Between design and digital: bridging the gaps in architectural education

Shelby Doyle
Iowa State University, doyle@iastate.edu

Nick Senske
Iowa State University, nsenske@iastate.edu

Follow this and additional works at: https://lib.dr.iastate.edu/arch_conf

Part of the Architecture Commons, and the Scholarship of Teaching and Learning Commons

Recommended Citation
https://lib.dr.iastate.edu/arch_conf/83

This Conference Proceeding is brought to you for free and open access by the Architecture at Iowa State University Digital Repository. It has been accepted for inclusion in Architecture Conference Proceedings and Presentations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.
Between design and digital: bridging the gaps in architectural education

Abstract
Developing technologies, such as computational design and digital fabrication, are transforming the design and construction of contemporary architecture. Today, architecture schools are tasked with introducing digital technologies as they are changing, creating an opportunity to develop innovative curricula and democratize access to these skills. However, the understanding of how to teach digital technology as an essential design skill has not kept pace with these rapid changes. Design education and digital technology education continue to be seen as separate loci of learning, separated by pedagogical gaps and teaching mindsets.

Disciplines
Architecture | Scholarship of Teaching and Learning

Comments
This proceeding is from Research Based Education (London: Bartlett School of Architecture, UCL, 2016).
Between design and digital: bridging the gaps in architectural education

Shelby Doyle & Nick Senske
Iowa State University, USA

INTRODUCTION

Developing technologies, such as computational design and digital fabrication, are transforming the design and construction of contemporary architecture. Today, architecture schools are tasked with introducing digital technologies as they are changing, creating an opportunity to develop innovative curricula and democratize access to these skills. However, the understanding of how to teach digital technology as an essential design skill has not kept pace with these rapid changes. Design education and digital technology education continue to be seen as separate loci of learning, separated by pedagogical gaps and teaching mindsets.

The aims of this paper are to take control of the pedagogical agenda for digital design in architectural education by debunking the myth of the digital native and to apply proven educational research to the pursuit of digital design. Two pedagogical proposals are put forward: learning objectives and soft skills for digital design in architecture. To be clear, this paper is a discussion of architecture, design, and education; not an argument for software and computer use in design. The relevance of this educational conversation extends only so far as it impacts the development of the profession’s relationship to digital technologies as these technologies are changing. The goal of this, and any, educational proposal for architecture must be improving the state of architectural design in addition to advancing learning in both the academy and the profession.

Much of architectural education today is what Bruner calls “folk pedagogy”, guided by implicit assumptions but not connected with educational theory or evidence beyond one’s experiences (Bruner, 1996). This places the architectural discipline in an unfortunate position where it neither benefits from nor makes contributions to the larger conversations occurring in educational research. In the past three decades, advances in cognitive learning theory and psychology, supported by empirical evidence collected from rigorous classroom assessment, have brought science into the art of teaching. This paper applies principles from educational research to improve digital design instruction by bridging the gaps between studio learning and technology (digital) learning. The first section of this paper describes learning objectives and Bloom’s
Taxonomy as tools of educational research designed to create clarity, transparency, and accountability among educators. Articulating learning objectives that are specific to digital design in architecture frames a conversation as to why there is such inconsistency and disagreement about the requirements of digital education across architectural curricula. The use of learning objectives may seem obvious or unnecessary if one is only considering their use in one’s own syllabi, but in terms of disciplinary alignment, digital design instruction could benefit from the additional clarity offered.

The second section of this paper describes a list of soft skills that support students as they learn digital design followed by several methods for integrating soft skills into digital design instruction. Soft skills are “soft” in contrast to more easily quantifiable “hard” skills such as operating a machine or knowledge of art history. Failure to acquire soft skills such as resourcefulness, good electronic communication etc. negatively impacts how technology is introduced, practiced, and developed in architectural studio culture. With the rapid pace of technological change, students need to be comfortable with and capable of learning, relearning, and integrating new programs and tools throughout their career. Soft skills provide a framework for helping students develop this mindset and facility.

BETWEEN DESIGN AND DIGITAL

Computer-Aided Drafting and Design (CADD) technologies have become commonplace in architectural practice as tools of efficiency and production. For these very reasons the introduction of CADD in early architectural curricula has been fraught with anxieties along a continuum: from the undoing of creativity through positivist and reductionist logic (Pullasmaa, 1996) to a firm belief that these technologies will revolutionize the way architects practice and think about design (Kieran and Timberlake, 2003). At the same time, there is a presumption that students who have grown up with digital technologies are “digital natives” who possess special aptitudes or insights which are disruptive to learning computing. The presence of these anxieties and biases often leads to gaps in architectural pedagogy, as digital tools are misunderstood and misappropriated by students and teachers alike.

