Teaching and Learning Software in Landscape Architecture: A Survey of Software Use Amongst Faculty and Students

Benjamin H. George[1], Peter Summerlin[2], Taz Fulford[2]

[1]Utah State University, Utah/USA · benjamin.george@usu.edu
[2]Mississippi State University, Mississippi/USA

Abstract: Competency in computer software is integral to the profession of landscape architecture, replacing many of the traditional drawing and drafting tools used. This article seeks to capture the current state of software in academia from the perspective of both faculty and students, and to determine if students are achieving the learning levels that faculty are targeting in their courses. Results show that students are attaining faculty's learning targets, yet students report discontent with the learning process, suggesting that more emphasis is needed to help students understand the role of software in the design process.

Keywords: Education, software, CAD, GIS, drafting, rendering, 3D modeling

1 Introduction

At the core of landscape architectural design education is the conceptual understanding and application of the design process. This conversation often permeates multiple parts of a landscape architecture academic curriculum, but the application is most apparent in the design studio, where students are executing and refining their personal process. Historically, this process was entirely analog and, fittingly, academic courses integrated instruction for using traditional analog tools. In recent decades, digital tools (primarily computer software) have supplemented or, in some cases replaced, many of the traditional drawing and drafting tools. The motivations for this technology integration vary. Often it is about efficiency, speed, and precision in design. In other instances, digital technology has made possible an increased complexity of expression. Technology integration has created a significant expansion of the potential tools in the design process, and the profession has passed a threshold to where it can no longer be reasonably expected for any one person to be proficient in all available tools. Instead, academics and professionals alike must decide which tools are most appropriate in their personal design process while remaining open to emerging and evolving tools.

For academics, the decision of which tools to integrate into the design process is extended to their students in how they build curriculum and allocate departmental resources. For digital software specifically, there are multiple influences that might dictate which software is emphasized. One likely influence is the consensus software used in professional practice, and existing surveys document this information (CALABRIA 2012, CALABRIA 2016). These surveys indicate that the most commonly used software programs are AutoCAD, Adobe Photoshop, SketchUp, and PowerPoint. The same surveys also indicated that the primary computer skills desired by firms that are hiring were digital drafting skills. Communication skills, assumed to be plan, section, and perspective renderings, were the second highest skills desired by hiring firms. Other influences that affect computer software decisions in academia

Journal of Digital Landscape Architecture, 4-2019, pp. 354-362. © Wichmann Verlag, VDE VERLAG GMBH · Berlin · Offenbach. ISBN 978-3-87907-663-5, ISSN 2367-4253, e-ISSN 2511-624X, doi:10.14627/537663038.

might be the training and proficiency of the instructor as well as faculty predictions of emerging software and trends. At the same time, students are making their own decisions about software as they expand their digital knowledge outside of the classroom.

As the available software tools continue to grow, it will be important to capture current usage data across the profession to make informed decisions in teaching and practice. This article seeks to contribute to that knowledge by analyzing survey results of software usage in academia from the perspective of both faculty and students. The paper will discuss the results of the surveys across a range of software categories to evaluate the current state of software use.

2 Background and Literature Review

The integration of computers and software in the design curriculum has seen steady progression. This technology was gradually tested and integrated at only a few academic institutions beginning in the 1960s. Only within the past two decades has design software become widely embedded in the design studio. The earliest integration was the development of SketchPad by Ivan Sutherland at MIT in the early 1960s, along with the technology behind Geographic Information Systems (GIS) at the Laboratory for Computer Graphics at the Harvard Graduate School of Design (ERVIN 1999). Although there were many predecessors, digital technology and software really emerged as a critical tool for the design professions in 1982 when AutoDesk released its computer-aided drafting software: AutoCAD (WEISBERG 2008).

