Peek-through Customization: Example-Based In-Context Sharing for Learning Management Systems

Laton Vermette
School of Computing Science
Simon Fraser University
Burnaby, BC, Canada
lvermett@sfu.ca

Joanna McGrenere
Dept. of Computer Science
University of British Columbia
Vancouver, BC, Canada
joanna@cs.ubc.ca

Parmit K. Chilana
School of Computing Science
Simon Fraser University
Burnaby, BC, Canada
pchilana@cs.sfu.ca

ABSTRACT

Learning to use a learning management system (LMS) can often be complex and challenging for instructors who have little time to explore the application interface. We ran formative interviews (N=10) suggesting that instructors often prefer to consult colleagues to seek examples of customizations and their explanations. Based on our findings, we designed and developed *Customizer*, an in-context example-based customization sharing platform that runs atop a widely-used LMS and facilitates discovery of relevant customizations shared by peers. Customizer allows instructors to experiment with shared customizations in familiar contexts by copying their course content into on-the-fly testing environments, minimizing any risk of breaking their live course setups. Our usability evaluation (N=10) showed that most users found Customizer intuitive and useful, and that its exploratory interface was helpful in a variety of use cases. Furthermore, many participants saw potential for Customizer to improve their workflows in other applications beyond educational contexts.

Author Keywords

Software customization; Example-centric customization; Customization sharing; Learning management systems

CCS Concepts

•**Human-centered computing** → **Human computer interaction (HCI);**

INTRODUCTION

At both post-secondary and K-12 schools worldwide, instructors are increasingly using learning management systems (LMS) to fulfill a variety of educational needs [1, 14]. To adapt the features and settings of an LMS to their course, instructors often have to perform some amount of *software customization* (e.g., tailoring to the course subject, grade level, teaching methods, or student accessibility needs). Widely-used LMSs tend to be feature-rich with many customizable

options, presenting a steep learning curve that instructors must overcome to make effective use of them. A key challenge that instructors have to tackle is taking time out of their busy schedules to figure out how to navigate and customize their LMS and other educational technologies [38].

In recent years, different LMS-specific support systems have emerged, such as online help, documentation, and hands-on training [18]. Many instructors nonetheless prefer social forms of help [41], such as "over-the-shoulder" learning [39], allowing them to get in-person task-specific advice from a more experienced colleague by looking at how they have customized their LMS. However, this type of learning can be time-consuming and challenging to coordinate among busy instructors, so cannot always be relied upon.

Even beyond educational technologies, a common challenge with customizing feature-rich software is that users are hesitant to take the risk of customizing [33]. In addition to the initial effort and time investment needed to customize, users are also concerned about the potential troubleshooting that will be required if the customization goes wrong. Instructors, in particular, are a class of users who routinely face difficult troubleshooting challenges in experimenting with their digital classroom setups [41].

Given the difficulties instructors have with relying on in-person help, we wondered about the possibility of alternative semi-social approaches to support software customization. In particular, we were curious about this question: if instructors had tool support to "peek" at and experiment with examples of their colleagues' course customizations in a risk-free manner, within their own LMS, would it help them more easily learn how to customize their own courses?

We began by conducting a formative interview study with post-secondary instructors aimed at better understanding how to support their strategies for customizing LMSs. These interviews elicited feedback on some early design concepts for LMS customization sharing tools. The findings highlighted instructors' frequent use of examples from other instructors' courses as templates and inspiration, but also the degree to which they rely on exploratory learning to fill in the gaps when social help is unavailable. In light of these findings, we sought to facilitate this exploration of examples by designing *Customizer*, a novel platform to support *example-based customization sharing* (Figure 1). Using Customizer, instructors can
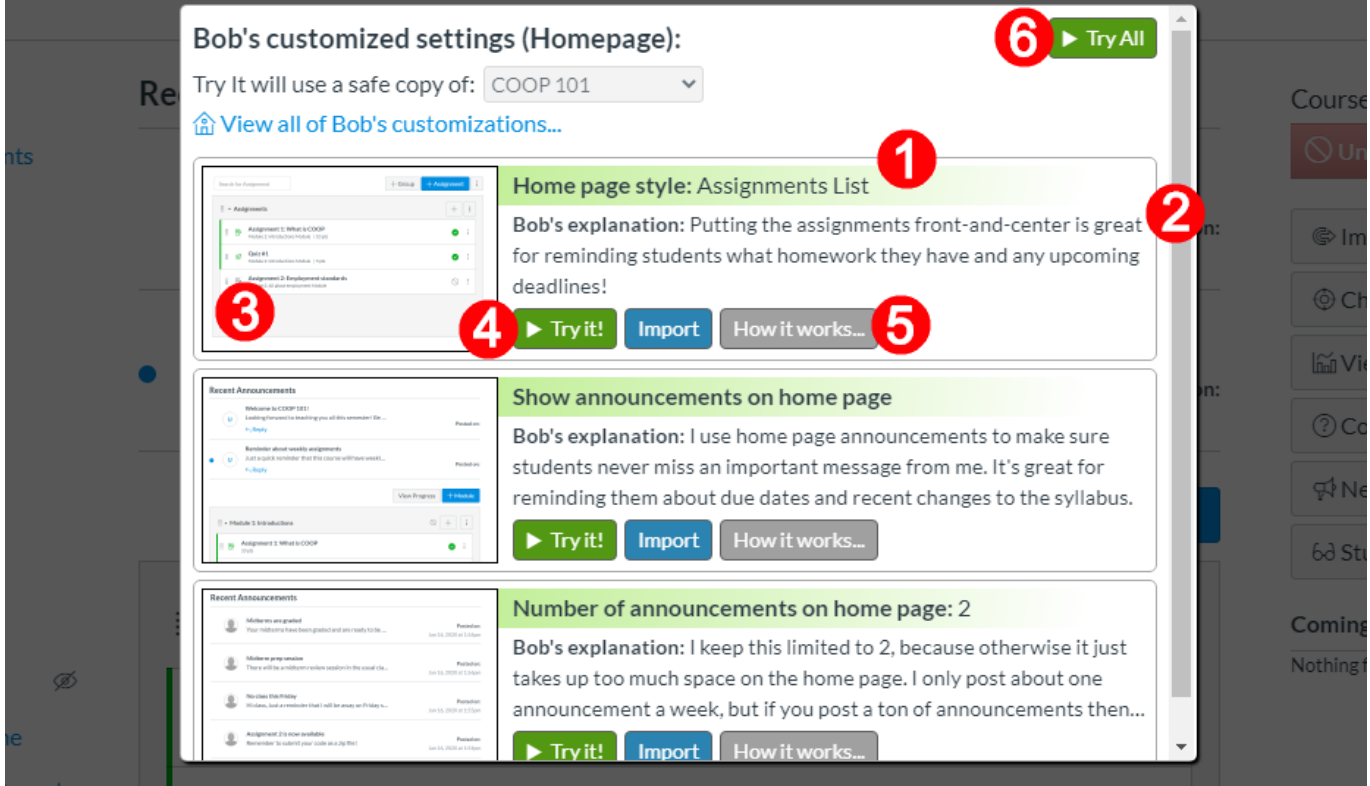
Figure 1. The Customizer dialog opens on top of a user's course homepage in the Canvas LMS, displaying three customizations that another instructor (Bob) has made to his own course homepage. The user can browse and explore each of Bob's customizations. Each "customization card" contains (1) the name of the customization, (2) Bob's freeform written explanation for it, (3) a preview image of the result, and (4) buttons to *Try It* in a safe test environment or *Import* it into the current course. The *How it works* (5) button expands out longer explanations and shows extra metadata about the customization's complexity and where it takes effect. The user can try all (6) of Bob's customizations at once. The user can also try any combination of the customizations together, by clicking multiple cards to select them.

discover examples of peer-generated customizations relevant to the LMS page they are on. Moreover, Customizer provides instructors with an *exploratory mode* [25,32] inside their LMS, letting them "peek and try" these shared customization ideas without the risk of breaking their real courses. Customizer also provides tools for instructors to analyze and reflect upon their own customizations and share them with colleagues.