Digital design is a term in common use, however its definition is unclear. One the one hand, there is very little architectural work today which does not use the computer in some capacity, and yet there are also designs which consciously engage in digital aesthetics and processes. The latter is obviously digital in aesthetic, but the former could still be considered digital by method. The very existence of the category of digital design is problematic because it implies two cultural silos in architecture: those who are digital and those who are not. This outlook potentially limits students’ educational and professional development.

Design is the verb in architectural education and in architecture; it is what architects do. For the purposes of this paper, digital design refers to the use of the computer and computer-driven tools (such as CNC machines, robots, etc.) when one designs architecture. The key is not what a person designs, rather whether that person employs the computer or not as a tool in architectural work. This paper interprets digital design as a broad skillset that should be available to all students, rather than
a niche specialization.

It is necessary to create the distinction between design and digital design— and to speak of teaching digital design—in this moment, because the introduction of the computer in architecture changes both what and how architects design. It introduces both new capabilities and new sources of bias and error. Therefore, it is necessary for architectural education to address and teach specific ways of designing with the computer— not how to use software or operate machines, but how to design digitally.

THE MYTH OF THE DIGITAL NATIVE

The common belief that students are self-regulating when it comes to learning and using technology may come from the notion of digital natives. The label “digital native” derives from a series of articles written by the technologist Marc Prensky during the early 2000s. Prensky describes the generation of young people born since 1980 as “digital natives” due to what he perceives as an innate confidence in using new technologies such as the internet, videogames, mobile telephones and “all the other toys and tools of the digital age” (Prensky, 2011). Enrique Dans counters Prensky’s claims: “Simply being born into the internet age does not endow one with special powers. Learning how to use technology properly requires learning and training, regardless of one’s age.” Dans goes on to expand upon the issues of assuming students do not need to be taught to use technology thereby becoming “digital orphans”, lacking in any model to copy or experiences that might have generated criteria for understanding (Dans, 2014).

For this reason, beyond basic fluency, architectural instructors are uniquely positioned to model substantive content creation and healthy critical thinking about these technologies. By perpetuating the myth of the digital native, architectural education is missing the opportunity to establish strong pedagogical foundations from which future digital advancements will emerge.

PEDAGOGICAL ALIGNMENT AND THE VALUE OF DIGITAL DESIGN

The lack of agreement and clarity among schools regarding digital design creates problems for the discipline. How can a skillset be taught without a clear definition? And how can the field evolve when there is such contention over education in a critical area? Dialog and common ground are needed.

A key reason for the confusion surrounding digital design instruction in the university setting is a misunderstanding of its educational value as a set of skills beyond technical skillling. One of the most significant changes made by educational research has been to redefine the goals of learning. Decades ago, before the development of contemporary learning theories, schools emphasized developing core skills such as reading and memorizing information such as dates and facts in a history class. The implicit assumption was that this level of learning was sufficient for students to write reports, solve problems, and produce other sophisticated applications of literacy. However, while many students could demonstrate ability at, for instance, providing the correct solution for a specific type of word problem, educational researchers found that students rarely understood what they had learned, nor could they easily apply their skills and strategies to new contexts (Clement, 1982). The students knew
their lessons by rote and adapted to succeed at their instructor’s tests, but they had a superficial understanding of the material. Today however, educational models and expectations have evolved, digital technology is often relegated to this type of learning.

While skills and facts remain important to learn, the goal of education today has been restated: to provide students with a foundation of deep learning and the intellectual tools to ask and address meaningful questions. (Bransford, Brown, and Cocking, 1999) In contrast to superficial learning of facts and procedures, deep learning entails knowledge of the underlying principles, domain structure, and strategies to activate skills and knowledge and apply them flexibly in a variety of conditions – particularly conditions which are different from the ones where learning originally occurred, such as the translation of design thinking from an academic to professional context. Deep learning is what most instructors would recognize as productive and transferable learning, yet few courses actually achieve. Architectural studios are examples of a deep learning environment.