The Landscape Architecture Accreditation Board conducted a survey of accredited landscape architecture programs in 1994 about their use of computer hardware (PALMER 1994). This survey found that students were the driving force behind computer adoption trends. Additionally, discussion amongst faculty revealed concerns that teaching computer software would consume significant curriculum time and distract students from mastering basic design concepts (PALMER 1994). However, by the turn of the century the adoption of software in practice had reached a level of importance that students without technical software skills would find it difficult to find employment upon graduation (TAI 2003).

Today, driven by the decreasing cost of computers and software, paired with the simultaneous increase in computer power and capability, the integration of technology continues to expand. Software is now only a segment of the technology used, with an array of digital tools like 3D printing and 3D scanning, computer numerical control (CNC) milling machines, unmanned aerial vehicles (UAV), virtual reality, augmented reality, and programming or coding becoming prevalent (GEORGE et al. 2018). As digital technology has become essential in the workflow of the contemporary landscape architect, it has become increasingly important to those developing landscape architecture curricula. The use of computers is ubiquitous in design curriculum and nearly every accredited landscape architecture program now requires students to own a computer (SUMMERLIN et al. 2017). In conjunction with a growing number of software programs, mobile applications already exist to aid in site analysis, design, construction and presentation (CAMUTI 2015). Where before faculty grappled with whether they should be teaching computer software in their curriculum, today the questions revolve around which software, of the many options, should they be teaching. It can be expected that these questions will only increase as mobile applications become more prevalent.

3 Methodology

3.1 Faculty Study

This study utilized two separate surveys. The first survey was sent to faculty, and the second to students, at all of the accredited landscape architecture programs in the United States. The faculty survey was used to assess the software currently taught. In the survey, software was categorized by function-groups: drafting, 2D rendering, 3D modeling, 3D modeling plugins, and desktop publishing. Respondents used a rating scale (see Figure 1) to indicate the learning goals of student proficiency for each software.

To determine which software to include in the survey, the researchers first identified the software taught within their own departments. The researchers then contacted peers at other programs to further expand the list. Finally, the researchers reviewed all recent (six months) entry-level landscape architect job postings on the ASLA JobLink website to identify software sought by employers. Respondents also had the ability to manually enter and rate additional software not listed in the survey.

Researchers solicited responses from all sixty-nine Landscape Architectural Accreditation Board (LAAB) accredited or candidacy programs in the United States. To limit response bias from faculty not familiar with the software, RICHIE & LEWIS' (2003) purposive sampling method was utilized, which has been effectively used in a similar study in landscape architecture (LI, YANG & YAN 2014). To identify participants, the researchers evaluated teaching assignments and stated research interests of the faculty members at each university. Individuals with a teaching assignment or research interest primarily related to software were invited to participate. When at least one faculty member was not identified at a university, the department head was contacted to assist in selecting a participant.

3.2 Student Study

Students were similarly asked to use a rating scale (see Figure 1) to rate their perceived level of proficiency with individual software programs, as well as provide feedback on their software learning experience. Students were asked about the same software as faculty, as well as additional software that had been submitted during the faculty survey, and the software was similarly categorized. Students were also invited to submit and rate software not found on the survey. In addition to proficiency, students were also asked about their attitudes towards their academic training regarding software, the resources they use, and the role they believe design software should play in their education. Qualitative responses to these questions were coded to identify prominent themes in the student responses. The student survey also gathered demographic data about students' age, gender, year in school, and computer use. This data was used to identify correlations with perceived proficiency.

The student survey was distributed to department heads, who were asked to forward it to their students. To maximize distribution, when contact information was available, the survey was sent to the student chapters of the American Society of Landscape Architects to distribute.

4 Results

Twenty-seven faculty responded to the survey request, representing 39% of the total LAAB accredited or candidacy programs in the United States. The highest-rated software is Photoshop (3.64), AutoCAD (3.52), InDesign (3.4), Illustrator (2.62), SketchUp (2.58), ArcGIS (2.46), Rhino (1.6), and Grasshopper for Rhino (1.12) (see Figure 1). Each of these programs has an average score above 1, which indicates that students are at least introduced to the software. In questions regarding the approach to teaching software, 65% of respondents reported that students are taught through specific software courses, 72% reported that they have a computer lab available for students to use, and 51% report that they support both Mac and Windows operating systems, with 49% supporting only Windows.