To evaluate this design, we developed a prototype running atop *Canvas*, a widespread LMS with currently more than 30 million users [35]. We ran an observational usability study with 10 post-secondary instructors in which they expressed the usefulness of having access to in-context customization examples and an environment in which they could safely explore how the changes would affect their course. In addition to identifying many ways to improve the design, participants described several different use cases for the exploratory mode beyond the original design intention, and were enthusiastic about generalizing Customizer to other non-LMS applications.

The main contributions of this paper are: (1) empirical insights into how instructors experience the process of setting up and personalizing an LMS, and how they share their knowledge and artifacts during that process; (2) the design and implementation of a novel example-based, in-context customiza-

tion sharing platform that allows instructors to discover example customizations for their courses, freely experiment with those possibilities in on-the-fly sandboxed copies of their real courses, and easily share their own customizations with others; and, (3) a usability study that provides preliminary support for the usefulness of our "peek and try" conceptual model, as well as several ways in which the design could be extended.

RELATED WORK
To contextualize our empirical findings and designs, we draw upon literature on instructors' usage and perceptions of digital classroom tools, example-based peer-to-peer knowledge sharing, and insights into appropriating and sharing software customizations.

How instructors use and perceive digital classroom tools
Studies on instructors in educational technology and HCI fields have contributed numerous insights into how learning management systems and other digital classroom tools are used and perceived. For example, Lai and Savage [24] discuss the roles, advantages, and shortcomings of LMSs in higher education, arguing that educators need to innovate to make the most effective use of their LMS, and that it is not enough to use them as "mere information transmission tools". While

a great deal of the literature on educational technology focuses on the student perspective, some research has looked into instructors' perceptions and motivations on technology integration [15, 18, 38], the design of classroom technologies to better support educators' needs [5, 6, 7], or more generally how instructors discover and integrate new software into their personalized "digital classroom ecosystems" [41].

Our paper complements this existing work by investigating the practices specific to customizing an LMS and designing a novel instructor-centered platform for sharing in-context examples of LMS customizations.

Example-based peer-to-peer knowledge sharing
It has been well-studied how educators often form *communities of practice* to share ideas and insights with one another [37, 42] and rely on their social connections to learn how to use and customize their digital classroom tools [41]. Within these peer groups, it is common to learn about software features and seek help through ad-hoc in-person interactions (e.g., looking at examples of a colleague's software via over-the-shoulder learning [39]), but this can be time-consuming and frustrating for the tech-savvy teachers who end up fulfilling an "unofficial tech support" role for colleagues on top of their teaching duties [41]. Our work seeks to alleviate these frustrations by surfacing peer-authored customization examples within an instructor's LMS and letting them remotely explore the effects of trying out their colleagues' customizations.

We draw inspiration from general-purpose systems that have built in-context crowdsourced software tutorials [30, 40], Q&A [13, 30], and action/command suggestions [17, 31] into web-based interfaces as a shareable help mechanism. Other systems like Unakite [26] provide the means for programmers to collect and share web-based code snippets with information about the trade-offs of different code solutions. Moreover, there is a long history of using example-based aids to assist programmers, both for novices in classroom environments [10, 19] and for professionals working with an integrated development environment (IDE) [22, 34]. Notably, Brandt et al. designed the *Blueprint* interface for retrieving example-centric code snippets based on context from the user's code within an IDE [9].

Our Customizer design takes these ideas beyond retrieving static examples for viewing, allowing the user to interactively experiment with LMS customization examples in the context of their own courses, and directly import those examples into their courses.

Appropriating and sharing software customizations
Early research into software customization highlighted the many triggers and barriers that influence users' customization practices [28]. Software customizations have since been characterized and compared according to a number of dimensions, such as what aspects of the user interface they modify [11, 20] and the degree of user control over the changes [16, 33]. Some studies have looked more closely at the personal side of customization, identifying key groups of users such as "translators" [27] and "tinkerers" [29] who in many ways bridge the gaps between expert customization authors and less experienced users. Furthermore, there has been evidence suggesting

the importance of seeing a customization in action [16] and reading the rationale behind it [12] to improve users' understanding of its benefits. It has been proposed that feature-rich applications can help users by offering an *exploratory mode*, where they are free to tinker with interface features and learn in a risk-free manner atop a "dummy document" [25, 32]. To our knowledge, Customizer is among the first systems to fully implement this concept.

Importantly, several works have studied software customization not an individual activity, but as a highly social one in which sharing occurs frequently. Mackay [27] detailed how software configuration files get shared and modified among colleagues within a company. More recently, Haraty et al. [21] explored how user communities come together in online *customization sharing ecosystems* that let them upload and share their customizations with one another. We also see parallels between our work and other domain-specific approaches to customization sharing, such as the ShareXP system [8] for software developers to share customized IDE perspectives.

Although past literature has uncovered the barriers that make it challenging to share customizations in general, software customization practices in the context of an LMS are underexplored. In fact, classrooms present an atypical scenario because the changes instructors make to their LMS and other classroom are meant to improve *students'* software experience, not necessarily their own [41]. We aim to partially address this gap with our studies of instructors' approaches to customization and their use of a novel customization-sharing tool.

FORMATIVE STUDY
To gather insights into LMS customization habits and to inform the design of our system, we conducted interviews with 10 participants who were instructors or teaching assistants at a large public university and had recently used and configured an LMS for their course. Our high-level research questions for this study were:

RQ1: How do post-secondary instructors and TAs learn to use and customize learning management systems?

RQ2: What methods do post-secondary instructors and TAs see as promising or useful for exploring the customizable features of a learning management system?

Participants
Our participants (4F, 6M) were all between the ages of 19–60 and had used an LMS within the past four months to teach a university-level course, either as the instructor or as a TA. Our goal was to have a mix of instructors and TAs with a range of experience. To that end, we took a hybrid sampling approach, drawing on the personal contacts of the research team and advertising on two different university mailing lists, followed by snowball sampling to recruit the remaining participants. The 3 instructors (P01, P08, and P10) had 11–15, 21+, and 16–20 years of teaching experience, respectively. Among the 7 TAs, 3 had less than a year of teaching experience, and the remaining 4 had 1–5 years.