In contrast to architectural studios, the current state of digital design instruction in architecture tends to follow an educational model which does not support deep learning. Presently, much of what students learn is by rote: sequences of commands and procedures intended to produce reliable results. While students can operate software and other tools with what appears to be great fluency, the vast majority do not have a deep understanding of computing or digital media principles (Senske, 2014). As a result, their work tends to be inefficient and derivative. Like the school teachers in the earlier example, digital design instructors emphasize core skills for using digital tools and then expect students to apply them towards design projects. This is the reason a learning gap exists. First, students do not learn the tools with significant guidance to develop depth and rigour; second, they are not taught explicit strategies for applying digital methods to design tasks. Students often fail to develop an understanding of digital design methods because the pedagogy is not aligned with the goal of deep learning. This leads to a frequently cited criticism of digital design: work which is repetitive or derivative because students are grappling with technology rather than controlling it. The technology does not make it this way – it is how it is used.

This is assuming such a goal is recognized in the first place. Learning digital tools is often seen – by students and faculty alike – as mere technical skilling rather than a way of thinking about design. Professional architectural accreditation (NAAB) in American schools uses a set of learning criteria which specify Ability and Understanding (NAAB, 2014). However, this set of criteria does not address digital design with any specificity. There is no agreement upon the value or content of a digital design education, and so student abilities can vary widely from school to school, and within academic units. Students are less inclined to develop a thorough knowledge of digital design because it is not universally considered a meaningful intellectual and creative pursuit. This not only hinders progress within the discipline, but, in practical terms, it affects the profession. Failure to recognize the principles of digital tools and structures of problems they address makes it more difficult for students to learn and retrain themselves in response to changing technology.
The educational model of the design studio is unique in its approach because it has many elements which contribute to the production of deep learning, such as opportunities for synthetic learning, active learning, complex problem solving, and self-reflection and critique. This is precisely the kind of approach that would benefit digital design education. Unfortunately, the architectural design studio is often seen as one type of learning, while digital design, which is thought of as mere technology, is seen as another. This disconnection is due to a misunderstanding about digital design due to a lack of clearly-defined and shared pedagogical goals. The present situation in education has come about because the implied goal of digital design education is mere tool operation (which does not require deep learning) when the expected outcome should be increased agency and sophistication of design ability. One way to address the problem of pedagogical misalignment is to develop learning objectives for digital design. Learning objectives have the benefit of being a structured, well-understood, and research-based approach to curricular development. This method informs clarity and represents an explicit way to connect the goal of deep learning with pedagogical execution.

BLOOM’S TAXONOMY

A useful tool for developing better learning objectives is Bloom’s taxonomy. The taxonomy is a hierarchical framework intended to help instructors coordinate their planning and assessment using a common language (Krathwhol, 2002). It represents the process of learning from acquiring simpler to more sophisticated thinking skills. The general idea of Bloom’s taxonomy is that lower levels of cognition support higher levels. For instance, one must understand the difference between different methods of constructing a surface (comprehending) before choosing which surface to use (applying).

In its revised form, Bloom’s taxonomy lists six levels of cognitive processes:

1. **Knowing**: memorization and factual recall
2. **Comprehending**: understanding the meaning of facts and information
3. **Applying**: selection and correct use of facts, rules, or ideas
4. **Analyzing**: breaking down information into component parts
5. **Evaluating**: judging or forming an opinion about the information
6. **Creating**: combination of facts, ideas, or information to make a new whole

A more recent addition to the discussion of the taxonomy is the inclusion of types of knowledge. Anderson and Krathwohl addressed criticisms of the taxonomy by recognizing that not all knowledge is equal in complexity and that knowledge tends to be developed from concrete (facts and concepts) to abstract (procedural) and finally to knowledge of one’s own cognition (metacognitive) (Anderson and Krathwohl, 2001). In concert with cognitive processes, the knowledge dimension of the revised taxonomy enables a more nuanced discussion of learning objectives. For instance, under the newer version, the taxonomy does not progress and stop with creating, but also includes thinking about one’s learning progress and how one creates.

Bloom’s taxonomy has been criticized because it does not represent the complex and interconnected nature of cognition (Furst, 1981), but the taxonomy was never conceived of as a model or theory. Nor is it a prescription for every course to follow. One could design a course with at least one learning objective at each level. Depending
upon the skills required, some levels may need additional objectives. Students with different abilities may be able to begin learning at higher levels. The value of the taxonomy is less that it represents exactly how learning works or that it tells instructors how to teach, but rather in how it helps to organize and align pedagogical thinking.

Educational frameworks like Bloom’s taxonomy are not in common use in architectural education. The reason for this is unclear but may derive from a disciplinary resistance to self-articulation. However, for those developing or revising architectural curricula, having access to a set of learning objectives that uses the taxonomy can enable a dialog within the discipline, with other disciplines and educational researchers.

