problems of designing and implementing a scalable, yet accurate and timely, performance feedback system for practical digital skills mastery. Grounded in intervention theory (Argyris, 1970), we develop a set of meta-requirements and design principles to guide the development and implementation of the ST artefact. The results of our evaluation provide "proof-by-demonstration" (Nunamaker, Chen, & Purdin, 1990, p. 98) that the ST artefact achieves three objectives: a) scalability, b) validity and reliability, and c) positive influence on learners' behaviour. Specifically, we show that the performance feedback system a) achieves a two orders of magnitude improvement in grading speed over human graders,

thus laying the groundwork for almost infinite scalability and real-time performance feedback; b) it provides reliable and valid evaluation of learners' performance, producing an incidence of false positive and false negative errors below human graders (3.65% or less in all evaluations); and c) it stimulates an improvement in learners' engagement with optional practice assessments.

The proposed artefact is the result of multiple iterations in the build and evaluate cycle (March & Smith, 1995) of Design Science Research (DSR). We conceptualize the feedback system as a socio-technical (ST) artefact because it is the combination of technical and social features that brings about the outcomes of its use (Silver & Markus, 2013). Thus, the feedback system comprises an IT core (i.e., a custom-developed grading engine)[1] working in synergy with experts (i.e., teaching assistants) and a carefully designed set of learning and performance feedback resources.

Our results show that a grading engine, the IT core, is not enough to build a scalable performance feedback system for practical digital skills mastery. Instead an ST artefact designed to perform the evaluation must integrate pedagogical considerations (i.e., what skills learners should master), assignments design (i.e., what tasks can best test whether mastery has occurred) and measurement (i.e., code functionalities to detect accurate task completion in the learner's file). Our immediate contribution is "to solve a client's specific problem by building a concrete artefact in that specific context" (Iivari, 2015, p. 107). From the work, we then distill some early "prescriptive knowledge to […] address a class of problems" (Iivari, 2015, p. 107). Thus, we claim an

[1] The code for the grading engine is available at:

https://github.com/digitaldatastreams/grader

"improvement knowledge contribution" focused on "developing new solutions for known problems" (Gregor & Hevner, 2013, p. 345).

The paper is organized as follows. In the introduction, we identify the class of problems and the motivation for our work. The next two sections briefly describe the kernel theory and the DSR methodology providing overall guidance to the work. The following section describes the design and development of the ST artefact, linking the overarching kernel theory underpinning the meta-requirements (MR) and design principles (DP) of the proposed ST artefact. The paper then reports the results of the artefact's formal evaluation and concludes with a discussion of the implications of the findings for future research.

Theoretical Foundations

Intervention theory provides the general framework for our research program. The theory explains how intervenors can help their clients (i.e., individuals or organizations) to "become more effective in problem-solving, decision making and decision implementation" (Argyris, 1970, p. 15). Three principles guide the design of interventions: leveraging valid and useful information, allowing free and informed choice by the client, and fostering internal commitment. Valid information is that which can be verified and has been shown to affect the phenomena the intervenor is seeking to influence. Useful information is that which the client can leverage to "control their destiny." Free informed choice points to the centrality of the client in the implementation of the intervention – and therefore in its design. A free and informed choice is particularly important in situations like learning, where internal commitment is a precondition to the success of the intervention. Internal commitment is the degree of ownership and responsibility the client feels with respect to the intervention. The power of internal commitment comes from individuals' sense of purpose for the initiative and

their beliefs about the control they exert over their actions and the outcomes (Argyris, 1970).

The three principles of intervention theory are interdependent. The availability of valid and useful information is necessary for the client to make decisions that are free and informed. At the same time, the outcome of these decisions provides information that contributes to the stock of valid and useful information available to the client and the intervenor. Moreover, to the extent that the results of choices being made by the client are positive, those choices should strengthen internal commitment (Argyris, 1970).

Methodology

This research project applies the Design Science Research Methodology (DSRM) process model (Peffers, Tuunanen, Rothenberger, & Chatterjee, 2007): problem identification and motivation, solution definition, design and development, demonstration, evaluation, and communication. Design science research is by its very nature a cyclical approach whereby the investigators search the problem space for optimal solutions (Simon, 1996). The DSRM process model emphasizes the role of design iteration by showing how evaluation feeds back into both the definition of the objectives and the design and development phases of a research project. Thus, each repetition of the design-build-evaluate cycle (Hevner, March, Park, & Ram, 2004) provides feedback and enables the researcher to accumulate theoretical and practical knowledge (Baskerville, Kaul, & Storey, 2017).

In this article, we report the results of a Design Science Research (DSR) project spanning three iterations, starting with intervention theory (Argyris, 1970) as the kernel theory providing overall guidance to the work. Given the cyclical nature of DSR, we describe early design decisions and a first iteration through the design-build-evaluate

cycle (i.e., pilot implementation). We leverage the results of this effort, coupled with our kernel theory, to inform the development of meta-requirements and design principles (Walls, Widmeyer, & El Sawy, 1992) for a second design iteration of the ST artefact (Figure 1). We then demonstrate how the artefact was used to address the specific problem of providing accurate and timely performance feedback to learners in a freshmen college course that concentrates on intermediate Microsoft Excel and Word digital skills. The evaluation of the ST artefact did not lead to the discovery of new requirements or principles. However, it offered indications for refinements that were implemented during a third iteration that was subsequently deployed at scale with 310 concurrent enrolled users. The context of our demonstration is an instance of the general class of feedback at scale problems, thus enabling a realistic and reliable assessment of the ST artefact.

[Figure 1 near here]

Each iteration ends with a summative naturalistic evaluation (Venable, Pries-Heje, & Baskerville, 2016) as the ultimate goal of the research is to assess the performance feedback system in a real organizational environment (i.e., a semester-long course in a university setting). Thus, each evaluation episode measures the scalability, reliability, and validity of the ST artefact. Scalability refers to the feedback system's ability to accommodate a large number of users. Reliability refers to the dependability of the artefact – its capacity to perform in a failure-free manner under a range of realistic use conditions (Baskerville et al., 2017). Validity represents "how well the artifact performs" (March & Smith, 1995, p. 254) – in our case in terms of performance feedback to the learners.

In the context of this study, the reliability of the ST artefact can be claimed if the performance feedback system consistently marks the same tasks under different circumstances. We use a diachronic approach (Baskerville et al., 2017) by analysing the

artefact's performance with different users on different problems. The validity of the ST artefact can be claimed if the performance feedback system correctly identifies instances of mastery of specific practical digital skills while avoiding both false positives and false negatives. False positives occur when the learners receive positive feedback, but they did not master the skill (e.g., a student did not complete a task correctly, but the task is marked as successfully accomplished). False negatives occur when the learners receive negative feedback even though they mastered the skill (e.g., a student did complete a task correctly, but the task is marked as incorrectly accomplished).

Solution definition

In the context of providing accurate and timely feedback to the learners, intervention theory calls for assembling valid and useful information about performance and enabling the client to exert free and informed choice about performing the work.

Furthermore, the class of problems we are tackling requires evaluation and feedback provision at scale, thus narrowing the solution space (Simon, 1996) to an ST artefact that accommodates an arbitrary number of learners. Assembling valid and useful information in support of the intervention requires the design and implementation of an ST artefact that integrates both practice assignments and the evaluation system to ensure that

(1) Tasks are realistic. Realistic tasks are representative of the practical digital skills learners are expected to master in their job as knowledge workers. Completion of realistic tasks provides valid information about skills acquisition.
(2) Task evaluation is reliable. Reliable evaluation occurs when the feedback system consistently marks the same tasks under different circumstances. Reliable

evaluation of the tasks is necessary to provide valid information to the learners and the intervenor.

(3) Task evaluation is valid. Valid evaluation occurs when the feedback system correctly avoids both false positives and false negatives in task evaluation. Valid evaluation is a prerequisite to provide useful information.

In the commercial space, there are products from major publishers that focus on basic digital skills acquisition in the context of Microsoft Office applications. After evaluating the most popular course solutions and using one of them during the first redesign of the course (semester 1), we uncovered major limitations that clashed with the tenets of intervention theory and with our ST artefact design objectives. To simplify course delivery and evaluation of task performance, these systems carry out the bulk of student learning in browser-based simulated environments rather than in the software application being learned (e.g., Microsoft Excel). Since the time spent in the application environment is related to proficiency and self-efficacy (Piccoli, Ahmad, & Ives, 2001), task simulation in the browser limits skill mastery. It constrains the type of mistakes and "alternative solutions" the student can practice or discover. Thus, students do not develop an overall conceptual understanding of how to use the application to achieve their goals (e.g., dynamic analysis in Excel, standardized and efficient document design in Word). Moreover, the simulated environment constrains learners to practice a series of individual independent tasks, resulting in learning that is formulaic and disjointed. Such a simulated environment is not a realistic representation of the way in which learners will use Microsoft Office applications during their college or professional careers. More generally, research shows that narrow performance feedback and the repetitive practicing of individual tasks leads to trial-and-error strategies rather than

stimulating the cognitive effort that leads to higher level learning (Hattie & Timperley, 2007).

Given the limitation of existing products, we deemed it impossible to use them as the IT core for a scalable ST artefact to deliver accurate and timely performance feedback for practical digital skills mastery. Instead we designed and developed a custom-made solution to automatically evaluate students' performance in practice assignments completed within Microsoft Office applications. In the next section, we introduce the meta-requirements and the design principles guiding the artefact's development.

Design: Requirements and Design Principles

The three tenets of intervention theory, collecting valid and useful information, enabling free and informed choice by the client, and fostering internal commitment provide the overarching architecture for deriving meta-requirements and design principles for the design and development of the ST artefact. The design principles capture the knowledge to guide the creation of "other instances of artefacts that belong to the same class" (Sein, Henfridsson, Purao, Rossi, & Lindgren, 2011, p. 39). Following the above solution definition, the performance feedback system focused on balancing task realism with the ability to provide reliable and valid feedback to the learners.