The majority (6/10) of participants had been teaching a computer science or math course, with the others teaching design,
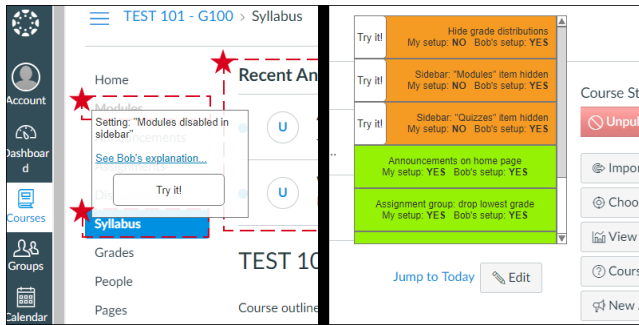
Figure 2. Two examples of the design mockups shown to participants. Left: mockup 1, showing badges and explanations on customized features. Right: mockup 2, a high-level list comparing course customizations between two instructors.

business, economics, or mechanical engineering. All participants described having used Canvas (the LMS deployed by their university) in their teaching role, and most (8/10) were also using other educational technology in their class beyond an LMS (e.g., Crowdmark, iClickers, XAMPP).

Procedure
We began by asking participants to fill out a brief demographic questionnaire. We then carried out semi-structured interviews, each lasting 30–45 minutes, asking about their past learning and usage of LMSs, their routine for setting up and customizing a new course, their tendency to share their configurations with peers and colleagues, and common challenges they had encountered throughout these tasks.

Following this, we showed them 3–4 different mockups of novel design ideas for sharing and browsing customizations within Canvas LMS, eliciting their feedback about the usefulness of each design and their ideas for improvement, both verbally and in drawn sketches. These mockups evolved over the course of the formative study, as we iterated on the designs to incorporate the most frequent feedback into them. They included designs which (1) visually marked customized features of an instructor's Canvas course with red borders, icons, and explanations from the creator; (2) added a panel listing out a high-level overview of the customized features throughout the course; (3) displayed an interactive split-screen of two differently-customized courses overlaid on top of each other; and (4) added a sidebar showing some customizations that other instructors had done. The mockups presented were partially interactive (as *Axure* prototypes overlaid atop a Canvas screenshot), and participants were encouraged to play with them and imagine how they might work in practice. The first two mockups are shown in Figure 2.

The interviews were audio-recorded and later transcribed to text for analysis. Participants were each offered a $20 gift card in appreciation of their time (3 declined).

Key Findings and Design Goals
Participants described several ways in which they had learned to use the features of an LMS, but among the most common was exploratory learning, where they would simply try out interface features and explore the interface on their own until

they felt they had a reasonable understanding of the range of features and how to use them. The second most common learning method was asking for task-specific help or instructive examples from a more knowledgeable colleague. However, these interactions were described as time-consuming and highly dependent on personal availability.

Sometimes it helps to just know what other people are doing... I know I can find a solution based on what other people have done and I can try it and see if it works, but sometimes it's just a really specific thing and I can't get what I need from the Internet. (P05)

Visual customization cues are very helpful
Among the design mockups shown, participants consistently noted a preference for ones that attached visual markers to customized elements of the page (e.g., Fig 2, left). While they had some concerns that there might be too much visual clutter in certain cases, the utility of these visual cues for *discovering* what has been customized was generally seen to outweigh the potential for clutter.

The thing I like about this is the red boxes with the star [indicating customized elements] because I don't have to worry about other jargon...if I only know what things they've changed, that actually helps because I know where to look instead of having to scan through each item of a list...it's really helpful. (P05)

In contrast, the idea of listing customizations in a central location *without* visual referents was mainly deemed helpful to self-identified "power users" who were fond of having all the information in one place, while others felt it could be overwhelming unless there were robust ways to filter it according to location or difficulty.

Instructors value expert explanations
Both instructors and TAs overwhelmingly felt that having a written explanation of *why* a customization was made would be very useful, particularly if it was written by the original creator detailing their intentions and experiences.

I think [explanations] are very helpful because sometimes it's not obvious... especially for someone who's less knowledgeable about Canvas, it's a bridge between pedagogical goals and Canvas features...If there was a way for me to view a course that had almost like a bit of a commentary or a viewpoint about pedagogical benefits or approaches for features that are being used, that would be extremely valuable. (P08)

Participants had similar feelings about adding additional contextual information like what scope of the application is affected, for a clearer idea of how extensive a change is and what it does. Furthermore, several participants also added that they could benefit from seeing a step-by-step explanation of *how* to set up each customization manually, often for the sake of learning more about the LMS features:

Give me a point-by-point explanation, or steps... like it's awesome if you just give me the whole [course], but I would like to learn too, I would like to know how to go about it and make alterations to it, just so that I'm completely familiar with what I am doing to my own course page. (P07)
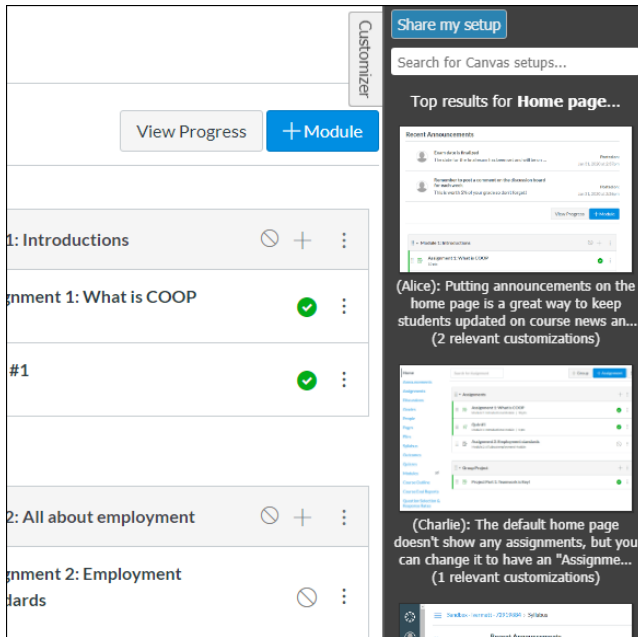
Figure 3. The Customizer sidebar, here shown opened on a Canvas course home page, lists several shared course customizations that are relevant to the user's current page. Each customization shows a preview of the visual result and the author's explanation. Clicking one of these customizations opens the full Customizer dialog (see Figure 1) with the clicked item highlighted. The user can also use the search bar to quickly find other instructors' customizations by freeform query. The means for them to share customizations from their own courses is provided as well.

Based on these findings, we derived four design goals for a system to help instructors find and share LMS customizations:

DG1: Encourage exploration. Because exploratory learning was so common among instructors, the system should allow users to explore and see the effect of different customizations.

DG2: Facilitate discovery of relevant examples. Given instructors' tendency to seek out example customizations for help and inspiration, the system should suggest relevant examples of customizations from other instructors.

DG3: Highlight customizations with visual cues. The system should take advantage of the visual nature of many customizations to make them stand out at a glance.

DG4: Enhance customizations with rationale and context. Instructors who share their customizations should be encouraged to provide information on *why* and *how* they made those changes to help other users assess relevance and difficulty.

SYSTEM DESIGN OF CUSTOMIZER
After several rounds of brainstorming, sketching, and iterative prototyping based on the above goals, we arrived at a design for Customizer, split across three main modes of interaction. Addressing DG2 and DG4, Customizer's *Discovery interface* provides a streamlined way to find new customizations and explanations that others have shared without having to leave the application user interface. Addressing DG1 and DG3, the *Exploratory interface* provides a safe interactive testbed environment for experimenting with customizations shared by oth-

ers. Lastly, Customizer also includes an *Authoring interface* which allows users to easily share their own customizations and add explanations.