Bloom’s taxonomy helps support the goal of developing deep understanding in digital design instruction. One way it accomplishes this is by establishing the basic cognitive processes involved in learning to design thoughtfully. To see all of these steps organized and consider them with respect to digital design is to shed light on what is often an opaque practice. The taxonomy makes it clear that one does not just use or not use various tools, but one must understand them, choose from them, and evaluate those choices as part of a design process. In this manner, an advantage of learning objectives developed through Bloom’s taxonomy is that they can elevate student outcomes towards higher-order thinking (Biggs, 1999). For example, without the proper outcomes articulated, a student might submit a design, but by merely applying a procedure. Bloom’s taxonomy makes it clear that creation depends as much on understanding one’s decisions (the “why”) as knowing the correct commands (the “how” – which is often students’ focus). For teachers and students alike, Bloom’s taxonomy helps clarify that the goal of digital design instruction is not only to learn how to use digital tools, but to apply them towards better designs and more sophisticated design thinking.

With regards to teaching methodology, the clarity of learning objectives derived from Bloom’s taxonomy can help motivate qualities of student performance which are often lacking in digital design courses, such as innovative solutions and well-crafted, thoughtful representation. As mentioned in the previous section, many learning objectives are not specific enough, sufficiently measurable, or targeted to student’s learning level. Bloom’s taxonomy can help ensure that students are practicing the skills that they should be learning in their activities and at an appropriate level of cognition. This enables the pedagogical gap between learning digital methods and creating designs to be filled with deliberate (or mindful) practice.

Deliberate practice is a recognized process through which individuals train themselves to high levels of performance. Research has shown that learning of complex skills is most effective when students engage with tasks that are appropriately challenging, with clear performance goals and feedback, and sufficiently frequent opportunities for practice (Ericsson, Krampe, and Tesch-Römer, 1993). The difference between merely making and deliberate practice is that a student monitors their progress towards a specific goal and changes their performance in response to feedback. The student continues to do so while increasing the challenge of the activity to further improve. Learning objectives assist students in deliberate practice by creating specific and appropriate performance goals which they can use to monitor their progress. This guidance directly supports the development of abilities on the highest (metacognitive)
level of the taxonomy, which are crucial for sophisticated work and achieving transfer of skills and knowledge to other domains (Perkins and Salomon, 1992). Thus, the notion of deliberate practice stands in contrast to the disengaged ways that many students learn and use digital tools, which is often oriented towards production for its own sake rather than for quality or thoughtfulness. Introducing deliberate practice is one way for schools to motivate deep understanding and to bring craft back into discussions about digital representation.

LEARNING OBJECTIVES FOR DIGITAL DESIGN

The idea of a learning objective is straightforward, but often misunderstood and misapplied. A learning objective is a specific statement which describes what a student will know (knowledge) be able to do (skills) as a result of engaging in a learning activity. A learning objective must have three parts: a measurable verb associated with the intended cognitive process, the necessary condition (if any) under which the performance is to occur, and the criteria for measuring acceptable performance (this is often implied). A simplistic example of a learning objective that fits this pattern is: “Given a set of contours the student will be able to generate a topographic model.” The
condition is having a set of contours and the implied measurement is an acceptable model. Learning objectives are focused solely on student outcomes and do not specify methods or other expectations for the teacher. They are not an attempt to create uniform classroom procedures or hinder instructor creativity through standardization. The teacher has flexibility in their approach, so long as the performance criteria are met. Learning objectives are useful because they help instructors with course planning and the creation of content. Furthermore, the explicitness of properly-constructed learning objectives establishes a basis for student assessment as well as the evaluation of teaching and curricula (Anderson, 2002). A primary challenge of digital architecture evaluation is the lack of criteria and therefore a lack of agreed-upon traits for which to evaluate whether digitally produced code, drawings or images are successful.

In this manner, learning objectives support better learning and provide a common framework for schools to organize their efforts at improving education. For this reason, many universities have standardized their syllabus policies to address learning objectives [see (Vanderbilt, 2016) and (Carnegie Mellon, 2016) for example]. The use of learning objectives may seem obvious or unnecessary if one is only considering their use in one’s own syllabi, but in terms of disciplinary alignment, digital design instruction could benefit from the additional clarity offered.