On the student survey, 214 students from 29 universities responded. The results found that graduating students' (n= 114) perception of their competency exceeded faculty's target level of competency for most software programs ($r(22) = .887$, $p = <.001$). Students especially exceed faculty expectation on software that is relatively new or niche in application, such as Lumion or Land F/X (see Figure 1). Despite good student performance, 48% of students believe faculty were insufficiently prepared to teach current software, and student comments reveal broad dissatisfaction amongst students who believe that they must rely primarily upon self-teaching and online resources. Several important correlations exist within the student responses, including gender, operating system, and attitude. Male students report higher levels of proficiency ($r(22) = .994$, $p = <.001$), students who use PCs report higher levels of proficiency ($r(22) = .972$, $p = <.001$), and students who report that they receive sufficient training from their academic institution report higher proficiency ($r(22) = .974$, $p = <.001$).

Students who responded that they did not feel faculty were adequately prepared to teach the software were asked the follow-up question: "What role should computers and design software play in landscape architecture education?" were *emphasized critical importance* (108), *communication tool* (40), and *as stand-alone courses* (21). The top responses to the question "Why are faculty not prepared to teach software?" The highest coded response for this question is *faculty unfamiliar with software* (60) and second-highest is *curriculum time constraints* (24). The list of coded responses are shown in Table 1.

Students reported that fellow students (5.32) were their most important learning resource, followed by video tutorials (5.23), website tutorials (4.64), professors (4.28), work experience (3.65), professional workshops (2.55), and textbooks or manuals (2.33). Two sources were clear student favourites to find tutorials: YouTube (156) and Lynda (59).

Table 1: Coded responses from student survey

What role should computers and design software play in landscape architecture education?		*Why are faculty not prepared to teach software?*	
Code	**Count**	**Code**	**Count**
Emphasize critical importance	108	Faculty unfamiliar with software	60
Communication tool	40	Curriculum constraints	24
Taught in stand-alone course	21	Faculty expect students to self-teach	23
Valuable, but not critical	14	Faculty are from different paradigm	15
Only a tool	13	Pedagogical issues	6
Connected to certain design stages	13	Faculty have not worked in practice	4
Provides efficiency/accuracy	12		

5 Discussion

5.1 Student Attitudes

The results of the survey suggest that students are achieving the proficiency level that faculty have targeted by the time they are graduating. While this is positive from a learning objectives standpoint, half of students were dissatisfied with their software learning experience. Students who responded negatively about their learning experience overall, and their perception of faculty's software ability, scored consistently lower across the software categories. Even among students who reported higher satisfaction with their instruction, a common complaint was that students had to frequently self-teach. Students who felt that they were receiving insufficient training in software were blunt in their assessments, commenting that faculty "do not know the programs sufficient to teach them" and that faculty "don't know which software are commonly used in offices." It is clear that students prefer that faculty members teaching software courses be proficient in the software and not rely upon third party materials, such as textbooks, teaching assistants, and tutorials to convey the content.

The findings of the survey make it clear that students place great emphasis on developing software skills. When students were asked if they believe software proficiency will help them get a job, the overwhelming majority responded that it was very important (141 of 214 responses), with a mean score of 4.61 (out of 5). It is not surprising that students feel this way. Many entry-level jobs emphasize software proficiency in job announcements and expect young designers to mature in their design capabilities while on the job. Therefore students, with some justification, feel their software skills are critical to getting a job early in their career. One student stated that the "expectation of all jobs is for candidates to have design software skills". Another student said that learning software "should be a priority as firms expect millennials to know software," and another noted that "from the people in firms that I have talked to, they expect students to be bringing the best and newest forms of technology".