We selected Canvas as the underlying LMS for exploring the design space since it has widespread adoption in both post-secondary and K-12 schools [35] and is available as an open-source project [2]. Like other LMSs, Canvas consists of many smaller sections (e.g., Assignments, Quizzes, Discussions) each with many individual pages, encompassing hundreds of ways instructors can further customize their courses.

Discovery Interface for finding customizations
As the entry point into Customizer, the discovery interface (Figure 3) enables users who *want* to customize to find relevant ideas for what to do and explanations for why and how to do it. We adopted a design similar to systems such as *Social CheatSheet* [40] and *Unakite* [26], adding a collapsible sidebar on the right side of Canvas web pages that lists customizations relevant to the current page, that have been shared by other instructors. The relevant customizations are presented with a preview image and a brief explanation written by the author.

Browsing and filtering by context
Customizer automatically uses the page URL as context information to determine which customizations are relevant to show in the sidebar. Navigating to a different page of the Canvas LMS will trigger the sidebar to update its contents with customizations that affect the user's current page or closely-related sections of the LMS. Alternatively, the user can also search through all of the shared customizations via a freeform query in the sidebar's search field, which matches customizations based on their descriptions and explanations. In this case, the sidebar (Figure 3) would list the search results instead of the page-filtered customizations (but customizations relevant to the user's current page are still ranked higher in the list).

Interactive customization cards
When the user clicks on one of the customizations in the sidebar, Customizer overlays a dialog on the user's current page showing a list of interactive customization "cards" (Figure 1). By default, the list shows all customizations shared by the same author that can affect the current page. Upon the dialog opening, the list automatically scrolls to and highlights the customization that was selected from the sidebar. From there, the user can easily browse any closely-related customizations from that author's *customized setup* (taken here to mean the aggregation of all the customizations applied to their course).

Each of the customization cards provides a descriptive name for the customization (e.g., "Don't let students attach files to discussions") and shows an image preview of what the customization does. Fulfilling DG4, it also shows a written explanation of the customization author's rationale for including this in their setup. For longer explanations, the full text can be opened by clicking the *How it works* button on each card. If the author did not provide any explanation of their own, a default description of the customization's purpose is provided as a placeholder in most cases, adapted manually from Canvas documentation guides [23]. The card header is shown with a background color indicating how complex the author felt
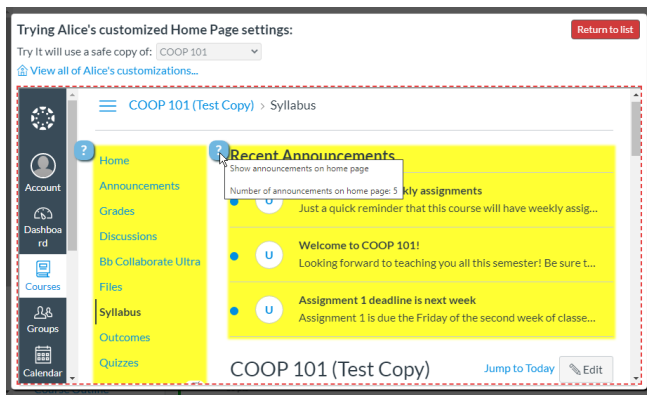
Figure 4. The *Try It* feature spawns an on-the-fly interactive testing copy of the user's course content, with another instructor's customizations applied to it. Those customizations are highlighted in yellow and marked with a small badge that provides more information. Here it shows a customized navigation bar and the addition of a Recent Announcements section on the Syllabus page.
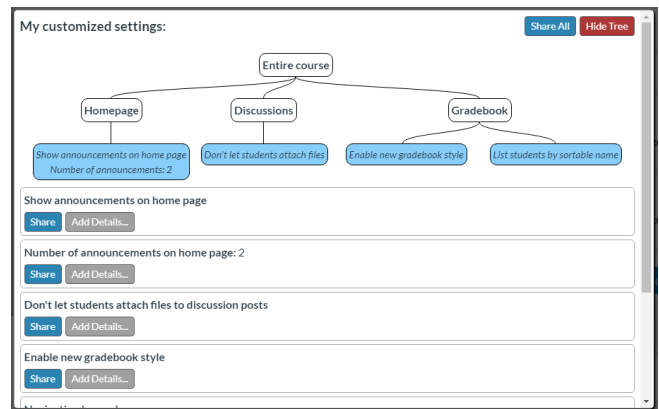

Figure 5. The authoring interface lets users analyze their course customizations, add explanations, and share them with others. The tree view visualizes where in the interface those customizations take effect.

this customization was to set up or maintain (green, yellow, or red), to help decide whether the customization is appropriate for their Canvas expertise level.

The list is initially filtered to show only customizations related to the context of the user's current page. These contexts (such as "Discussions", "Gradebook", and "Assignments") connect the available customizations to distinct areas of LMS functionality for ease of navigation and to avoid overwhelming the user. Additionally, the dialog header has a link to *Show all customizations by <author>* if the user wants to see more.

The customization cards are selectable and allow multiple selections at once for quickly importing or trying out combinations of customizations together.

Finally, each customization card has a *Try It* button to immediately test out the customization, and an *Import* button allowing the user to immediately copy any customization into one of their own courses (explained further below).

Exploratory Interface for trying out customizations
Once the user has discovered a customization that interests them, they can import that customization directly into one of their own courses, or use Customizer's *Try It* feature (Figure 4) to safely experiment with the customization before they commit to making any changes.

"Try It" feature for risk-free interface exploration
Customizer implements an exploratory mode [25, 32] for the Canvas LMS by creating an on-demand "sandbox" environment in which new customizations can be experienced and experimented with safely (i.e., without the risk of making inadvertent changes to one's live courses). The exploratory interface creates a new course on-the-fly for these testing purposes, mimicking the user's existing course by copying its content and settings, and then applying the selected customizations on top. Prior to clicking *Try It* (Figure 1.4), the user can select which of their courses they want to mimic, or can opt to try it in a new blank course with only the default content. Within this generated sandbox course, the user is now

free to make any changes they want without worrying about unintended side-effects, as the copy is entirely separate from their real courses and not linked in any way. Following DG3, Customizer highlights any page elements in the sandbox that have been altered by the active customizations, and adds a small badge indicating which customization is causing the effect.

From exploration to implementation
If the user is satisfied that the customization they have just tried will work well for their course, they can click the *Import* button to apply that customization to one of their real courses. A warning popup is first displayed to ensure that the user understands that the customization will potentially be visible to other course participants immediately, and encourages them to test it with the *Try It* feature first. In this way, users can make use of the exploratory mode to experiment with a variety of features and then keep only the ones they liked.

Authoring Interface for sharing customizations
For instructors who *have customized* their courses (or are in the process of doing so), an authoring interface is provided for them to (1) keep track of what features they've customized, (2) organize them into logical groups, and (3) share those customizations with others.

Analyzing a course's customizations
Customizer provides users with the ability to quickly analyze an entire course and generate an overview of what settings have been modified from the defaults (Figure 5). The results are displayed both as a context-sensitive list (automatically filtered by which LMS pages are affected by each customization), or as a tree visualization showing the hierarchy of different LMS components and the changes made to them.