Collect Valid and Useful Information

According to intervention theory, information is valid when it is shown to affect the outcome while being independent of it. For this reason, the ST artefact must collect, or generate, the data necessary to provide accurate and timely feedback. Information is useful when it is actionable. For example, while natural aptitude is linked to skill mastery, it is not actionable as it cannot be easily changed. Conversely, performance on

assignments provides both valid and useful information since it can inform future behaviour. Thus, a performance feedback system for practical digital skills must gather both performance and behavioural data.

MR1: A performance feedback system for practical digital skills accurately records learner behaviours and performance.

- DP1.1: Collect completion data for required and optional assignments.
- DP1.2: Collect learning resources utilization data (online or in person) for required and optional resources.
- DP1.3: Reliably measure performance ensuring consistent evaluation of the same task over time and across learners.
- DP1.4: Produce valid measures of skill mastery (i.e., minimizing false positive and false negative evaluation errors).

Enable Free and Informed Choice

The second principle of intervention theory is free and informed choice intended to empower learners' own decision making. In the specific context of college courses, free and informed choice is a trait of effective teachers who "avoid using grades to persuade students to study" (Bain, 2004, p. 36). Instead they motivate the importance of the material while deliberately giving the students a sense of control over their learning outcomes.

The literature identifies different types of feedback based on the level at which they are aimed. Feedback can be directed at performance on the deliverable (i.e., task or product), at the process used to complete the task/product, or alternatively at the individual in a way unrelated to the specifics of the task (Hattie & Timperley, 2007). Meta-analytic research about performance feedback interventions reports a generally

positive, albeit not univocal, effect of feedback aimed at task performance (Kluger & DeNisi, 1996). Specifically, feedback interventions that focus attention toward task-motivation or task-learning processes, such as task-specific comments, tend to improve performance by focusing attention on the skill to be mastered. Conversely, feedback interventions that direct attention to meta-task processes, such as generalized praise or overall grades, attenuate the intended positive effects of the intervention by directing attention to the self (Hattie & Timperley, 2007; Kluger & DeNisi, 1996).

Previous research also shows how formative assessments, those specifically intended to generate feedback on performance to improve learning (Sadler, 1998), should be designed on the assumption that learners are self-regulated entities (Nicol & Macfarlane-Dick, 2006). This approach, which is in line with the basic tenets of intervention theory, calls for the design of performance feedback systems that provide learners with performance feedback as a way to help them improve their learning process and self-regulation abilities (Hattie & Timperley, 2007). Note that while grades can serve as a form of performance feedback, it is important to separate feedback from assessment and to ensure that assessment is closely related to learning objectives (Hattie & Timperley, 2007). Thus, a performance feedback system for practical digital skills should de-emphasize grades and focus the evaluation on skills acquisition while enabling informed choice.

MR2: A performance feedback system for practical digital skills does not conflate behaviour with learning.

- DP2.1: Treat assignments and homework as a service to learners and ensure that they do not impact the individual's learning assessment.
- DP2.2: Measure learning independently of learner behaviour by relying only on dedicated ad-hoc evaluations (i.e., exams).

MR3: A performance feedback system for practical digital skills treats learners as self-responsible units and maximizes learner control.

- DP3.1: Provide regular homework and practice assignments to learners to test their progress in skills acquisition.
- DP3.2: Provide feedback to learners for all the assignments learners voluntarily submit.
- DP3.3: Direct learners toward appropriate resources, online or physical, for specific skills that are not mastered.

MR4: A performance feedback system for practical digital skills contextualizes behavioural and performance data for learners as soon as it becomes available.

- DP4.1: Provide feedback immediately after task completion.
- DP4.2: Enable learners to interpret the feedback by contextualizing it appropriately.

Foster Internal Commitment

The third principle of intervention theory is internal commitment. Internal commitment refers to the degree of ownership and responsibility the client feels with respect to the intervention. The power of internal commitment comes from individuals' sense of purpose for the initiative and their beliefs about the control they exert over their actions and the final outcome. Thus, a performance feedback system for practical digital skills must eliminate obstacles to skills mastery while enabling learners to make tangible progress toward it.

The progress principle suggests that emotional well-being and motivation at work or during task completion depends primarily on the perception of progress toward

a meaningful goal (Amabile & Kramer, 2011). The progress principle is grounded in the more general theory of small wins. A small win is a "concrete, complete, implemented outcome of moderate importance" (Weick, 1984, p. 43). The theory has wide applicability beyond formal learning environments, and it posits that task activities engendering a series of controllable successes of modest size combine to create progress and institutionalize change (Reay, Golden-Biddle, & Germann, 2006). At the individual level, the design of tasks that favour small wins enables people to sustain effort over time by producing visible progress toward the goal (Amabile & Kramer, 2011) – in our case, skills mastery.

MR5: A performance feedback system for practical digital skills fosters a sense of progress in the learners.

- DP5.1: Create realistic practice exercises that learners view as instrumental to their future success.
- DP5.2: Create manageable practice exercises that fit easily within the schedule and habits of the learners.
- DP5.3: Create practice exercises that are limited in scope and enable learners to focus on specific skills and receive precise feedback.

In the context of the introductory college course, the primary redesign called for by the refined meta-requirements is that assignments are manageable. Manageable projects are those that, based on current learners' ability at any specific point in time during the learning journey, are successfully completable. Designing assignments that are too simple (or complex) creates an artificial floor (or ceiling) to the ability to provide actionable feedback to the learner. More importantly, manageable exercises are more likely to produce small-wins and foster increased completion.

Demonstration: Artefact Development and Implementation

During the development of the artefact, it is imperative to create synergies between the structure of the practice assignments and the feedback system as a prerequisite to the design-build-evaluate cycle (Hevner et al., 2004) required by the DSR approach. To achieve this synergy, one member of the research team developed practice assignments by first creating an inventory of digital skills required to complete realistic tasks (e.g., import data from external sources) in Microsoft Word or Excel. The researcher also mapped the pedagogical sequencing of these digital skills in order to identify what tasks would be prerequisites to others. A second member of the team accrued substantial expertise with the commercial products available on the market. He taught classes using two of the major commercial course solutions: SIMnet by McGraw Hill Education and MindTap by Cengage. The two researchers then iterated multiple times in the creation of practice assignments that substantiated the pedagogical requirements. Assignments were tested by the researchers and then by teaching assistants who had previously taken the class.

During the design of the practice assignments, the two researchers met weekly with the members of the team in charge of the technology architecture for delivering content and the design of the software evaluating the practice assignments – the grading engine. During these meetings, test cases were evaluated to ensure that the intended pedagogical objectives could be achieved. The team overcame limitations and errors by either modifying the evaluation code or the practice assignments. For example, one of the pedagogical objectives in Excel is the use of conditional formatting. However, it is difficult for the grading engine to disambiguate multiple conditional formatting rules on the same range of cells. Instead of trying to improve the code to overcome this limitation, the team changed the task description to ensure that any range of cells would

require at most the application of one conditional formatting rule. The pedagogical objective was achieved while maintaining feedback scalability through automated evaluation of learners' work.

The technology core of the ST artefact is composed of two major elements: the learner application and the grading engine. We introduce them below, and, following Meth, Mueller, & Maedche, (2015), we identify the specific design features (DF) of the artefact (see Table 1 for a summary). The meta-requirements and design principles introduced earlier are generic and applicable to any instantiation of the artefact. Conversely, design features are "specific ways to implement a design principle in an actual artifact" (Meth et al., 2015, p. 807).

[Table 1 near here]

Learner Application

The learner application was custom-developed on a backend using the MEAN free and open-source JavaScript stack (MongoDB, Express.js, AngularJS, and Node.js), running on a Linux OS instance from Amazon Web Services (AWS). Cloud hosting frees developers from server maintenance and security responsibilities, while at the same time providing computing and cost scalability proportional to the number of users and calls to the host server. It also affords traceability, thus enabling the efficient logging of users' activity (DF2).

The application adopts a responsive framework, thus being accessible and usable on a personal computer, tablet or smartphone. As described earlier, a custom solution was necessary because successfully implementing the tenets of intervention theory requires the design of an ST artefact that simultaneously balances pedagogical priorities, the functionalities of the grading engine, the structure of the assignments and the structure of data files. Designing and developing the performance feedback system

from the ground up provided full control over the entire process: from digital skills identification to framing of the exercises, to the evaluation approach and the degrees of freedom exercised in the assessment of students' work. Controlling the structure, strengths, and limitations of the grading engine enabled the design of practice assignments that are simultaneously realistic (DF12), pedagogically sound (DF7) and evaluable at scale (DF10). Furthermore, the application supports both Windows and Mac versions of Microsoft Office and does not rely on a simulated environment, thus supporting the realism design principle.

For the application front-end, we adopted the Bootstrap library and the Embedded JavaScript (EJS) template language, ensuring native mobile and desktop responsiveness while guaranteeing compatibility on all platforms. The application was entirely developed with HTML, JavaScript, and Cascading Style Sheets (CSS). The code is version controlled using Git, and it is hosted on GitHub to facilitate collaborative programming.

The ST artefact design implemented MR1 by requiring students to sign into the course application to access course material (DF1): practice assignments, theoretical topics, test schedules, and performance results. The application tracks and monitors traffic using Google Analytics (GA). GA enables the collection of students' behaviour data at the session, page, and click event level, while completion data for assignments is tracked via file timestamping (DF2). Once students are ready to complete an assignment, they work in a copy of the application (e.g., Microsoft Word) installed on their own personal machine (DF12). Upon completion, they save and submit the file through the learner application. The grading engine then evaluates the file and returns feedback to the learners within the course application under the performance results tab. Results can be accessed using any device (Figure 2) and are presented as an itemized

list of all the tasks in the assignment and display the maximum point value of each task (DF8). Next to each item, the report shows how many points the student received. Values range from zero to the maximum points for the item, with intermediate values representing partial credit for partial completion of the task (Figure 3).