The real issue is not that learning objectives do not exist for digital design courses, but rather that they are not often used correctly, in response to the findings of educational research. First, many stated learning objectives do not take into account the learning process for developing complex skills and thinking. As mentioned earlier, traditional digital design pedagogy tends to emphasize learning through design tasks. The tacit learning objective of most activities, ostensibly, is to design something via digital methods. However, this does not acknowledge the steps involved to prepare students for design, such as learning about the tools, practicing methods, comparing and selecting methods, etc. These skills and knowledge are implied by the goal of designing, but by not stating this explicitly, the instructor might neglect teaching and assessing the constituent skills and knowledge that students need, but might not manage to learn on their own.

When developing learning objectives, it is important for digital design instructors to acknowledge how learning occurs as a developmental process. Creativity and autonomy, abilities exercised in design work, are higher order thinking skills. Higher order thinking is dependent upon requisite technical skills and other cognitive resources (Weiss, 2003). As such, these activities may not be beneficial learning experiences for beginner and intermediate students. Research shows the importance of matching learning objectives to student level (Klahr and Nigram, 2004). Novices benefit from direct guidance in basic skills and knowledge, while objectives for advanced students should emphasize synthesis and independence.

Second, many learning objectives for digital design instruction conflate activities and goals with learning outcomes. A goal is a statement of the overall intended outcome of a learning activity or course. Learning objectives are specific achievements which contribute to the goal (Ferguson, 1998). For example, a course description that says “students will be exposed to digital fabrication technologies” has presented
a goal, but not stated a specific, measurable outcome. Likewise, a statement such as “students will fabricate a small-scale physical model” describes an activity, but does not provide enough information to discern what students are supposed to learn from the activity. A learning objective that addresses these issues would be: “students will use GIS data to generate a small-scale physical model using appropriate digital fabrication techniques.” This objective presents a condition (GIS data), an outcome (the model), and assessment criteria (are the techniques appropriate? / is the model correct?). Understanding the learning objective helps define the cognitive skill level of the activity and the appropriate assessment. For instance, if the objective was to learn about computing concepts, issuing a quiz with questions about procedures would not be a helpful measurement. To facilitate effective instruction, goals, activities, and learning objectives must be aligned with one another.

Last, many learning objectives as presented do not support a means of formative assessment. Most courses only assign grades for projects, which are typically creative or design work. Again, these are higher order thinking skills and may not be appropriate to assess from novices. Grading project submissions does not give the instructor or the student much opportunity to remediate skills or knowledge that were misunderstood or not acquired. Moreover, feedback on a design artifact may not help instructors and students achieve the goal of deep understanding because it makes conceptual and procedural knowledge indistinguishable from the outcome. Studies have shown that ability to perform procedural tasks does not mean students are able to explain what they are doing or why (Schoenfeld, 1985). This is not to say that instructors should never grade projects. This is appropriate when the intent is to assess creative work and problem solving, particularly from an advanced class. Learning objectives should measure the correct student outcomes for the level of the student and in a manner that allows students to respond with changes in their performance.

SOFT SKILLS AND FOSTERING LEARNING HABITS

The development of rigorous learning objectives is the first part of creating a learning environment for digital design. The second proposal of this paper is to cultivate a
set of complementary “soft” skills which are currently missing in most digital design instruction. Computer use in architecture is often discussed and taught as a series of technical or “hard (as in absolute)” skills. In contrast, “soft” skills are related to emotional intelligence, attitudes, habits, and interpersonal relationships. An example of a soft skill is resourcefulness: being inclined and able to find alternate solutions to a problem, rather than giving up or deferring responsibility. In this manner, soft skills influence the ways that an individual applies technical skills to achieve goals, such as a design. Learning soft skills has been related to improved employment outlook and better job performance (Andrews and Higson, 2008; Nealy, 2005). Professions such as business and information services have cited employees’ lack of soft skills as one of the primary reasons why projects fail (Bancino and Zevalkink, 2007). Thus, for students, developing soft skills is equally as important, if not more important, than learning technical skills. This is because soft skills can be reapplied to changing technology, whereas hard skills may fall away as technology changes.

The influential Boyer report on architectural education concluded that: “[A]rchitectural education is really about fostering the learning habits needed for the discovery, integration, application, and sharing of knowledge over a lifetime” (Boyer, 1996). Soft skills are the learning habits Boyer references and as such must be taught rather assumed to be pre-existing skills. This also extends to those soft skills which relate to digital design in architecture. Hereafter, ‘digital tools’ refers to software programs, computing devices such as laptops, tablets, etc., fabrication systems (laser cutters, 3d printers, CNC machines, etc.), robots, embedded systems, and anything else that involves computers.