5.2 Recommendations for Instruction

While the survey reveals discontent amongst students about their software training, student comments provide some insights into changes that departments might consider. Students repeatedly commented that the teaching assistants are more knowledgeable than the instructor regarding software use, and fellow students are also rated as their top learning source. Departments could formalize this informal learning network in two ways. First, departments could help organize student-led software training outside of class period. Several students commented on their departments using such a model successfully. Second, where appropriate, students could act as instructors in technology training courses, which could also help free faculty's time. Departments should also ensure that faculty are given the time and financial resources needed to seek additional training in software to stay more current on trends, if faculty are so inclined.

Knowing the weight that software carries with students, faculty should ensure that students recognize the proper role of software as a tool to design – not as a replacement for design skill – and that the students not become fixated on software and technology. It is critical that students recognize software as tools, for by its very nature software is constantly changing. In the book Digital Drawing for Landscape Architecture, Landscape Architect Ken Smith states:

Today, my office uses an array of representation techniques ranging from drawing to physical model building to digital modeling, and all sorts of combinations of digital imaging and animations... Ultimately, the best design still results from thinking, designing and representing with multiple scales, views and methods. Just as it was impossible to practice 25 years ago without knowledge of ozalid printing, letraset, zipatone and rapidiograph use, today it is unimaginable to practice in a world without Photoshop, Illustrator, 3D Studio Max, Rhino, SketchUp and CAD (CANTRELL & MICHAELS 2015).

Students who fail to recognize that software is a tool with which to realize their design ideas will invariably find themselves lacking in advanced design skills. Indicative of this problem is that many students were critical of senior faculty because of a perceived lack of software skills, describing them as holding "outdated beliefs" and not "advancing digital skills". These faculty, however, have a wealth of knowledge regarding design process and other skills and possibly offer a broader perspective about the fluidity of technology in the profession. However, another student astutely observed that faculty are "focused on teaching design principles, not technology".

Several options might be considered to encourage students to avoid developing a software fixation. Suggested strategies might include having assignments place a greater focus on design process, rather than final outputs. Such projects can help students focus on design skills, instead of producing polished outputs which lend themselves to heavy software use (DUTTON 1987). This can help students be grounded in the design thinking process without quickly jumping to output production. Emphasizing multiple methods of production could help students to recognize software as tools, and that knowing when and how to use a tool is an equally important part of their workflow. This can be done using a mixture of digital and analog tools, but also by using a variety of software options that introduce students to software beyond what is familiar. Additionally, departments should encourage student learning towards emerging software and technology trends and provide opportunities for students to explore these areas. Some of these trends include the use of coding and scripting tools, BIM, parametric modelling, the creation, management, and use of datasets, gaming engines, immersive visualization tools (VR, AR, and MR) and novel integrations of existing tools. With the immense amount of new technological innovation occurring, it will likely become more common that landscape architects develop technological specializations. In many ways, students learning how to self-teach themselves software and other technologies is a very valuable skill that will make them more attractive to future employers (SANDERS 2018).

5.3 Performance Variables

Two correlations warrant further discussion. First, gender emerged as a significant correlation with reported proficiency, and female students report markedly lower levels of software proficiency. Males rate their overall proficiency to be 3.50 (out of 5), while females rate it at 3.01. Without performing a skills-test it is not possible to determine how accurate these perceptions are, however it is clear that females perceive themselves to be less skilled with software and at a disadvantage in this area. This may put extra stress on female students, especially because they believe proficiency is important to their job prospects at a higher level (1.58 out of 2) than males (1.42), and therefore labour under the double burden of perceiving their skills to be lower while believing their skills must be higher to obtain employment. Providing female mentors who are software or technology experts could be an effective way to boost female students' confidence. These mentors might be faculty, peers, practitioners, or even online tutorial hosts.