[Figure 2 near here]

[Figure 3 near here]

Grading Engine

The grading engine was custom developed in Python 3.7.0. Leveraging the fact that Microsoft Office documents are collections of compressed XML files, the automatic grading software takes a "key file" created by the assignment designer as the basis for task evaluation. The key file is a Word document or Excel spreadsheet that contains all the tasks in the exercise correctly completed. Once unzipped, the application parses the key using the *xml.dom.minidom* package and extracts the content of relevant cells with the *openpyxl* package. Student files follow the same processing pipeline upon submission.

Once all features of the key and learner documents are extracted, the grading engine calls a sequence of generic functions that have been mapped to the relevant tasks and subtasks in the assignment (see Figure 4 for sample code and the online repository for all of the code). Using the *numpy* package to perform array calculations, the software matches learners' work to the key. Note that the matching is not done at the level of outputs (e.g., computation results), but at the level of the XML code representing the students' Excel sheets (e.g., workbook and worksheet level commands, formatting selections, syntax of formulas contained in specific cells). The grading engine can be configured to seek exact matches, when relevant, but it is robust to acceptable alternative solutions through parameters and helper functions. Moreover,

when appropriate, the grading engine loads multiple key files that contain equifinal solutions for the task (DF3 and DF4). Note that in keeping with DP3.3, the grading engine is intended to be a component of the ST artefact rather than the sole element of the feedback system. Thus, the software does not attempt to provide definite answers or comprehensive pointers to self-help resources the way an online learning system would. Instead the IT core provides the first layer of feedback that allows students to verify their own progress.

An important element of the ST artefact is the "open lab," where the instructor and teaching assistants are available to review student progress and help them master specific skills. The grading engine is an integral part of this learning process because it creates substantial efficiency by empowering learners to independently verify their progress. Armed with feedback from the online application students can review learning resources in an attempt to overcome their difficulties (DF9). They can also visit the open lab where interactions with experts focus on specific problems flagged by the grading engine (DF9). This design, combining both software and human resources (Figure 2), is central to the ST artefact scalability objective by integrating IT-enabled automation with the human interactions instrumental to quality education (Chambliss & Takacs, 2014; Silver & Markus, 2013).

[Figure 4 near here]

Evaluation: Iterative deployment and assessment

The DSRM process model (Peffers et al., 2007) applied in this research calls for multiple iterations of the design-build-evaluate cycle (March & Smith, 1995). Following this approach, we present the evaluation results of each of the three iterations we performed.

First Iteration

The result of the first design iteration, a pilot study, was an ST artefact supporting the delivery of six Excel and three Word practice assignments. The assignments averaged 37 and 33 tasks for Word and Excel, respectively (range: 21 to 52), with a number of tasks encompassing multiple sub-tasks. Once a week, using a flipped-classroom pedagogy, the learners met in class and worked on a new practice assignment. Each assignment followed an ongoing scenario that spanned the duration of the semester and contained links to the official Microsoft Office documentation (descriptive and/or video sources) for the skills it required. The scenario gave continuity and realism to the work (DF12). Therefore, learners could see how their acquisition of increasingly advanced skills would translate into their ability to carry out progressively more complex and realistic work (DF7).

During this first iteration, students had to submit their file via email before the next class (one week). Those who did so received an email with a detailed task-by-task report before the next meeting. The report listed, for each required task, the importance of the task (i.e., assigned points value) and the actual points earned by the students (i.e., percentage of successful completion). Keeping with the tenets of intervention theory, the design of the ST artefact fostered free and informed choice. Thus, while feedback was provided to those who submitted the work by the deadline, the evaluation was not used in the computation of the final grade (DF5). Only formal exams, based on the skills learned in the assignments, contributed to the learners' final evaluation (DF6).

The research team held weekly meetings to review the practice assignments and to discuss the limitations and refinements of the grading engine during the pilot of the feedback system in use (semester 2). These analyses revealed an important trade-off between realism, as originally designed in the practice assignments, and the reliability

of performance feedback for practical digital skills mastery. For example, in a realistic analysis in Excel, a student needed to import or create data, clean or structure the data, then analyse it or process it, and finally present the results. However, if all these tasks were structured within the same file, there was a high probability of compounding errors throughout the document, particularly with novice users. Consider the following examples: a) when reading external data into Excel (e.g., from a website) it is possible that numeric data are formatted as text. If a learner fails to appropriately reformat the data, any computation on those data will not work or produce misleading results. b) When converting data with a "nested IF" statement, a misspelling may lead to incorrect intermediate results. Any further processing using the wrong intermediate results could provide misleading results or may not allow the learner to practice the ensuing tasks. During the pilot, the team identified classes of skills and task designs that engendered these limitations. More generally, the evaluation demonstrates that task realism, an important design principle for practical digital skills learning (DP 5.1), must be implemented through modular exercises that are evaluated as self-contained units.

With respect to learners' behaviour, class attendance was consistently above 75%, and the formal course evaluation showed the course was well received 3.53/4 (college total 3.35/4). Learners were present and actively engaged during the in-class sessions. However, they did not consistently complete the assignments. On time completion of each assignment was the only requirement to receive feedback on that specific homework. Completion rate, measured for each assignment as the percentage of students who submitted at least once before the deadline, steadily declined during the semester (Table 2).

[Table 2 near here]

These findings indicate that the first design failed to ensure that students stayed on track with the progression of the course. In an effort to develop realistic assignments, the ST artefact made it difficult for learners to complete the assignment in one session. The inability to conclude the work seemed to add to the challenges the students encountered in mastering the material (e.g., cognitive load), as 22 of the 27 respondents to an end-of-semester survey agreed with the statement: "shorter practice assignments would be more effective." Thus, in keeping with design science fundamentals (Hevner et al., 2004; Sein et al., 2011; Walls et al., 1992) and recent guidance on the DSR process (Gregor & Hevner, 2013; Sein et al., 2011), we iterated the ST artefact design and its evaluation.

Second Iteration

Our analysis of the first iteration results indicated that the overwhelming nature of the assignments and low speed of the feedback system (weekly cycles) were responsible for the observed low completion rates. The search for a solution spurred a redesign of the assignments and the evaluation processes by 1) chunking assignments (DF13), 2) adding structured starter files (DF14), and 3) introducing fast-cycle feedback without deadlines.

Assignment Chunking

To implement the progress principle and make assignments more manageable (DF13) while maintaining a high degree of realism (DF12), we split them into shorter exercises called "chunks" (Figure 5). This reformatting produced an average of 4.33 pedagogically consistent chunks (range 3-5) for each of the assignments. The chunks each averaged 9.54 tasks (range 4–17) compared to 35.66 tasks for the assignments in the first implementation (range: 21-52). Chunking of assignment made the individual

assignments less daunting when learners began working on them (Figure 5). Moreover, students now had intermediate completion points for the work, and they could submit chunks independently when they needed to stop. When resuming work, they could begin with a new starter file at the beginning of a fresh assignment component. Finally, chunking improves learner control, allowing students to decide what tasks they could skip or what skills they needed to practice multiple times. For example, a student who is well versed in Excel may skip the first chunk of an assignment if she is very familiar with the skills in it. Conversely, a student who has difficulties with a particular set of skills may resubmit the relative chunk as many times as needed.

[Figure 5 near here]

Structured Starter Files

Each assignment chunk required that students download the structured starter file to initiate work. Starter files provided both the data for the learners to use and serve as containers for their work. For each chunk subsequent to the first one in an exercise, the structured starter files are a replica of the key of the previous chunk. In other words, the starter file for chunk *x* in exercise *y* is the complete and correct solution to the previous chunk in the same exercise (i.e., *y.x-1*). The objective of using target files in each chunk is to increase the precision of the assignments (DF14), to improve the reliability (DF3) and validity (DF4) of the feedback systems, and to enable learners to more easily interpret the feedback from the grading engine (DF11). Appropriately constructed starter files reduce the problem of compounding errors as well. Consider the earlier example of a nested IF where a misspelling in the formula may lead to incorrect intermediate results. Segregating sequential tasks in different chunks eliminates carryover mistakes. Moreover, structuring the starter files with document protection and visual cues helps to prevent minor mistakes that do not warrant negative feedback. For

example, the simplest way for the Excel grading engine to identify worksheets in an Excel workbook is either by position or by name. If a student renames or moves a worksheet when not instructed to do so, she may prevent accurate feedback on all tasks contained in that sheet. Requiring the download of a structured starter file with locked worksheets eliminates this type of false negative error and strengthens the validity of the feedback system.

Fast-cycle Feedback

The redesigned ST artefact simplifies and speeds up the process of submission, evaluation and performance feedback (DF10) using new technology in the learner application and a new workflow for running and releasing evaluations. Specifically, a drop-file feature was introduced for each of the chunks in the nine assignments (Figure 5). Rather than emailing their assignments, as they did during the first iteration, learners uploaded their file immediately upon completion directly in the application. The structure of the performance report remained unchanged: for each task and sub-task, the assigned points value and the actual points earned by the learners were listed. However, in the redesign, the evaluation was not emailed as a pdf report. Instead it was accessible privately by each student directly in the learner application by displaying the results produced by the grading engine in a section titled “Practice Results.” These new features enabled fast-cycle feedback without deadlines to increase feedback timeliness (DF10) while better accommodating student schedules (DF13). Any file dropped by the student was evaluated within 24 hours, and results became available in the practice results section of the app regardless of sequencing of completion. This approach also allowed students to drop multiple versions of the same exercise chunk, thus enabling repetition and providing feedback on each attempt.

As in the first iteration, the ST artefact design implemented MR2 and MR3 by providing nine optional practice assignments (DF7) and two required exams (DF6) over the course of the semester. Work voluntarily submitted by the students was evaluated, and performance feedback was returned to the learner (DF5).

A summative naturalistic evaluation (Venable et al., 2016) of the ST artefact was performed after the second iteration by addressing a) the scalability of the feedback system, b) the validity and reliability of the feedback system, c) the influence of the feedback system design on learner's behaviour and engagement. The chunking of assignments, the addition of structured starter files and fast-cycle feedback produced the expected results with respect to users' perception. Only 12.5% of respondents to the end-of-semester survey perceived the length of the assignments as inappropriate, as compared to 81.5% of users in the first implementation of the ST artefact.