Architectural education must recognize that university students are not comprehensively or consistently trained in digital technologies when they arrive on campus. This is exacerbated when less privileged students are potentially less digitally skilled than students from economically privileged backgrounds. By not addressing these inequalities, institutions such as architecture schools are perpetuating disparities through education.

TRADITIONAL VS. DIGITAL SOFT SKILLS

The type of soft skills described in this paper are not entirely the same as soft skills introduced in the previous section. While traditional soft skills such as conscientiousness and empathy are helpful for architects, digital soft skills have a different purpose and apply specifically to the tools and processes used in digital design. Digital soft skills, such as asking clear questions, estimation, and planning skills, enable effective collaboration with other people while using digital tools and promoting effective workflows for collections of digital tools. Digital soft skills support students as they are learning digital design and, later, help students apply technical skills successfully and with sophistication and to adapt to a rapidly changing technologic landscape. Digital soft skills also differ from traditional soft skills because they take into account the particular challenges of computing and digital machinery. The special attributes of digital tools that make them powerful, such as symbolic logic, abstraction, and automation, can invite cognitive biases when designers operate those tools simplistically, at face-value (i.e. using a computer like a cell phone, a pencil, or a typewriter). Humans must adapt their thinking, expectations, and habits, as their
natural inclinations can interfere with working effectively with digital tools (Sheil, 1983). Even those who work with digital tools frequently need to learn digital soft skills, as they may have developed bad habits and misconceptions over time. Merely using digital tools is not enough to cultivate mindfulness of the medium and one’s responses to it.

To cite an example: digital tools are often “black boxes” with complex layers of interrelated procedures that make it difficult for users to be aware of what they are doing and how their software operates. Users expect simple cause-and-effect relationships between their operations and the results on a screen, when the reality is that many “hidden” processes are at work and can affect the outcome of an interaction (Blackwell, 2002). This is also one reason why computers are not always dependable and why they tend to break down in obscure and obtuse ways. Working responsibly with digital tools requires a certain level of comfort and responsiveness with an opaque tool. Students who lack the digital soft skills to understand and respond to this condition often have a poor attitude when faced with computer problems and may spend their time in unproductive ways trying to “hack” solutions to technical
problems (Pea, 1987). This affects not only the quality of their final designs, but their outlook on technology in general. Digital soft skills are similar to traditional soft skills in the way they affect how students apply technical skills. They are the bridge across the gap that often exists between design skills and technical (hard) skills like digital methodologies. Unfortunately, very little time, if any, is given in architectural curricula to the explicit cultivation of digital soft skills.

SAMPLES OF DIGITAL SOFT SKILLS

The following list is a representative sample of digital soft skills which could be taught in an architectural curriculum, organized according to four primary headings.

Communications Skills
Communicating clearly with others is a critical set of soft skills for architects, particularly when using digital tools. For instance, many students have never been explicitly taught how to ask a question via email: to provide necessary information and files upfront, anticipate follow-up questions, and to communicate their expectations for resolution. This is important not only professionally, but especially when trying to learn or fix something like a new piece of software.

Collaboration - The ability to work with others digitally, particularly at a distance. One aspect of this is organizing files and sharing them across computing platforms and software versions.

Authorship - This is the ability to understand digital intellectual property and to distinguish between resourcefulness and plagiarism. This notion of authorship becomes increasingly important when the line between programmer and designer is blurred by the use of digital tools. Of particular note is the downloading of code or Grasshopper definitions which are then deployed as design generators.

Support - Architects should be able to seek, locate, and pursue support for software and technical issues, many of which might exceed the abilities of the instructor or the support offered by an academic institution. These skills include asking fellow students, contacting the software maker directly, and using the Internet as a resource.

Adaptability
Adaptability is resiliency in response to imperfect tools and a field constantly in change. Digital designers should work with the understanding that failures are to be expected, while being empowered to seek alternatives. They must also update their skills and abilities often while remaining critical users of technology.

Autodidacticism – The ability and inclination to teach oneself (quickly) is a valuable skill for designers. This includes planning and scheduling regular time to learn and a recognition of common concepts and methods shared between tools, which can make learning more efficient.

Conversion – An effective strategy for error recovery is knowing how to share data several between types of files and programs. It is important to also note that many computer programs are able to convert various file formats and
often have similar procedures.

**Time Management**

Digital design projects in architecture are often complex, involving many different programs and machines, as well as human team members. Some of these elements can be hands-off (such as rendering) or very hands-on (supervising CNC fabrication). Part of completing them successfully is knowing the workflows involved and having a sense of their coordination and time requirements.