Annotating with explanations and additional metadata
In keeping with DG4, Customizer provides a way for customization authors to attach their own free-form explanations to each customization. This creates the opportunity to provide, for instance, one's rationale for making the change or any steps necessary to create or maintain the customization. Furthermore, this enables customization authors to reflect upon

the changes they have made to their courses and can serve as a future reminder of what steps were taken to arrive at their present setup or why they made certain choices. Customizer also keeps track of the most recent actions taken in Canvas and uses this information to generate template "how-to" instructions for customizations made by the user.

In addition to these explanations, two other types of metadata are attached to each customization: (1) the complexity/difficulty of a customization on a 5-point slider (selected by the user), and (2) information about which Canvas pages are affected by the customization (generated automatically).

Sharing customizations with other instructors
Users can share their customizations (or collections thereof) with their colleagues by clicking on the *Share* button attached to each individual customization or to an entire shared course. The user may select a subset of the customization cards shown after analyzing their setup if they only want to share a part of their setup but not the full package.

Implementation Details
We implemented Customizer as a Google Chrome browser extension, made to work on top of the Canvas LMS. We decided to use a browser extension over other potential implementations (such as the Learning Tools Interoperability standard [3]) to facilitate fine-grained manipulations of the HTML DOM for some of our features. It is likely that our design could be adapted to other formats with some engineering effort. Each course is represented by a collection of all of the settings and structural changes that have been made to it — essentially a "diff" between the current course and a freshly-created default one. The possible settings captured in this collection comprise nearly every built-in means of changing the UI or structure of a course, short of modifying the LMS source code directly. Customizer is also able to detect and install external plugins (Canvas "apps") that add new functionality to the application.

The exploratory *Try It* interface leverages several existing Canvas features for (1) creating new "sandbox" courses for testing, (2) importing/exporting course settings and content, and (3) resetting a course back to its defaults. The generated sandbox course is displayed within an HTML iframe inside the customization dialog (see Figure 4).

USABILITY STUDY
To evaluate the extent to which the design of Customizer meets instructors' needs, we ran a usability study with 10 additional educators who had recently taught a course using an LMS (none had participated in the formative interviews).

Methodology
During our study, we collected data through a demographic pre-test questionnaire, three observed usability tasks (described in detail below), a post-task usability questionnaire, and a brief follow-up interview. Participants were encouraged to think aloud while completing the tasks. Sessions were audio-recorded and transcribed for analysis.

In addition to some basic demographic information, the pre-questionnaire asked participants about their teaching experience and their past/current use of LMS or other similar tools.

The post-task usability questionnaire consisted of several 7-point Likert scale responses assessing the main features and goals of Customizer. Finally, the interview probed further into various threads such as how using Customizer compared to instructors' past course-creation experiences or their usual workflow, and how well the participant understood the "peek and try" conceptual model underlying the system.

This study protocol received ethics clearance at two universities where instructors were recruited. Participants were each given a $20 gift card in appreciation of their time.

Participants
Our 10 additional participants (3F, 7M) included a university professor, 3 post-doctoral researchers, and 6 teaching assistants who had all taught a university-level course within the past two years. All were between the ages of 19–40, with different levels of completed education (5 Bachelor's, 1 Master's, 4 PhD) and had experience teaching courses in subjects such as, Computer Science, Engineering, English, and Design. All but one of the participants had some prior experience with using Canvas, and five had also used another LMS.

Usability Tasks
Our study included three usability tasks designed to provide a broad coverage of all of Customizer's features. The first two tasks were made to be basic and advanced "consumer" tasks, respectively. Both asked participants to use the discovery and exploratory interfaces to look at other instructors' setups and decide how they should customize their own course. The third task was designed as a basic authoring task to familiarize the participant with viewing and sharing their own customizations. Each task was assigned a time limit (differing by task, see below), though if a participant reached this limit while still making progress and not overly frustrated, they were given two more minutes to try finishing the task. The content available in Customizer was bootstrapped by the research team (in consultation with other instructors) with six fictional customized setups, comprising 44 customizations (24 distinct, the rest overlapping between instructors), and ensuring that each part of the first two tasks had one or more customized setups that were related to it. Participants were also given a brief (2-3 minute) demo before the first task to give them some basic familiarity with the Customizer interface and its available features.

Task 1 (10 minutes) described a scenario in which the participant is setting up a course and needs to model its look and feel after other instructors in their department who have shared their setups through Customizer. This was an introductory task consisting of three simple subtasks: (1) enabling/disabling some optional discussion board features, (2) changing the course homepage to show additional information about upcoming assignments and announcements, and (3) choosing one of two optional course features that enable real-time collaboration with students. The goal for participants was to compare these different Canvas features based on viewing other instructors' setups and reading their explanations, before deciding which customizations to use themselves.

Task 2 (10 minutes) presented the participants with a screen-shot of two non-standard Canvas features (buttons for embedding YouTube and Wikipedia content in the page editor), and asked them to find a way to add those features to their setup. This functionality required the addition of external "apps" to the Canvas course, and would generally be challenging to figure out for a novice Canvas user. This task described a scenario where it is the middle of the school term and making untested changes to the live course setup could carry the risk of impacting students. We wanted to create an opportunity for participants to use the *Try It* feature and explore the customizations before importing them.

Task 3 (5 minutes) asked users to "pay it forward" by using Customizer to share one or more of their customizations from the prior tasks. The goal here was to encourage the participant to explore how the authoring interface works, so the task was left largely open-ended.

We iterated on the tasks several times based on pilot testing to strike a good balance between challenge, allotted time, and comprehensibility. Because participants' prior experience with Canvas varied, certain subtasks may have been more or less challenging for different participants. However, we designed all of the tasks to be sufficiently open-ended that they would elicit exploration and decision-making rather than relying solely on knowledge of the Canvas interface.

Data Analysis
We relied on regular discussions among the research team during data collection to identify key emerging themes from the task observations and interviews. After 10 participants, we had begun to see many recurring findings rather than new insights. We then used affinity diagramming to categorize our findings and bring the key themes into better focus.

Findings
Contrary to expectations, participants had more overall difficulty completing Task 1 (the "basic" task) than Task 2. Only 7/10 participants completed Task 1 within the allotted time, while all participants were successful at completing the other two tasks on time. This was likely at least partly due to learning effects, as the order of tasks was not randomized. Related, most participants' approaches to completing Task 2 were not qualitatively different from Task 1 — what would ordinarily be considered an "advanced" task (installing external plugins to add new course functionality) appeared to have been made conceptually simpler through Customizer's design.

From our questionnaire results, almost all participants found Customizer easy and enjoyable to use, with only one participant (P02) indicating that it was slightly confusing. A large majority of participants indicated that they would like to try Customizer with their real courses (9/10) and in other non-educational contexts (9/10). Every participant indicated that they would likely recommend Customizer to a friend or colleague, and 7/10 participants said they would rather build a course with Customizer than without it.

Through our observations and post-task interviews, we gained many deeper insights into participants' experiences with Customizer's in-context discovery interface, the exploratory mode,

and the built-in authoring system. The participants also shared their thoughts about ways Customizer could be useful in application domains other than LMSs.