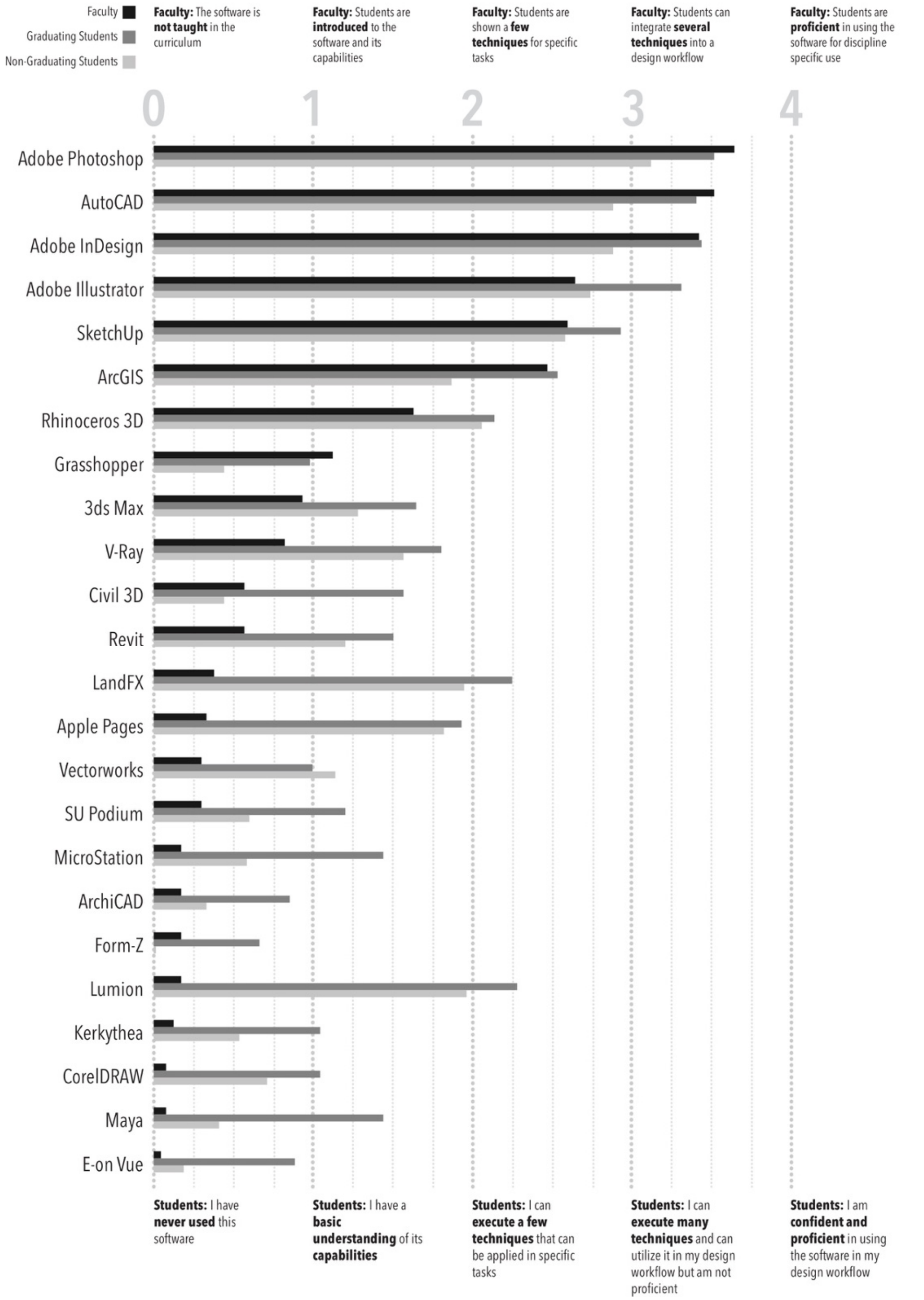

Fig. 1: Comparison of survey responses from faculty, graduating students, and all students

Finally, the operating system used by students has an impact on students' ability to master software. This is not surprising, as some software is only available for a PC, and therefore Mac users are disadvantaged by either not having access to the software or having to run it using a virtual operating system. However, this cannot explain the total difference in scores, as even on software that runs natively on both Mac and PC, students using PCs reported higher levels of proficiency. Because students responded that they frequently use online tutorials, it is possible that there are fewer high-quality software tutorials for Mac users and using the PC-specific tutorials might slow their learning.