*Estimation* - There is a common misconception that technology makes design faster and easier. It takes experience and skill to determine the full amount of time needed to complete a digital task or processes (e.g. milling, printing, rendering).

*Sourcing* - The ability to identify the most effective tool and process for the development of the idea and in relation to the time available for production. This requires understanding the different elements of digital production such as the difference between a raster and a vector.

*Preparation* - Plan for contingencies and alternatives. Assume some things will inevitably not go as expected and know the options available.

*Scheduling* - Develop internal deadlines, realistic calendars, and skills for planning and implementing a multi-step process. For instance: development of a digital file for fabrication, then fabrication, then post-production.

**Digital hygiene**

Digital hygiene refers to the good habits of caring for equipment, computer hardware and software as well as preventing and recovering from errors.

*Organization* - Maintain files in a structure which is both navigable and searchable by users.

*Backups* - Create a backup routine that is an embedded part of the digital process (cloud, physical media, & storage). This also includes knowledge and use of software auto-backup and recovery. Keep at least one physical backup off-site.

*Clean-up* - Regularly sort, store, and purge project files to manage storage and make important files easier to locate.

**TEACHING DIGITAL SOFT SKILLS**

Many of the examples listed under soft skills can be classified as character or personality traits. Successful students may already practice soft skills and therefore it is often assumed that these are character traits rather than teachable attributes. One might wonder, given the age of many college students, if such habits can be changed. However, the very notion of “soft skills” implies that these behaviors and habits can be taught to students. There is evidence to support the idea that, with training, young adult students can learn new traits and learning strategies (Perkins, 1989). Another common argument is that soft skills are best learned in the workplace. While the workplace presents an authentic context, it does not offer the same opportunities for focused learning as design school. Moreover, one of the reasons for learning soft
skills is to make one more competitive in finding employment. Students should have a sense of how these skills translate into practice before they enter the market.

How can schools teach digital soft skills? Merely lecturing to students about them is not an effective strategy. While lectures can be helpful for delivering information or persuading an audience, changing and developing habits requires more engagement. The method of training varies depending upon the attribute and the audience, however, generally-speaking, habits of learning can be developed through a process of investment and practice.

Supporting a new habit which a student does not create themselves requires helping them understand its meaningfulness. It can be easy to dismiss soft skills out of hand because they might seem to be obvious or less interesting than learning technical skills. For this reason, it is important for the instructor to communicate why new strategies and habits are helpful (McCombs, 1996). Investment begins by identifying the soft skills in question and explaining to students the value of the skills within design and production workflows. To be most effective, those values should be immediate and goal-oriented. Although it is true that developing soft skills can help a student get a job in the future, explaining to a student (for example) that organizing their files saves them time and reduces errors on their current project is less abstract and applies to their current situation. Helping students understand the gaps in their present abilities and how learning soft skills can help close those gaps is the first step toward effective habituation.

To be most effective, teaching soft skills should be integrated with hard skills teaching and preferably in the context of a project (White and Frederickson, 1998). It is not necessary to revamp an entire course around soft skills. An instructor can introduce them where they naturally occur within design and production processes. For example using an error that students commonly encounter to introduce search, problem-solving, and communications skills. Relevant material like this helps focus student attention while a legitimate context helps them retain and access what they have learned later.

Demonstrations can be more effective when they are supported by teaching materials that help organize knowledge for students (Bransford, Brown, and Cocking, 1999). A simple check-list, for example, can help students remember how to organize a digital group project. Once students have mastered the soft skills involved, the student will not need the scaffolding provided by the list. However, if the student makes a mistake or needs to refresh their learning later, the list provides a useful reference and a prompt for activating digital soft skills. Externalizing implicit practices and helping students focus on relevant information and methods improves the effectiveness of soft skills teaching.

Delivering soft skills in class benefits from a coaching approach. Because the goal is to change student attitudes over time, rather than delivering information or procedures, a "one and done" demonstration is not an appropriate teaching style (Mistrell, 1989) (Bransford and Stein, 1984). With coaching, the instructor discusses the advantages of a skill (creating investment), then models the behavior while explaining to the student what they are doing and why. This last step is important because students
need to understand when to apply a skill as much as they need to know the technical operations involved (Scardamalia, Bereiter, and Steinbach, 1984) (Simon, 1978).

Next, students demonstrate the skill and receive feedback from the instructor on their performance. This is followed by more practice and feedback over time and in concert with other skills to approximate holistic design activities. The goal of coaching is to cultivate not just practice but deliberate practice over time – making the student aware of their own actions and motivating retention and refinement (Ericsson, Krampe, and Clemens, 1993). This creates deep and lasting learning.