Both exams required the learners to complete realistic tasks. The first one tested data analysis in Excel (27 tasks) to be completed during a 60 minutes exam session. The second one tested word processing skills in Word (14 tasks) to be completed during a 15-minute exam session. Three independent experts manually evaluated each task for each learner. The experts were very familiar with the content of the exam, knew how to complete the practical skills being tested, and had deep knowledge of common errors students make when completing the tasks in the exams. Moreover, they logged the time of completion required to evaluate each document. Including all subtasks, each file comprised 50 and 25 items for each student in the Excel and Word exams, respectively. On each item, the raters as well as the grading engine could provide full or partial credit. Each rater provided a total of 1,300 evaluations for Excel and 550 for Word. The raters were experts in the use of the target applications (i.e., Word and Excel), took the exam as practice prior to grading and had full access to the exams' keys (i.e., the

solution files that included all the possible acceptable ways to complete each task). They worked independently, grading at a pace they would normally adopt as teaching assistants. This approach simulated a realistic manual evaluation session for college-level digital skills. In case of any doubts, for example about alternative solutions to a task, the raters sought guidance from the member of the research team in charge of the assignments and exams content. In these cases, the raters made a note explaining the nature of the doubt and their rationale for point allocation.

Scalability

Despite not testing with a large number of students per class, the evaluation suggests that the ST artefact did achieve the scalability objective. The three raters took an average of 10:63 minutes (637.8 seconds) per Excel exam (range 7:23–17:04 per rater) and 6.68 minutes (400.8 seconds) per Word exam (range 3:23–12:23 per rater). The speed of the grading engine varied with the characteristics of the machine used. Using a Dell XPS 15 9650 series personal computer with 16 GB of memory and Intel Core i7-7700 CPU (2.80 GHz), the average evaluation time was 3.61 seconds per Excel file (range 2.59-4.82) and 0.45 seconds per Word file (range 0.17-1.37). While both the manual and automated systems scaled linearly with the number of documents to evaluate and the number of tasks in each exam, the grading speed differential is more than two orders of magnitude. It follows that only the feedback system enabled by the grading engine can feasibly scale to support hundreds or thousands of learners.

Validity and Reliability

We employed a diachronic approach to validity analysis (Baskerville et al., 2017) by investigating the ST artefact's performance with different users on a wide range of realistic tasks (i.e., those tasks included in the official examination of learners' skills

mastery assessment). Three teaching assistants carried out a realistic evaluation under time constraint designed to evaluate scalability as described above. At the end of this first evaluation, which was completed "at speed to simulate a typical manual grading session, the rater with the highest level of mastery and grading accuracy reanalysed each file to corroborate the assessment provided by the raters at speed. This step aimed at creating the "ground truth," an exact assessment of each task in each file, to ensure the evaluations provided by the three raters were both accurate and consistent across all tasks and learners. Thus, the rater was instructed to take as much time as needed to compute the correct evaluation of each task and sub-task and to focus solely on precision. Moreover, whenever she had any doubt with respect to the correct evaluation of a task or found any inconsistency among the evaluations of the original raters, she would submit the work to the research team member in charge of the evaluation. The ground truth so established was used to evaluate the precision of the grading engine and the human raters. Overall, this evaluation indicates that the grading engine was superior to each human rater, both in terms of the total number of errors and the score value of those errors (Table 3). The grading engine was more accurate in evaluating learners' digital skills when compared to human raters engaged in a typical manual grading session.

[Table 3 near here]

The validity of an ST artefact designed to provide accurate performance feedback can be claimed if the artefact correctly identifies instances of mastery of specific practical digital skills, while avoiding both false positives and false negatives. False negatives occur when learners receive negative feedback, even though they have mastered the intended skill. False positives occur when learners receive positive feedback but have not mastered the intended skill.

False negative errors stem from the inflexibility of rule-based systems. A simple case is a misspelling. In some instances, misspellings are consequential and indicative of lack of skills mastery. For example, misspelling a formula name or misspelling text to be identified in a conditional formatting rule or as the argument of a function will produce an error. In other cases, misspellings are inconsequential and do not indicate a lack of skills (e.g., misspelling a label or a title in a chart). In the case in which misspellings are inconsequential, we introduce a tolerance level using regular expressions. More generally, eliminating false negatives calls for a task design that is robust to different solutions and resilient against distraction mistakes (e.g., a student typing the correct formula in the wrong cell location).

False positives errors occur when the combination of task design and evaluation code functionality is such that a specific mistake cannot be detected. In these cases, it is difficult to check enough elements of the learner's work to be precise enough to reliably identify the error. There is an inherent trade-off between false positive and false negative feedback errors. Interestingly, this trade-off reinforces why the grading engine is not enough to accomplish the goal of providing accurate and timely performance feedback at scale. An ST artefact designed to perform the evaluation must integrate pedagogical considerations (i.e., what skills learners should master), assignment design (i.e., what tasks can best test whether mastery has occurred) and measurement (i.e., code functionalities to detect accurate task completion in the learner's file). Depending on the interaction of pedagogical priority, assignment design, and evaluation code functionality, the same learner action may be evaluated as correct or generate false positive/negative errors. Consider as an example how a simple sum of two cells in Excel can be written as an infinite number of equifinal strings: 1) = A1 + A2; 2) = (A1 + A2); 3) = A1 + A2; 4) = SUM(A1:A2). While all four cases produce the same numeric

result, deciding whether the learner has mastered the skill of using addition in Excel depends on the emphasis placed on competing pedagogical objectives. In this case, if the expectation is that the student is efficient in computing the result, then only solutions (1) and (4) should trigger positive feedback. If the expectation is that the learner is able to compute the correct score, then all four approaches appear to be equally good. However, solution (3) is not only inefficient but also conceptually incorrect, since it uses absolute referencing when it is not required. It is also potentially detrimental to future tasks if, for example, the formula is to be replicated over a range of cells. Endowing the ST artefact with the ability to mix exact matching and flexibility to eliminate false positive and false negative errors in feedback delivery is the nature of the design challenge we are addressing. Note that this trade-off exists in manual performance evaluation systems as well. The human evaluator needs to decide when deviations from the expected result (e.g., the key) are inconsequential and should trigger positive feedback or when they are indicative of lack of mastery, calling for negative feedback.

We claim validity for the current instantiation of our ST artefact with false positives occurring in slightly more than 1% and 3.5% of cases, respectively in Excel and Word, and false negatives not reaching the 1% in Excel and the 3.5% mark in Word (Table 4). In other words, while maintaining the realism of the overall task, the combination of skills selection, tasks design, structured starter files, and code functionality bundled in our ST artefact performed satisfactorily.

[Table 4 near here]

With respect to reliability, evaluation of the pilot results showed that one common source of unreliability is the evaluation of tasks that depend on the correct completion of a previous one. The redesign effort focused on identifying and eliminating these occurrences through assignment chunking and careful design of

structured starter files. As a consequence, the second instantiation of the performance feedback system generated only 6 inconsistencies, 3 out of 1,300 evaluated Excel tasks and 3 out of 550 evaluated Word tasks.

Completion Rate

It is imperative that learners use the feedback system if they are to receive feedback on their level of digital skills mastery. To overcome the low completion rate of practice exercises by the learner population during the first iteration, the ST artefact design leverages innovations inspired by the progress principle – namely, chunking of assignments, the use of structured starter files, and the introduction of fast-cycle feedback without deadlines. Feedback percentage provides a measure of success on the objective of stimulating artefact utilization. We defined feedback percentage as the number of individuals who submitted their work for evaluation. We measured it by counting, for each assignment, the number of students who submitted at least one of the chunks. As compared to the results during the pilot (Table 2), feedback percentage was significantly higher during the second iteration (Table 5). The average completion rate per assignment was 53.78% (range 37.04% - 78.57%), and the differential over the nine assignments as compared to the first implementation was 27.54%. Thus, on average, one more learner out of every four enrolled received evaluations of their work when using the redesigned performance feedback system (range 9.13% - 48.40%). The impact of the redesigned ST artefact was stronger on those assignments that had the lowest completion rate during the pilot.

[Table 5 near here]

Given the high degree of reliability achieved by the artefact on existing practice assignments, it was possible to scale and further automate the workflow through a third design-build-evaluate iteration.

Third Iteration

During the third iteration of the design, no new requirements or design principles were introduced. The iteration focused on further improving the ST artefact's ability to produce fast-cycle feedback without deadlines (DF10) while further improving scalability in support of MR3. The grading engine was rearchitected. Grounded in the cloud-first paradigm, the code was refactored as a series of Lambda functions[2] hosted on the same AWS infrastructure housing the code for the responsive course application. This change did not impact the process followed by users to submit their work for evaluation, but it eliminated all manual handling of the submitted files. Once a file was submitted, the AWS storage service S3 triggered the execution of the appropriate AWS Lambda function in the grading engine. The submitted file was evaluated in near-real time. Upon completion of the evaluation, the feedback file was stored in a JSON format in the MongoDB instance powering the app, thus becoming immediately visible to the learners through the practice results page. Students no longer needed to wait 24 hours to receive feedback.

The improved ST artefact was implemented with a cohort of 310 users in four sections of the course to enable a summative naturalistic evaluation episode at scale (Venable et al., 2016). The evaluation approach followed that of the second iteration in both methods and focus (i.e., scalability, validity and reliability, and learner's engagement), but replicated the results at scale. Given the large number of files to be evaluated, five raters were recruited. Three raters who had also performed the

[2] AWS Lambda is a cloud-based service that enables event-drive code execution. The Lambda service is based on a serverless architecture that automatically manages and provisions the computing resources required to run the code, thus enabling cloud-native application development and deployment.

evaluation during the second iteration and two new teaching assistants familiar with the content of the course and of the examinations. As in the second iteration, the raters and the grading engine could assign full or partial credit to each task, thus simulating a realistic manual digital skills evaluation session. However, due to the large number of tasks being evaluated – 5,727 in Word and 8,608 in Excel, the files were divided evenly between the raters. In case of any doubts, the raters sought guidance from the member of the research team in charge of the assignments and exams content.