In-context discovery of customizations is intuitive
Most participants were quick to rely on Customizer's in-context discovery mechanism, intentionally starting each task by navigating to a Canvas page they felt was the most likely location for it before opening the sidebar there. Notably, most participants seemed to use this contextual mechanism preferentially over the search bar for finding relevant customizations. This was particularly noticeable during the first task, where participants were seeking customizations to a few different settings in Canvas, and related customizations could be found easily in the sidebar on Canvas pages relevant to those settings (e.g., navigating to a discussion thread to find discussion-related customizations). P01 was especially happy that Customizer was fully built into the existing Canvas page:

It's good to have something that is built-in. Most of the time I actually need to Google, I need to figure out the underlying system... Or there's always like a help button, but it brings me to a separate page... So I like the fact that [Customizer] is built-in. I don't need to go anywhere else. (P01)

Viewing shared examples provides a confidence boost
Three participants mentioned that having access to shared example customizations helped them feel more confident about making similar changes to their own course:

If I see someone else took the time to set that up and post it [to Customizer], then it must work. So I feel like I have a little bit more confidence in that feature seeing that someone else has implemented it and wants to share it with other people. (P07)

In particular, seeing that multiple people had performed the same or similar change provided further confidence in that customization's usefulness:

I'll import the announcements as well because I think that's useful to have, and that was common between the three profs, so that's also an indication that it has been useful for more people...it gives me more confidence... Having the number of people that recommended that [customization], it did increase my confidence. (P08)

Similarly, some participants desired to see indicators of how popular certain customizations were among all other users, to get a better sense of what the most helpful customizations might be. Some also expressed wanting to dig further and see which customizations were the most widespread among other instructors teaching the same course subject as them.

However, two participants (P02, P03) were slightly concerned that Customizer might lower the barrier to customizing *too much*, thereby leading to instructors making changes without careful consideration of their consequences. Interestingly, part of P02's confusion had to do with the act of customization being oversimplified:

I prefer full control... but for me it was oversimplified. I usually feel that when things are oversimplified I lose my control... It's almost like drag-and-drop, but instead of dragging, you

just click on import...I kind of feel that the [customizations] provided by Customizer are just black boxes. You don't get any control for customizing them by your own needs. (P02)

How-to and rationale-based explanations are helpful
Most of the participants described finding it useful to see explanations attached to each customization: *"Reading Bob's explanations is really helpful in deciding what they [customizations] do" (P01).*

Interestingly, several participants drew a distinction between two different types of written explanations (which were not otherwise distinguished by Customizer): those which describe the author's *rationale* for a customization, and those which describe *how to use* it. Participants valued explanations of rationale for deciding whether a customization would match a desired use case, or to compare the benefits of different customizations, while how-to explanations were valued for details related to tweaking or reverting the changes.

People value the exploratory mode for many different reasons
In keeping with our original design goals, a large majority of participants found the *Try It* feature useful for risk-free exploration of customization options:

Without [Try It], it would be kind of stressful... Because there is this sort of worry of screwing up your course and because real students are using it. And you're like, "What if they're looking at this right now, and then I break something? ... Will I break this for people that are using this right now?" (P10)

In particular, multiple participants appreciated the *Try It* feature as it more fully gave them the *"other instructor's perspective" (P01)* with permission to make changes and experiment. Participants found this to be an advantage over current LMS features that allow instructors to share their setup with viewing privileges only.

More surprising was the number of different use cases that participants articulated, unprompted, for the *Try It* feature. For instance, some participants described how they were using it to confirm their intuition about what exactly a given customization does, especially when the functionality was not apparent from the customization's name or explanation:

Oh, here's one [customization]. I think this is what I want... Okay, from the description I'm not actually sure, so I'll Try It then. I think it's probably right because it showed YouTube... Ah yeah, there you go, so it's the right one to import. (P04)

Several other participants were using the exploratory mode as a visual comparison tool, to explore two or more instructors' course setups and spot any major differences:

This one is Bob's, I see there's an assignment on the home page. So I'm clicking the Assignments one first, just because I saw Alice's announcements... Oh, Alice doesn't have announcements, she had assignments only, so I think that I'll probably just import the two from Bob. (P06)

Some of these participants expressed a desire to see new features or extensions to the prototype that would make these comparisons even easier, such as the ability to view two *Try It* windows side-by-side, or to mix and match customizations

from multiple instructors in a single *Try It* environment. However, other participants wanted to perform these comparisons with something more akin to a list-style "diff" of two instructors' setups, rather than an interactive exploratory mode.

Beyond "peeking": trade-offs with over-the-shoulder learning
We asked participants to compare their experience with Customizer to other strategies they had used in the past to learn features of an LMS. Several participants described in-person demos as one of these strategies and raised the existence of a trade-off between the amount of time or coordination required and the richness of the interaction. Although over-the-shoulder demos allow for real-time, personalized Q&A sessions, it can be a hassle to find and coordinate mutual free time. Furthermore, P07 pointed out that even in an in-person scenario, there is little room to experiment with tweaks or explore new ideas because he wouldn't want to make accidental changes to other instructors' real courses any more than he would to his own:

If you had someone else and they just opened up their Canvas and they're like, "This is how we did it," you wouldn't have this environment where you could try something, or it might get published to a front-facing part of their Canvas... So, I think the sandbox feature with this [Customizer] is nice, because you have the freedom to really mess things up... then you can just exit out, and your actual courses are fine. (P07)

Sharing customizations can help oneself in the future
When using the authoring and sharing tools in Customizer (during the third task), 4/10 participants went beyond the task requirements by adding a written explanation and a complexity value to their customizations before sharing them. Several noted that in addition to helping other instructors, their annotated customizations would be especially valuable for *themselves* down the road, as a way to quickly recall what they had previously changed, and why or how.

I guess I would even use this as a reminder for myself, to collaborate with myself because I may not use it for only one course, right? [...] So I would say my own customization also adds value to myself in the future. I can reflect for myself about "Oh okay, this will be useful, I can do it differently if I know about this feature." (P04)

Customizer has potential to generalize beyond LMSs
From our post-task questionnaire, 9/10 participants agreed (5+ on a 7-point Likert scale) that they would like to use Customizer in other contexts outside of education. In the interviews, they described many different applications where Customizer could be useful. For instance, some of their examples included integrated development environments for programming, website builders, and image/video editing applications. Often they described how their typical use of these applications included tasks such as installing plugins from a public repository, navigating lists of shared templates, or working collaboratively with shared customized layouts.

DISCUSSION
We now step back to discuss some of the broader implications of this research and potential next steps.

Building in-application instructor communities

Many of our participants expressed a desire to see more community-like features within Customizer, such as user profiles and the ability to link different setups together (e.g., "See @Bob's setup for more similar ideas"). Others wanted the ability to ask questions of the customization author within the interface, or by allowing multiple instructors to explore and modify a shared course setup collaboratively. In particular, such Q&A and collaboration features could be a way to close the gap in interactivity between our design and in-person knowledge sharing, making it easier to get the benefits of over-the-shoulder interactions. Furthermore, the ability to collaboratively customize a course could be helpful in providing remote troubleshooting when things go wrong. At some schools and universities, there may also be many other people involved in course design and setup, such as instructional designers or learning experience designers. Customizer could serve as a platform to better connect these roles with instructors for communicating course standards and best practices.

Although we designed Customizer to accommodate one-on-one sharing between colleagues at the same school, we can see potential benefits to broadening the sharing mechanism to include larger communities of instructors. This could also be seen an extension of more general-purpose help-based communities (e.g., Super User [4], Social CheatSheet [40]) restricted to the domain of LMS customization for course instructors.