6 Conclusions and Future Research

It should be noted that the primary constraint of this survey was the reliance upon perception of skills attainment in both the faculty and student surveys. Attempts were made to mitigate for this by using a rating scale with explicit definitions of skill levels (see Figure 1). A more accurate assessment would have been to administer skills-tests, however this was not feasible to do on a national scale.

Overall, the data suggests that student learning outcomes and faculty expectations are well aligned. At the same time, the results of the survey include several concerning trends, and suggest that departments and schools should assess how they provide software instruction and what role software and technology should play in curriculums in the future. The discussion should be wide ranging, ensuring to include current trends and predict future ones. For instance, with the advances being made in mobile computing, mobile software will become more prevalent in workflows and may supplant traditional desktop software for some tasks. Faculty need to consider how these mobile technologies should be integrated into design and technology curricular strategies. How well the results match practitioner expectations of graduating students, as well as similar academic surveys in regions beyond the United States, is an area that would benefit from future research. These existing and future studies all have potential to aid academics and students alike as they evolve their curriculum objectives for teaching software. Additionally, a common theme from students is that faculty can be better prepared to teach technology and software. How to provide faculty with additional training, and if it is a reasonable expectation, is a discussion that could happen within each department.

References

CALABRIA, A. (2012), Tech Trends. Landscape Architect and Specifier News, 28 (1), 16-17.

CALABRIA, A. (2016), Tech Trends. Landscape Architect and Specifier News, 32 (4), 14.

CAMUTI, LIZ (2015, AUGUST 11), Smartphone Apps for Landscape Architects: Useful Tools for Site Analysis and Design. THE DIRT. https://dirt.asla.org/2015/08/11/smartphone-apps-for-landscape-architects-analysis-and-design-part-1/ (February 28, 2019).

CANTRELL, B. & MICHAELS, W. (2015), Digital Drawing for Landscape Architecture: Contemporary Techniques and Tools for Digital Representation in Site Design. 2nd Edition. Wiley, Hoboken, NJ.

DUTTON, T. A. (1987), Design and studio pedagogy. Journal of architectural education, 41 (1), 16-25.

ERVIN, S. M. & HASBROUCK, H. (1999), Thirty Years of Computing in Landscape Architecture. Landscape Architecture, 89 (11), 54-56.

GEORGE, B. H., SUMMERLIN, P. & FULFORD, T. (2018), Digital Technology Professional Practice Network Meeting. Annual Meeting of the American Society of Landscape Architects. Philadelphia, PA.

PALMER, J. F. (1994), Use of Computers in Landscape Architecture Programs. ResearchGate. https://www.researchgate.net/publication/280836680_Use_of_Computers_in_Landscape_Architecture_Programs (September 4, 2017).

SANDERS, M. L. (2012), Becoming a learner. Institution for Communication & Leadership, Logan, UT.

SUMMERLIN, P., GEORGE, B. H. & FULFORD, T. (2017), Which software are you teaching? A survey of design software usage in landscape architecture curricula. The International Journal of Architectonic, Spatial, and Environmental Design, 11 (4).

TAI, L. (2003), Assessing the Impact of Computer Use on Landscape Architecture Professional Practice: Efficiency, Effectiveness, and Design Creativity. Landscape Journal, 22 (2), 113-125.

WEISBERG, D. E. (2008), The engineering design revolution: the people, companies and computer systems that changed forever the practice of engineering. Cyon Research Corporation. http://www.cadhistory.net.