Scalability

The evaluation corroborates the results from the second iteration regarding the scalability of the solution. Using a Dell XPS 15 9570 series personal computer with 16 GB of memory and Intel Core i7-8750H CPU (2.20 GHz), the average evaluation time was 3.63 seconds per Word file (range 2.45-5.73) and 1.97 seconds per Excel file (range 1.08-13.43). The main driver of grading speed is the file opening stage, which grows as a function of the size of the data file used in the exam. This phase, however, is necessary also for manual grading. While the speed of the grading engine is measured in seconds, the speed of raters is measured in minutes (average 5:25, range 1:05 – 13:58). The scalability of the solution was most evident during this iteration, when considering that the serverless AWS Lambda architecture enables near-real-time feedback to hundreds of simultaneous users. This is a level of scalability simply impossible for a manual evaluation system to attain.

Validity and Reliability

The creation of the "ground truth" followed the same procedure employed during the second iteration. At the end of the first evaluation step completed "at speed" by all teaching assistants, the same rater who created the ground truth for the second iteration

reanalysed each file. Again, the focus of this stage was the accuracy and consistency of the evaluation of each task in each file by taking as much time as needed to compute the correct evaluation. Whenever she had any doubt, she would discuss it with the research team member in charge of the evaluation. The ground truth was used to identify all false positive or false negative errors of the performance feedback system.

The results corroborate those of the second iteration when the grading engine performance is compared to the ground truth (i.e., the correct assessment of each file). Correct rates measured 94.88% over 5,727 tasks evaluated in Word and 96.88% over 8,608 tasks evaluated in Excel (Table 6).

Validity results for the third instantiation of the ST artefact confirm the quality of the solution. Specifically, false positives occurred in 0.69% (Excel) and 3.20% (Word) of cases, and false negative occurred in 2.44% (Excel) and 1.92% (Word), respectively. In other words, when scaling the number of users and the tasks evaluated, the ST artefact performance did not degrade and, in fact, slightly improved. The improvement is ascribable to the introduction of multiple keys designed to identify and evaluate "edge cases." Such cases occur when a student implements an unexpected solution for a specific task that, while non-standard, is nevertheless correct. Given the evaluation speed of the grading engine, it is possible to check each document against a set of keys, not just one solution. Thus, a small decrease in evaluation speed leads to a further validity improvement.

A complete reliability analysis, similar to the one carried out for the second iteration, was not warranted due to the minimal incidence of reliability errors in the previous iteration and due to the large number of tasks to be evaluated. Thus, reliability analysis was carried out on a subset of the exams. The five tasks that had the greatest evaluation discrepancy between submitted files were manually inspected (a total of

1345 tasks for the Excel exam and 1245 for the Word exam). This analysis showed that, after including the multiple keys design, grading inconsistencies on these five tasks had gone down to zero.

[Table 6 near here]

Completion Rate

As compared to the results during the pilot (Table 2), the feedback percentage was significantly higher during the third iteration as well (Table 7). The average completion rate per assignment was 58.62% (range 29.60% - 96.80%), and the differential over the nine assignments as compared to the first implementation was 32.39%. Thus, on average, one more learner out of every three participating in the course received evaluations of their work when using the redesigned performance feedback system (range 22.70% - 49.46%). Despite the almost ten-fold increase in the number of users, from 34 to 310 learners, the overall completion rate improved over the second iteration by an average of 4.85% per assignment (range 7.50% - 25.36%).

Results for average chunk completion improved for the average assignment over the second iteration as well, from 63.68% to 75.12%. They also became more consistent, with the standard deviation dropping from 21.43% to 4.69%. Taken together, these results suggest that increasing the cycle speed of the feedback engendered the intended positive effect on completion rates while providing no grading incentive for completing the assignments (DP2.1).

[Table 7 near here]

One could argue that feedback percentage is a misleading metric because assignment submission in the first design of the ST artefact should equate to submission of all the chunks in the redesigned artefact. While creating a precise equivalence is not possible, it is important to note that the submission of an assignment during the pilot

implementation did not guarantee the assignment was completed in its entirety. In fact, we recorded many instances in which learners would submit incomplete assignments because they preferred to receive partial feedback than forgo the opportunity altogether. Moreover, our data show that for all but two assignments, the percentage of chunks completed exceeded 50% during the second iteration (Table 5) and grew to more than two thirds (67.50%) of all assignments during the third iteration (Table 7). In other words, there is evidence that by chunking long assignments, the ST artefact stimulates learners to increase their engagement with the performance feedback system and that speeding up the feedback system to provide near real-time results generates a further improvement in learner engagement. In the following section, we discuss these results along with avenues for future artefact improvement and future research.

Discussion

In this work, we claim an improvement knowledge contribution focused on "developing new solutions for known problems" (Gregor & Hevner, 2013, p. 345). We argue that the results of our evaluation provide "proof-by-demonstration" (Nunamaker et al., 1990, p. 98) that the socio-technical artefact we designed achieves the three objectives of a) scalability to a large number of learners, b) validity and reliability of the feedback provided, and c) positive impact on learner's behaviour and engagement. The artefact was evaluated over three iterations, culminating with a summative naturalistic evaluation (Venable et al., 2016) at scale (310 users).

Scalability of the ST Artefact

All the components of the ST artefact (i.e., online content app, grading engine, open labs) performed reliably and as expected. Completion and behavioural digital data streams (Pigni, Piccoli, & Watson, 2016) are seamlessly tracked by the application.

Learners logged into the app for access to learning resources. This result confirms the outcome of the pilot implementation and suggests that resource utilization can be reliably tracked by the ST artefact. The results of the evaluation corroborated the value of automating the current workflows. The drop-file feature in the ST artefact allows students to submit their work and receive immediate feedback at any time from any location using the app hosted on the AWS infrastructure. As shown in the evaluation, the grading engine achieved a degree of accuracy on existing practice assignments that allowed confident automation of the evaluation process. This fully automated workflow is in line with the objectives of providing near-real-time feedback while scaling to a large number of learners.

While our work demonstrates the scalability of the ST artefact, future research should investigate the limits of such scalability. The current version of the IT core can theoretically scale infinitely, but there may be a linear relationship between the number of learners and support staff in the open labs. While the current design creates substantial efficiencies in the provision of performance feedback, the scalability of the social elements of the artefact awaits formal evaluation. Learners can attempt assignments multiple times and receive a fresh evaluation in near real-time for every try. Since assignments are not used for grading purposes, the learners also have access to the practice solutions (i.e., the key files) and can, therefore, work independently or with peers to understand and correct their mistakes. Students reach out to staff during open labs only when they have been unable to independently resolve their doubts. Moreover, when they do so, they have very specific questions that can be resolved efficiently. While scalability is designed in the performance evaluation system through the synergy between the technical and social elements of the ST artefact, formal evaluation of scalability limits is an open question. Future research could directly

investigate this question by tracking the utilization rate of physical resources (e.g., staff and classrooms where open labs are held) as a function of the number of learners engaged in the course. Note that in keeping with DP3.3, the objective of this research would not be to eliminate physical social interactions, as in an online course, but to improve the efficiency of feedback provision during open labs.

Validity and reliability of the ST Artefact

With respect to validity and reliability, the ST artefact achieved satisfactory levels. An analysis of results against true performance scores for each evaluated task by each learner demonstrates the superiority of the grading engine versus each individual human rater. The main source of advantage for the grading engine is its access to the XML representations of the learners' work, coupled with its processing speed. In the interest of efficient completion of the evaluations, human graders will typically check multiple tasks that evaluate the same skill by sampling – particularly if the task is repeated many times. A good example is offered by a task that required learners to perform a data cleaning step by removing hyperlinks from text downloaded from the Web and pasted in Excel. The task repeats for 25 links, each in a separate cell. Learners who mastered the data cleaning skill and understood the goal of efficiency taught in the course would select the full cell range and remove all the hyperlinks at once. Students who may not know how to remove hyperlinks may simply change the appearance of the text. Others who may understand how to remove the hyperlinks correctly but who have yet to internalize the efficiency principles introduced in the course will repeat the task over the 25 cells individually. In both cases, the student should be alerted by the feedback system that their work is not correct. However, for this kind of task, manual graders will typically review thoroughly only a small sample of cells and visually inspect the whole range. By doing so they can identify learners who are unable to remove hyperlinks on

individual cells, but it is unlikely they will identify users who perform the task inefficiently. The difficulty of evaluating all cells becomes even greater as the task becomes more realistic (e.g., requiring data cleaning over 250 or 2500 cells). Conversely, since the grading engine has direct access to the XML representation of the Excel worksheet and can quickly parse each cell in the range, it will detect a mistake even if it occurred in only one out of 25 cells (an actual evaluation case in our research). A similar example, in the context of word processing skills, is the inability of human graders to evaluate whether a student used efficient strategies (e.g., table autofit to create professional looking tables) or how precisely they performed a task (e.g., cropping a picture). For skills of this type, the grading engine is not only faster but also more accurate in the evaluation. As a consequence, its use enables superior precision in performance feedback on realistic tasks when compared to a manual feedback system.

The ST artefact evaluation also demonstrates that the number of inconsistencies produced by the grading engine, false positive and false negative errors, are objectively small (Table 6). While the result is very promising, it is important to note that we have no way to validate tasks until they are implemented in assignments. In other words, because the grading engine is built as a general rules engine to evaluate XML files, it is difficult to provide a priori generalized validation. An important avenue for future research is the study of the optimal blend of strict assessment of task performance (i.e., exact matching) versus flexible evaluation. Since assignments and homework are a service to learners and have no bearing on the individual's learning assessment (DP2.1) during the completion of practice assignments, exact evaluation is optimal. Faced with an error, the learner will need to re-evaluate the task and think about why it was marked as incorrect. Failing to resolve the issue independently, the student would know that they need to visit the open lab and discuss the problem with an expert (Figure 2). It is in

this venue that the staff can reinforce concepts such as efficient completion, rather than mere completion, of tasks. However, in an exam setting in which the assessment is used for formal grading (DP2.2), this level of precision could be overly penalizing. In other words, the configuration of the grading engine may need to be different for practice assignments (stricter) and exams (more flexible).