Improving relevance with personalized recommendations

Our participants appreciated the in-context retrieval of example customizations based on their navigation of the Canvas interface, and this idea could be further extended to provide more personalized recommendations of customizations. For instance, future extensions to this idea could use information about the course topic, instructor preferences, and frequently-combined customizations to further discern which customizations an instructor is likely to consider the most helpful.

Customizing for oneself vs. others

Although we focused on instructors' practices in customizing their LMSs, an important consideration here is that the instructors' goals for customization are not necessarily to improve *their own* productivity or efficiency, as is often the case for other kinds of software customizations (e.g., [28]). Rather, for LMSs and many other digital classroom tools [41], the end users targeted by these customizations are *students* whose user experience may be greatly affected by the changes made.

It is also the case in Canvas (as in many LMSs) that students have many customizable options of their own to change the features of their LMS interface. It could be fruitful to study how students perceive the course customizations their instructors work so hard to set up, and what might influence their own decision to customize (or not) within educational applications.

Incentives for sharing customizations

Our usability study of Customizer focused on understanding how instructors discover and explore potential customizations, with a smaller focus on the authoring and sharing mechanisms. It is well-known from past research that most people are far less willing to author and publish content than they are to consume it [36], which raises the question of whether a system like Customizer could be sustained by only a small number of content creators. However, past research has shown that communities of educators often rely primarily on a small handful of tech-savvy colleagues for much of their tech support and troubleshooting needs [41], who would perhaps be willing to take the lead on these authoring tasks. Not only would this be beneficial to all of the instructors who have access to their knowledge base of example courses, but it could also help to reduce some of the burden facing these "tech support hubs" by offloading some of their ongoing support roles to Customizer. Future work could investigate the possibility of providing additional incentives for instructors to share their customized setups as examples for others.

Example-based customization in other contexts

We heard many different possibilities from participants about how Customizer's reach could be extended to improve their workflows in applications other than their LMS. We believe that the general idea of discovering, exploring, and authoring customizations within an application UI can be extended to other applications with some engineering and design efforts. A key challenge would be enumerating and representing the varied possibilities for customizing different applications. Nonetheless, our studies suggest there could be substantial benefits in making customization examples more easily discoverable and explorable within the context of other applications.

Limitations

Our in-lab task-based usability study provided a useful window into how instructors perceive example-based customization sharing in limited settings, but it provides few insights into their real-world usage of such a system. To fully understand how it might change instructors' behaviour, Customizer needs to be explored through a larger-scale longitudinal deployment. Furthermore, our Customizer prototype was only implemented and tested on top of one widely-used LMS. While different LMSs share a core set of similar features (e.g., assignment submission, gradebook mechanisms, communication features, online quizzes), there may be large differences in the set of customizations available for altering those features and the degree of freedom afforded to instructors. Though the design is generalizable, some additional work would be needed to migrate the implementation beyond the Canvas environment.

CONCLUSION

In this paper, we have introduced the concept of in-context example-based customization sharing and designed the Customizer system to realize it within an LMS. With Customizer, instructors can quickly discover relevant customizations for their LMS and interactively explore them in the context of their own courses. Our studies with instructors have yielded important insights into the potential benefits of peeking at and experimenting with other instructors' LMS setups.

ACKNOWLEDGMENTS

REFERENCES

[1] 2019. Top Learning Management System Statistics For 2020. (Nov. 2019). `https://elearningindustry.com/top-learning-management-system-lms-statistics`

[2] 2020. Github repository: instructure/canvas-lms. (Jan. 2020). `https://github.com/instructure/canvas-lms`

[3] 2020. Learning Tools Interoperability. (2020). `https://www.imsglobal.org/activity/learning-tools-interoperability`

[4] 2020. Super User: Q&A for computer enthusiasts and power users. (2020). `https://superuser.com/`

[5] Erik van Alphen and Saskia Bakker. 2016. Lernanto: Using an Ambient Display During Differentiated Instruction. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA '16)*. Association for Computing Machinery, New York, NY, USA, 2334–2340. DOI: `http://dx.doi.org/10.1145/2851581.2892524`

[6] Pengcheng An, Saskia Bakker, Sara Ordanovski, Ruurd Taconis, Chris L. E. Paffen, and Berry Eggen. 2019. ClassBeacons: Enhancing Reflection-in-Action of Teachers through Spatially Distributed Ambient Information. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems (CHI EA '19)*. Association for Computing Machinery, New York, NY, USA, Article Paper INT002, 4 pages. DOI:`http://dx.doi.org/10.1145/3290607.3313239`

[7] Saskia Bakker, Elise van den Hoven, Berry Eggen, and Kees Overbeeke. 2012. Exploring Peripheral Interaction Design for Primary School Teachers. In *Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction (TEI '12)*. Association for Computing Machinery, New York, NY, USA, 245–252. DOI: `http://dx.doi.org/10.1145/2148131.2148184`

[8] Grégory Bourguin, Arnaud Lewandowski, and Myriam Lewkowicz. 2012. ShareXP: An Eclipse plug-in for expertise sharing among developers.

[9] Joel Brandt, Mira Dontcheva, Marcos Weskamp, and Scott R. Klemmer. 2010. Example-Centric Programming: Integrating Web Search into the Development Environment. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10)*. Association for Computing Machinery, New York, NY, USA, 513–522. DOI: `http://dx.doi.org/10.1145/1753326.1753402`

[10] Peter Brusilovsky and Michael V Yudelson. 2008. From WebEx to NavEx: Interactive access to annotated program examples. *Proc. IEEE* 96, 6 (2008), 990–999.

[11] Andrea Bunt. 2007. *Mixed-initiative support for customizing graphical user interfaces*. Ph.D. Dissertation. University of British Columbia.

[12] Andrea Bunt, Joanna McGrenere, and Cristina Conati. 2007. Understanding the utility of rationale in a mixed-initiative system for GUI customization. In *International Conference on User Modeling*. Springer, 147–156.

[13] Parmit K. Chilana, Andrew J. Ko, and Jacob O. Wobbrock. 2012. LemonAid: Selection-Based Crowdsourced Contextual Help for Web Applications. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. Association for Computing Machinery, New York, NY, USA, 1549–1558. DOI: `http://dx.doi.org/10.1145/2207676.2208620`

[14] Reports and Data. 2019. Learning Management System (LMS) Market Analysis, By Component Type (Solution, Services), By Delivery Mode (Distance Learning, Instructor Led, Blended), Deployment Type, By End User (Academic and Corporate), and By Region, Forecasts to 2026. (Nov. 2019). `https://www.reportsanddata.com/report-detail/learning-management-system-lms-market`

[15] Peggy A. Ertmer and Anne T. Ottenbreit-Leftwich. 2010. Teacher Technology Change: How Knowledge, Confidence, Beliefs, and Culture Intersect. *Journal of Research on Technology in Education* 42, 3 (March 2010), 255–284. DOI: `http://dx.doi.org/10.1080/15391523.2010.10782551`

[16] Leah Findlater and Joanna McGrenere. 2004. A comparison of static, adaptive, and adaptable menus. *Conference on Human Factors in Computing Systems - Proceedings* (04 2004), 89–96. DOI: `http://dx.doi.org/10.1145/985692.985704`

[17] C. Ailie Fraser, Mira Dontcheva, Holger Winnemöller, Sheryl Ehrlich, and Scott Klemmer. 2016. DiscoverySpace: Suggesting Actions in Complex Software. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems (DIS '16)*. Association for Computing Machinery, New York, NY, USA, 1221–1232. DOI: `http://dx.doi.org/10.1145/2901790.2901849`

[18] Cynthia Gautreau. 2011. Motivational factors affecting the integration of a learning management system by faculty. *Journal of Educators Online* 8, 1 (2011), n1.