In our research, we have already experimented with this approach, by creating multiple keys to feed to the grading engine. A simple approach to the creation of multiple keys is to systematically extract all solutions learners have implemented from the set of submitted files. Upon filtering the unique set of solutions, the correct ones can be evaluated and coded into the keys. This is a time consuming and reactive process that can only be initiated after all files are collected – as in the context of an exam. Future research should extend the concept of multiple keys to practice assignments. One approach is to apply the extraction and filtering process described earlier to the files submitted. As the number of files grows, we can expect the chances of new correct solutions to existing tasks to drop asymptotically to zero. An alternative is to develop a method for creating multiple keys a priori. Given the characteristics of a task being evaluated by the grading engine, the method would enable complete identification of all the correct possible solutions. We believe that such method, developed as a design science research project, would be a significant knowledge contribution in its own right.

Pedagogical Implications

The above discussion is central to the challenge of designing an ST artefact that follows the tenets of intervention theory. In an effort to create a feedback system that can fulfil the realism principle (DP 5.1), the design team faces a constant trade-off between precision and flexibility that requires the direction of well-defined pedagogical objectives. While formative assessments (i.e., practice assignments) that serve the sole

purpose of providing feedback (DP 2.1) can mechanically evaluate each task, evaluative assessments (i.e., exams) should provide an accurate evaluation of the learner's abilities (DP 2.2). Considering the example of repeated subsequent errors in which a learner might make the same mistake multiple times in the same file: how should the system handle such a situation during evaluative assessments? There are two possible approaches. One is to create a configuration file that would specify a fixed "decay rate" for repetition of the same error on multiple tasks. Such a decay rate would mechanically capture the effect of penalizing a learner less and less for repetitive mistakes. An alternative solution, the one adopted in our design, leverages the interplay of pedagogical considerations, assignment design, and measurement. Thus, it incorporates the limitations of the grading engine, such as its natural inflexibility into the design of the exams. With this approach, repetitive tasks are treated as one task and weighted accordingly in the final score. For example, in one examination the students have to repeat a SUM, AVERAGE, MIN and MAX computation over eight columns. The 32 formulas are treated as one task with partial credit associated with each formula. As such, the task is implicitly weighted. More subtly, and more importantly, a design that follows the realism principle seeks to develop "digital meta-skills" that go beyond the mere use of the application. For example, learners should use Excel/Word effectively (using the right functionality for each task) but also efficiently (moving quickly through the file with keyboard shortcuts, never repeating a task that can be systematized, envisioning general solutions before applying them to a repetitive task). Thus, if a student only misses one or two of the 32 formulas in the task, they will be marginally penalized. But the penalty will not be associated with a typo or small distraction mistake. It would instead be indicative of the fact that the student did not prepare the formula once, with the appropriate configuration (e.g., mixed cell referencing), and then

efficiently dragged it across all the repetitive tasks. While necessary for accurate evaluation during the examination, this design is also very useful in formative assessment. In our experience, the majority of the learners do not intuitively grasp the importance of digital meta-skills. However, when they visit open labs to discuss the perceived unfairness of being deducted points for "minor mistakes," the instructor and the teaching assistants have an opportunity to reinforce the importance of digital meta-skills (Figure 2). In a well-crafted assignment, there are many opportunities to build into the exercise key pedagogical objectives rather than providing feedback mechanically. We are not aware of any integrative framework that conceptualized and organized the classes of digital skills. Such a framework would be a valuable future addition to the literature, and it would simplify the systematic design of content for a feedback system like the one we developed.

Learners' Engagement

With respect to learner engagement, we claim considerable improvement over the pilot implementation. Our data do not allow us to conclusively disambiguate the effect of chunking versus fast cycle feedback and this could be an interesting avenue for future research in the behavioural science paradigm (Hevner et al., 2004). From a design science research standpoint, the evaluation of our redesigned ST artefact suggests that we are making progress in "changing existing situations into preferred ones" (Simon, 1996, p. 111). Yet, despite claiming an improvement over the previous approach, we argue that completion rates at the individual chunk level are still low (Table 7). This observation calls for future research to identify interventions that can foster internal commitment and stimulate learners to complete practice assignments despite the free and informed choice constraints posed by intervention theory (e.g., MR2). A promising research avenue consists of the implementation of digital nudging principles

(Weinmann, Schneider, & Brocke, 2016). Digital nudges attempt to influence the affected parties with the objective of making the choosers better off while still allowing freedom of choice (Sunstein & Thaler, 2003; Weinmann et al., 2016). For example, digital nudging principles can guide the development of ST artefacts, such as a text-based chatbot aimed at reducing learners' tendency to engage in academic procrastination on weekly assignments (Rodriguez, Piccoli, & Bartosiak, 2019). With respect to learners' engagement and utilization of the performance feedback system, future research needs to also investigate the question of whether the ST artefact leads to improved mastery of digital skills. Such evaluation would be more appropriate in the behavioral rather than design science tradition (Hevner et al., 2004), requiring the set-up of experimental and control conditions that enable systematic comparison of learners aided by the performance feedback system with counterparts in a traditional environment.

Digital Skills Mastery at Scale

It is estimated that global demand for data scientists is currently exceeding supply by over 50% (Gershkoff, 2015). Furthermore, as the role of technology becomes increasingly important in our society, we can expect this misalignment to widen in the near future. With both the public and private sectors investing resources in reducing this gap, the design of efficient and effective feedback systems for digital skills mastery is an important area of research. We posit that the meta-requirements and design principles we introduced constitute a basis for action for the design and implementation of feedback systems for a broad range of digital skills (e.g., data visualization, machine learning). Specifically, our work spotlights the importance of conceptualizing the feedback systems as a ST artefact in which both the social and technical components assume equally important roles (Silver & Markus, 2013). On the one hand, the grading

engine and course application enable the delivery of the feedback system in a scalable manner under resource constraints. On the other hand, pedagogical objectives and the open lab are crucial for achieving the learning objectives while fostering learners' motivation and control. Thus, our work suggests important practical implications for the allocation of resources and implementation of policies. Educational organizations need to carefully balance the trade-off between the two components of the ST artefact to guarantee the achievement of their learning objectives. For example, the implementation of technologically advanced off-the-shelf course solutions does not guarantee the achievement of clear-cut pedagogical objectives and learners' motivation (Piccoli et al., 2017). Similarly, policies directed to the improvement of digital skills at scale need to provide the necessary resources – technical, organizational and financial – to support the implementation of an ST artefact like the one we propose.

Conclusions

The global demand for formal learning has resulted in increasing pressure to "massify" education. As digital skills development becomes an increasingly foundational skill for the labour force and society at large, information systems researchers should take the lead in devising learning systems that can scale to fit demand without sacrificing quality. Our research shows how the synergistic contribution of pedagogical prioritization (i.e., what skills to cover), assignment design (i.e., what tasks to use to evaluate mastery), and automated measurement (i.e., grading engine functionalities for error detection) underpins the design of a scalable performance feedback system. We maintain that information systems scholars, with their research interest at the intersection of technology developments and organizational practice, are best positioned to help devise solutions to alleviate the gap in digital skills development. We hope that our work supports and stimulates further efforts in this area.

References

Amabile, T. M., & Kramer, S. J. (2011). The power of small wins. *Harvard Business Review*, *89*(5), 70–80.

Argyris, C. (1970). *Intervention Theory and Method: A Behavioral Science View*. Reading, MA: Addison-Wesley.

Bain, K. (2004). *What the Best College Teachers Do* (1 edition). Cambridge, MA: Harvard University Press.

Bandiera, O., Larcinese, V., & Rasul, I. (2010). Heterogeneous Class Size Effects: New Evidence from a Panel of University Students*. *The Economic Journal*, *120*(549), 1365–1398.

Baskerville, R., Kaul, M., & Storey, V. (2017). Establishing Reliability in Design Science Research. *ICIS 2017 Proceedings,* Seoul, South Korea.

Benton, S. L., & Pallet, W. H. (2013). Essay on importance of class size in higher education | Inside Higher Ed. Retrieved March 1, 2019, from https://www.insidehighered.com/views/2013/01/29/essay-importance-class-size-higher-education

Burning Glass Technologies. (2017). *The Digital Edge: Middle-Skill Workers and Careers*. Boston, MA: Burning Glass Technologies.

Chambliss, D. F., & Takacs, C. G. (2014). *How College Works* (1 edition). Cambridge, MA: Harvard University Press.

Cuseo, J. (2007). The Empirical Case against Large Class Size: Adverse Effects on the Teaching, Learning, and Retention of First-Year Students. *Journal of Faculty Development*, *21*(1), 5–21.

Dean, T., Lee-Post, A., & Hapke, H. (2017). Universal Design for Learning in Teaching Large Lecture Classes. *Journal of Marketing Education*, *39*(1), 5–16.

Dijk, J. van, & Deursen, A. van. (2009). Inequalities of Digital Skills and How to Overcome Them. In E. Ferro, Y. K. Dwivedi, J. R. Gil–Gracia, M. D. Williams (Eds.), *Handbook of Research on Overcoming Digital Divides: Constructing an Equitable and Competitive Information Society* (pp. 278–291). Hershey, PA: Information Science Reference.

Drucker, P. (2001, November 1). The next society. *The Economist*. Retrieved from https://www.economist.com/node/770819

Gershkoff, A. (2015, December 31). How To Stem The Global Shortage Of Data Scientists. Retrieved June 24, 2019, from TechCrunch website: http://social.techcrunch.com/2015/12/31/how-to-stem-the-global-shortage-of-data-scientists/

Gregor, S., & Hevner, A. (2013). Positioning and Presenting Design Science Research for Maximum Impact. *Management Information Systems Quarterly*, *37*(2), 337–355.

Hattie, J., & Timperley, H. (2007). The Power of Feedback. *Review of Educational Research*, *77*(1), 81–112.

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design Science in Information Systems Research. *Management Information Systems Quarterly*, *28*(1), 75–105.