[19] Nicolas Guibert, Patrick Girard, and Laurent Guittet. 2004. Example-Based Programming: A Pertinent Visual Approach for Learning to Program. In *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI '04)*. Association for Computing Machinery, New York, NY, USA, 358–361. DOI: `http://dx.doi.org/10.1145/989863.989924`

[20] Mona Haraty and Joanna McGrenere. 2016. Designing for Advanced Personalization in Personal Task Management. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems (DIS '16)*. Association for Computing Machinery, New York, NY, USA, 239–250. DOI: `http://dx.doi.org/10.1145/2901790.2901805`

[21] Mona Haraty, Joanna McGrenere, and Andrea Bunt. 2017. Online Customization Sharing Ecosystems: Components, Roles, and Motivations. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing (CSCW '17)*. Association for Computing Machinery, New York, NY, USA, 2359–2371. DOI: `http://dx.doi.org/10.1145/2998181.2998289`

[22] Björn Hartmann, Leslie Wu, Kevin Collins, and Scott R. Klemmer. 2007. Programming by a Sample: Rapidly Creating Web Applications with d.Mix. In *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology (UIST '07)*. Association for Computing Machinery, New York, NY, USA, 241–250. DOI:`http://dx.doi.org/10.1145/1294211.1294254`

[23] Instructure Inc. 2020. Canvas Guides | Canvas LMS Community. (2020). `https://community.canvaslms.com/community/answers/guides/canvas-guide`

[24] Alyssa Lai and Philip Savage. 2013. Learning management systems and principles of good teaching: Instructor and student perspectives. *Canadian Journal of Learning and Technology/La revue canadienne de l'apprentissage et de la technologie* 39, 3 (2013).

[25] Rock Leung, Charlotte Tang, Shathel Haddad, Joanna Mcgrenere, Peter Graf, and Vilia Ingriany. 2012. How Older Adults Learn to Use Mobile Devices: Survey and Field Investigations. *ACM Trans. Access. Comput.* 4, 3, Article Article 11 (Dec. 2012), 33 pages. DOI: `http://dx.doi.org/10.1145/2399193.2399195`

[26] Michael Xieyang Liu, Jane Hsieh, Nathan Hahn, Angelina Zhou, Emily Deng, Shaun Burley, Cynthia Taylor, Aniket Kittur, and Brad A. Myers. 2019. Unakite: Scaffolding Developers' Decision-Making Using the Web. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology (UIST '19)*. Association for Computing Machinery, New York, NY, USA, 67–80. DOI: `http://dx.doi.org/10.1145/3332165.3347908`

[27] Wendy E. Mackay. 1990. Patterns of Sharing Customizable Software. In *Proceedings of the 1990 ACM Conference on Computer-Supported Cooperative Work (CSCW '90)*. Association for Computing Machinery, New York, NY, USA, 209–221. DOI: `http://dx.doi.org/10.1145/99332.99356`

[28] Wendy E. Mackay. 1991. Triggers and Barriers to Customizing Software. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '91)*. Association for Computing Machinery, New York, NY, USA, 153–160. DOI: `http://dx.doi.org/10.1145/108844.108867`

[29] Allan MacLean, Kathleen Carter, Lennart Lövstrand, and Thomas Moran. 1990. User-Tailorable Systems: Pressing the Issues with Buttons. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '90)*. Association for Computing Machinery, New York, NY, USA, 175–182. DOI: `http://dx.doi.org/10.1145/97243.97271`

[30] Justin Matejka, Tovi Grossman, and George Fitzmaurice. 2011. IP-QAT: In-Product Questions, Answers, & Tips. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology (UIST '11)*. Association for Computing Machinery, New York, NY, USA, 175–184. DOI: `http://dx.doi.org/10.1145/2047196.2047218`

[31] Justin Matejka, Wei Li, Tovi Grossman, and George Fitzmaurice. 2009. CommunityCommands: Command Recommendations for Software Applications. In *Proceedings of the 22nd Annual ACM Symposium on User Interface Software and Technology (UIST '09)*. Association for Computing Machinery, New York, NY, USA, 193–202. DOI: `http://dx.doi.org/10.1145/1622176.1622214`

[32] Joanna McGrenere, Ronald Baecker, and Kellogg Booth. 2002a. *The design and evaluation of multiple interfaces: A solution for complex software*. University of Toronto Canada.

[33] Joanna McGrenere, Ronald M. Baecker, and Kellogg S. Booth. 2002b. An Evaluation of a Multiple Interface Design Solution for Bloated Software. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '02)*. Association for Computing Machinery, New York, NY, USA, 164–170. DOI:`http://dx.doi.org/10.1145/503376.503406`

[34] L. R. Neal. 1989. A System for Example-Based Programming. *SIGCHI Bull.* 20, SI (March 1989), 63–68. DOI:`http://dx.doi.org/10.1145/67450.67464`

[35] I. B. L. News. 2020. Canvas LMS Increases Its Lead to 30 Million Users, According to Its CEO's Data | IBL News. (2020). `https://iblnews.org/canvas-lms-increases-its-lead-to-30-million-users/`

[36] Jenny Preece, Blair Nonnecke, and Dorine Andrews. 2004. The top five reasons for lurking: improving community experiences for everyone. *Computers in Human Behavior* 20, 2 (2004), 201 – 223. DOI: `http://dx.doi.org/https://doi.org/10.1016/j.chb.2003.10.015` The Compass of Human-Computer Interaction.

[37] Susan M. Printy. 2008. Leadership for Teacher Learning: A Community of Practice Perspective. *Educational Administration Quarterly* 44, 2 (April 2008), 187–226. DOI:`http://dx.doi.org/10.1177/0013161X07312958`

[38] Chris Shelton. 2017. Giving up technology and social media: why university lecturers stop using technology in teaching. *Technology, Pedagogy and Education* 26, 3 (2017), 303–321. DOI: `http://dx.doi.org/10.1080/1475939X.2016.1217269`

[39] Michael B. Twidale. 2005. Over the Shoulder Learning: Supporting Brief Informal Learning. *Comput. Supported Coop. Work* 14, 6 (Dec. 2005), 505–547. DOI: `http://dx.doi.org/10.1007/s10606-005-9007-7`

[40] Laton Vermette, Shruti Dembla, April Y. Wang, Joanna McGrenere, and Parmit K. Chilana. 2017. Social CheatSheet: An Interactive Community-Curated Information Overlay for Web Applications. *Proc. ACM Hum.-Comput. Interact.* 1, CSCW, Article 102 (Dec. 2017), 19 pages. DOI: `http://dx.doi.org/10.1145/3134737`

[41] Laton Vermette, Joanna McGrenere, Colin Birge, Adam Kelly, and Parmit K. Chilana. 2019. Freedom to Personalize My Digital Classroom: Understanding Teachers' Practices and Motivations. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. Association for Computing Machinery, New York, NY, USA, Article 318, 14 pages. DOI: `http://dx.doi.org/10.1145/3290605.3300548`

[42] Etienne Wenger. 1999. *Communities of practice: Learning, meaning, and identity*. Cambridge university press.