Iivari, J. (2015). Distinguishing and contrasting two strategies for design science research. *European Journal of Information Systems*, *24*(1), 107–115.

Kluger, A. N., & DeNisi, A. (1996). The effects of feedback interventions on performance: A historical review, a meta-analysis, and a preliminary feedback intervention theory. *Psychological Bulletin*, *119*(2), 254–284.

Kokkelenberg, E. C., Dillon, M., & Christy, S. M. (2008). The effects of class size on student grades at a public university. *Economics of Education Review*, *27*(2), 221–233.

March, S. T., & Smith, G. F. (1995). Design and natural science research on information technology. *Decision Support Systems*, *15*(4), 251–266.

McAfee, A., & Brynjolfsson, E. (2016). Human Work in the Robotic Future: Policy for the Age of Automation. *Foreign Affairs*, (95(4)), 139–150.

Meth, H., Mueller, B., & Maedche, A. (2015). Designing a Requirement Mining System. *Journal of the Association for Information Systems*, *16*(9), 799–837.

Nicol, D. J., & Macfarlane-Dick, D. (2006). Formative assessment and self-regulated learning: a model and seven principles of good feedback practice. *Studies in Higher Education*, *31*(2), 199–218.

Nunamaker, J. F., Jr., Chen, M., & Purdin, T. D. M. (1990). Systems Development in Information Systems Research. *Journal of Management Information Systems* *7*(3), 89–106.

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management Information Systems*, *24*(3), 45–77.

Piccoli, G., Ahmad, R., & Ives, B. (2001). Web-Based Virtual Learning Environments: A Research Framework and a Preliminary Assessment of Effectiveness in Basic IT Skills Training. *Management Information Systems Quarterly*, *25*(4), 401–426.

Piccoli, G., Rodriguez, J., Palese, B., & Bartosiak, M. (2017). The Dark Side of Digital Transformation: The Case of Information Systems Education. *ICIS 2017 Proceedings*, Seoul, South Korea.

Pigni, F., Piccoli, G., & Watson, R. (2016). Digital Data Streams. *California Management Review*, *58*(3), 5–25.

Reay, T., Golden-Biddle, K., & Germann, K. (2006). Legitimizing a new role: Small wins and microprocesses of change. *Academy of Management Journal*, *49*(5), 977–998.

Rodriguez, J., Piccoli, G., & Bartosiak, M. (2019). Nudging the Classroom: Designing a Socio-Technical Artifact to Reduce Academic Procrastination. *Proceedings of the 52nd Hawaii International Conference on System Sciences*, Maui, HI.

Sadler, D. R. (1998). Formative Assessment: revisiting the territory. *Assessment in Education: Principles, Policy & Practice*, *5*(1), 77–84.

Sapelli, C., & Illanes, G. (2016). Class size and teacher effects in higher education. *Economics of Education Review*, *52*(C), 19–28.

Sein, M., Henfridsson, O., Purao, S., Rossi, M., & Lindgren, R. (2011). Action Design Research. *Management Information Systems Quarterly*, *35*(1), 37–56.

Silver, M., & Markus, M. L. (2013). Conceptualizing the SocioTechnical (ST) artifact. *Systems, Signs & Actions*, *7*(1), 82–89.

Simon, H. A. (1996). *The sciences of the artificial*. Cambridge, MA: MIT Press.

Sunstein, C. R., & Thaler, R. H. (2003). Libertarian Paternalism Is Not an Oxymoron. *The University of Chicago Law Review*, *70*(4), 1159–1202.

Venable, J., Pries-Heje, J., & Baskerville, R. (2016). FEDS: a Framework for Evaluation in Design Science Research. *European Journal of Information Systems*, *25*(1), 77–89.

Walls, J. G., Widmeyer, G. R., & El Sawy, O. A. (1992). Building an Information System Design Theory for Vigilant EIS. *Information Systems Research*, *3*(1), 36–59.

Weick, K. E. (1984). Small wins: Redefining the scale of social problems. *American Psychologist*, *39*(1), 40–49.

Weinmann, M., Schneider, C., & Brocke, J. vom. (2016). Digital Nudging. *Business & Information Systems Engineering*, *58*(6), 433–436.

Yoo, Y. (2010). Computing in Everyday Life: A Call for Research on Experiential Computing. *Management Information Systems Quarterly*, *34*(2), 213–231.

Table 1: Summary of requirements, principles, and features of the ST artefact

Meta-Requirements	**Design Principles**	**Design Features**
MR1: A performance feedback system for practical digital skills accurately records learner behaviours and performance	DP1.1: Collect completion data for required and optional assignments	DF1: Individualized login: Learners use their Slack account to gain access to the application
	DP1.2: Collect learning resources utilization data (online or in person) for required and optional resources	DF2: Activity logging: Google Analytics, system logs, and manual tracking is used to log users' activity
	DP1.3: Reliably measure performance ensuring consistent evaluation of the same task over time and across learners	DF3: Reliability: The grading engine uses custom Python code to ensure consistent evaluation of learners' work
	DP1.4: Produce valid measures of skill mastery (i.e., minimizing false positive and false negative evaluation errors)	DF4: Validity: The grading engine uses custom Python code to ensure accurate evaluation of learners' work
MR2: A performance feedback system for practical digital skills does not conflate behaviour with learning	DP2.1: Assignments and homework are a service to learners and have no bearing on the individual's learning assessment	DF5: Learner control: While evaluated if submitted, assignments do not contribute to the computation of the final grade
	DP2.2: Learning assessment is measured, independently of learner behaviour, through dedicated ad-hoc evaluations	DF6: Mastery evaluation: Exams cover all skills learned during the course and are the sole measure of skills mastery
MR3: A performance feedback system for practical digital skills treats learners as self-responsible units and maximizes learner control	DP3.1: Provide regular homework and practice assignments to learners to test their progress in skills acquisition	DF7: Skills progression: Microsoft Excel (6) and Word (3) assignments are designed to build a progression of skills
	DP3.2: Provide feedback to learners for all the assignment learners voluntarily submit	DF8: Performance feedback: The grading engine uses custom Python code to provide automated evaluation of each task in the assignments
	DP3.3: Direct learners toward appropriate resources, online or physical, for specific skills that are not mastered	DF9: Guidance and support: Links to the official documentation and open labs where learners receive help
MR4: A performance feedback system for practical digital skills contextualizes behavioural and performance data for learners as soon as it becomes available	DP4.1: Provide feedback immediately after task completion	DF10: Feedback immediacy: Near real-time evaluation of learners' work
	DP4.2: Enable learners to interpret the feedback by contextualizing it appropriately	DF11: Contextualized feedback: Performance evaluations are linked to expected skill targets
MR5: A performance feedback system for practical digital skills fosters a sense of progress in the learners	DP5.1: Create realistic practice exercises that learners view as instrumental to their future success	DF12: Realism: Assignments present realistic tasks to be completed within Microsoft Excel (6) and Word (3)
	DP5.2: Create manageable practice exercises that fit easily within the schedule and habits of the learners	DF13: Manageability: Assignments are chunked into sets completable in single sessions

	DP5.3: Create practice exercises that are limited in scope and enable learners to focus on specific skills and receive precise feedback	DF14: Structure: Use of structured data files to ensure limited scope and precise feedback

Table 2: Assignment Completion Rate (Pilot Implementation)

Word 1	Word 2	Word 3	Excel 1	Excel 2	Excel 3	Excel 4	Excel 5	Excel 6
69.44%	44.44%	36.11%	34.38%	10.34%	13.79%	17.24%	3.45%	6.90%

Table 3: Comparison of Grading Engine and Human Raters

	Excel				Word			
	Task Evaluation %		Absolute Score Error		Task Evaluation %		Absolute Score Error	
Rater	**Correct**	**Incorrect**	**Overall**	**Average per File**	**Correct**	**Incorrect**	**Overall**	**Average per File**
Grading Engine	98.38	1.62	51.5	1.98	92.87	7.13	28.75	1.31
Human Rater 1	93.08	6.92	165.0	6.35	88.48	11.52	52.08	2.37
Human Rater 2	95.23	4.77	135.0	5.19	88.48	11.52	48.13	2.19
Human Rater 3	96.00	4.00	85.0	3.27	89.76	10.24	40.65	1.85

Table 4: Summary of Grading Engine Performance (Second Iteration)

	Microsoft Excel		Microsoft Word	
	Number	**Percentage**	**Number**	**Percentage**
Tasks	27		14	
Evaluated elements per exam	50		25	
Total tasks evaluated	1,300		550	
Total errors	21	1.62%	39	7.09%
False positives	15	1.15%	20	3.64%
False negatives	6	0.46%	19	3.45%

Table 5: Assignment Completion Rate (Second Iteration)

	Excel 1	**Excel 2**	**Excel 3**	**Excel 4**	**Excel 5**	**Excel 6**	**Word 1**	**Word 2**	**Word 3**
Feedback Percentage	78.57%	57.14%	60.71%	46.43%	50.00%	39.29%	62.96%	51.85%	37.04%
Differential with first iteration	9.13%	12.70%	24.60%	12.05%	39.66%	25.50%	45.72%	48.40%	30.14%
Average Chunk completion	2.86/4	4.88/5	3.06/4	3.69/5	2.57/5	4.18/5	2.13/4	1.56/4	0.79/3
Percentage	71.59%	97.50%	76.47%	73.85%	51.43%	83.64%	53.26%	39.06%	26.32%

Table 6: Summary of Grading Engine Performance (Third Iteration)

	Microsoft Excel		Microsoft Word	
	Number	**Percentage**	**Number**	**Percentage**
Tasks	18		12	
Evaluated elements per exam	32		23	
Total tasks evaluated	8,608		5,727	
Total errors	269	3.13%	293	5.12%
False positives	59	0.69%	183	3.20%
False negatives	210	2.44%	110	1.92%

Table 7: Assignment Completion Rate (Third Iteration)

	Excel 1	**Excel 2**	**Excel 3**	**Excel 4**	**Excel 5**	**Excel 6**	**Word 1**	**Word 2**	**Word 3**
Feedback Percentage	96.80%	82.50%	60.00%	64.60%	42.50%	39.60%	66.70%	45.30%	29.60%
Differential with second iteration	18.23%	25.36%	-0.71%	18.17%	-7.50%	0.31%	3.74%	-6.55%	-7.44%
Average Chunk completion	2.88/4	3.38/5	2.93/4	3.56/5	3.81/5	3.81/5	3.17/4	3.06/4	2.54/3
Percentage	72.20%	67.50%	73.30%	71.20%	76.20%	76.20%	79.20%	76.40%	84.70%

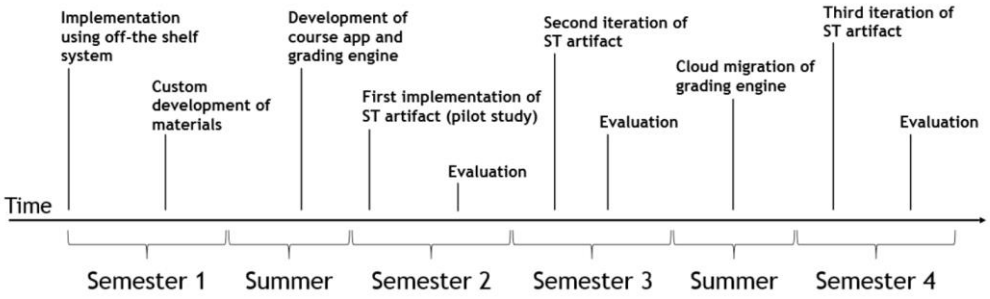

Figure 1

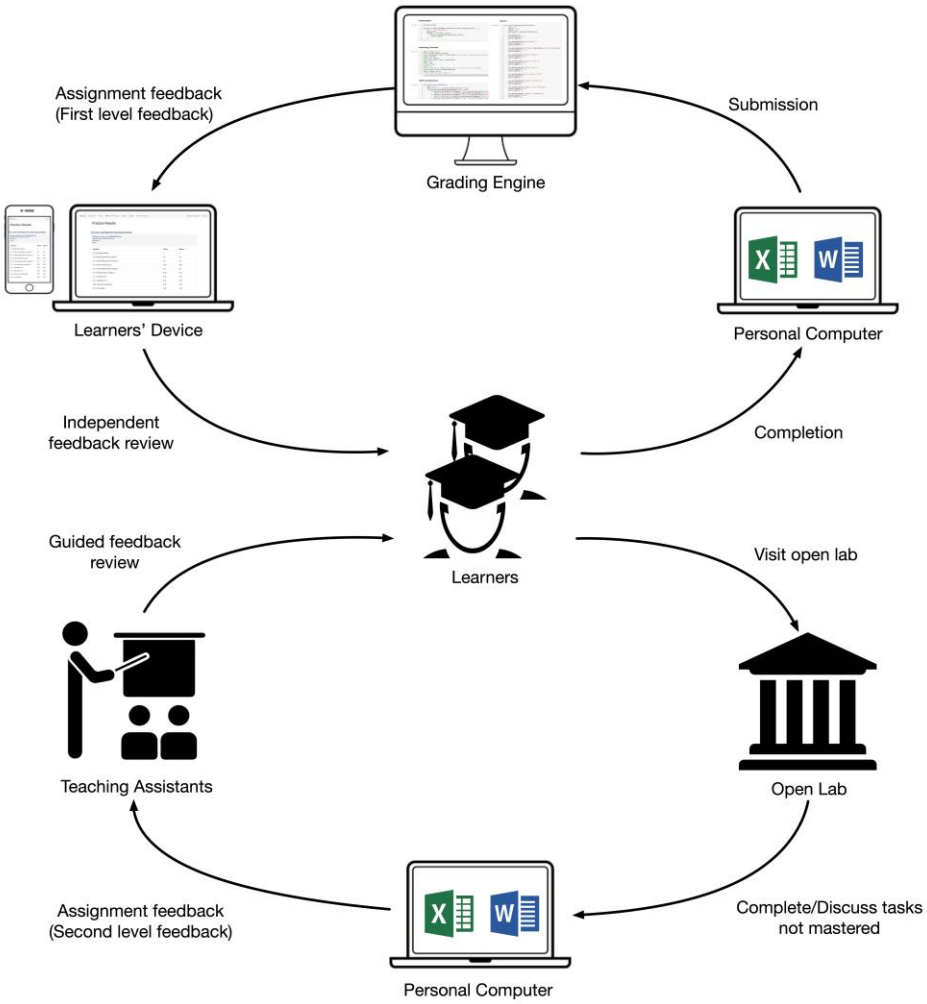

Figure 2

Question	Points	Earned
4 - Copy data in columns B, F and G	0.5	0.5
4.c1 - Rename Sheet1	0.25	0.0
4.c2 - Rename Sheet2	0.25	0.25
5.a - Compute the Gross Sales per unit	0.25	0.25
5.b - Format the header	0.25	0.12
5.c - Use relative references to drag the function	0.5	0.0
6 - Type Total and Average	0.2	0.2
6.a - Format cells in bold	0.2	0.2
6.bc - Copy and paste the formulas	0.4	0.2
6.d - Formula range correct	0.2	0.2
7 - Calculate the average	1	0.5
8 - Use a Top Border	1	1.0

Figure 3

Initialization

In [1]:
```
%%javascript

Jupyter.keyboard_manager.command_shortcuts.add_shortcut('/', {
    help : 'run all cells',
    help_index : 'zz',
    handler : function (event) {
        IPython.notebook.execute_all_cells();
        return false;
    }}
);
```

Importing Libraries

In [2]:
```
import numpy as np
from xml.dom import minidom
from prettytable import PrettyTable # pip install https://pypi.python.o
import zipfile
import os, shutil
from collections import defaultdict
import re
import math
# import time
import csv
#import enchant # unfortunately only is built for 32-bit python
from difflib import SequenceMatcher
import pandas as pd
# from Exams_Fall2017 import *
```

Styles Dictionaries

In [3]:
```
def borderStyle(position):
    tDic = {}
    tDic["style"] = position.getAttribute("style")
    if position.getElementsByTagName("color")[0]:
        if position.getElementsByTagName("color")[0].hasAttribute("them
            tDic["colorTheme"] = position.getElementsByTagName("color")
        if position.getElementsByTagName("color")[0].hasAttribute("tint
            tDic["colorTint"] = position.getElementsByTagName("color")[
        if position.getElementsByTagName("color")[0].hasAttribute("rgb"
```

Exam 1

In [29]:
```
def exam1(filename,styleFile,df):
    ans = []
    qLabel = []
    point = []
    definedSet = getStyle(styleFile)

    ans.append(0)
    qLabel.append("1")
    point.append(0)

    ans.append(grade("fontScheme"))
    qLabel.append("2")
    point.append(1)

    ans.append(grade("styleSet",definedSet,["title","heading1",
    qLabel.append("2.a")
    point.append(1)

    ans.append(grade("title","caps"))
    qLabel.append("3.a")
    point.append(1)

    ans.append(grade("title","color"))
    qLabel.append("3.b")
    point.append(.5)

    ans.append(grade("title","size"))
    qLabel.append("3.c")
    point.append(.5)

    ans.append(grade("doc","pStyle","0"))
    qLabel.append("4")
    point.append(1)

    ans.append(grade("header","picture"))
    qLabel.append("5")
    point.append(1)

    ans.append(grade("header","jc","drawing"))
    qLabel.append("5.a")
    point.append(.5)

    ans.append(grade("header","picHeight"))
    qLabel.append("5.b1")
    point.append(.5)

```

Figure 4

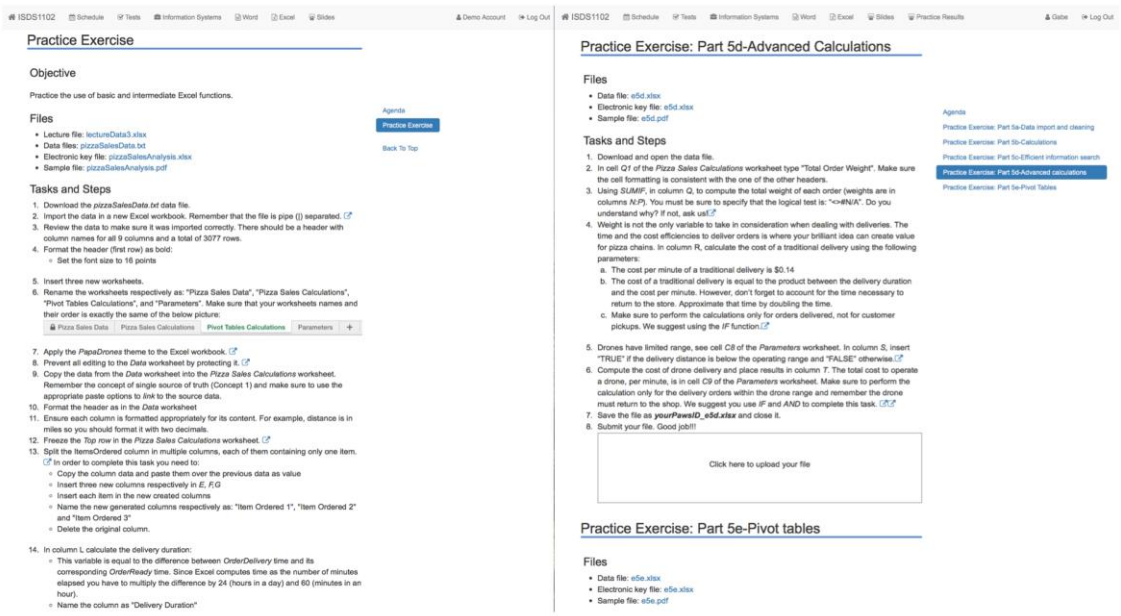

Figure 5

Figure captions:

1. Figure 1: Project Timeline
2. Figure 2: ST artefact design diagram
3. Figure 3: Results page for an assignment chunk
4. Figure 4: Sample grading engine code
5. Figure 5: Chunked assignments with drop file feature