Adopting a coaching style of instruction requires a change in how students are graded and given other feedback. Most assessment in studios and seminars is summative, meaning it measures the final outcome of a student’s work. This is suboptimal for shaping behaviors, as it does not measure the process sufficiently and is often too late to influence a student’s soft skills. Formative assessment techniques, which encourage personal reflection, timely feedback, and student response are useful support for the “coaching” (Vye et al., 1998). To supplement these techniques, instructors should not only observe student behaviors but review digital files, as well. Many courses emphasize the final artifact and never look at the files involved. Reviewing files is critical so the instructor can observe attributes such as organization, efficiency, and other procedural nuances.

Lastly, in order to properly cultivate habits, soft skills should be reinforced in the studio and lab even when they are not being formally taught. Instructors should be mindful and consistent in their own habits, demonstrating modeled behaviors in their personal actions. For example, an instructor’s demonstration files should be well-organized to set a good example for the students. Student interactions should also emphasize consistent behavior. If a student asks for help with a tool, for instance, the instructor should evaluate how the student asks questions and replay the scenario with them while making explicit the strategies involved. Learning should be embedded in the classroom experience. It must be a continuous practice, not merely an exercise.

DISCUSSION

The challenge of making claims about design pedagogy interventions, like soft skills, is proving their effectiveness. In educational research the difficulties of empirical measurement in traditional subjects like math and reading are well-known (Black and William, 1998; Shepard, 2000), but the challenges of demonstrating the impact of an intervention upon design outcomes – which are not easily measured or quantified – make this task even more burdensome and its conclusions unreliable. As such, there is no accepted model for proving the effectiveness of design pedagogy. What is more important and perhaps easier to ‘prove’ is that well-articulated digital soft skills create a framework and a platform where technology can be used expansively and in unique ways rather than reductively and repetitively. The value of digital soft skills is to suggest a replicable model which remains relevant and useful for students as technology changes, improves, and adapts.

With regards to learning objectives, their value is not what they add to a syllabus, but rather how they prompt a larger conversation about educational and professional
values and standards. Creating learning objectives for digital design in architecture exposes many implicit assumptions about what faculty believe about learning and the role of computing in the studio. At the same time, discussing learning objectives is a provocation towards architecture schools to consider digital design as more than merely learning to operate tools and software (activities which are not themselves valid learning objectives) and to instead connect these practices to design thinking and the development of architectural designs.

Bloom’s taxonomy assists in framing a more constructive discussion about learning to design digitally by offering a structure of cognitive accomplishments for students. This helps re-align architectural educators away from frameworks derived from folk pedagogy and towards established theories and research into educational psychology and learning cognition. Instead of teaching and learning digital skills and knowledge through a hierarchy of the tool’s features or increasing complexity, Bloom’s taxonomy foregrounds processes of remembering, thinking, and judgement. These objectives are more closely aligned with deeper understanding and integrative mastery. This type of learning is precisely the antidote to the kind of superficial engagement one often finds in architecture schools that prompts negativity towards the use of computing in design.

The purpose of reflecting upon learning objectives for digital design in architecture is not to produce a definitive list of what students ought to learn. Learning objectives are written for specific curricula, student needs, and faculty interests. They are useful because they provide a clear definition of expected outcomes and which becomes a point of dialogue. In order to evaluate something, it first must be named. Through evaluation and discussion, a discipline develops. When Bloom created the learning taxonomy, this was the goal. Not to explain or lay claim to how students must learn, but to provide a shared structure so educators could compare their approaches. In a similar manner, creating and sharing learning objectives for digital design instruction can produce a more organized dialogue about how to align the use of digital tools with the core values of architectural education and the development of the discipline itself. The development of a more coherent set of evaluation criteria in digital education will increase the rigor of conversations about the future of digital design in architecture. Learning objectives are not only for evaluating one’s students or teaching. They help departments and educators understand whether they are teaching the right things. The question should always be: “how does this improve design?”

CONCLUSION

While digital design skills are critical for 21st century designers, architectural education must also recognize and deliver more than technical proficiency. Working creatively and effectively with computers, digital fabrication machines, and other devices requires a new set of workflows and adaptations to academic and professional behaviours. Boyer’s report makes it clear that one of the key values of an architectural education is developing learning habits. A present gap in student learning is that traditional learning habits have not been updated in response to changes in technology (Boyer, 1996). Learning objectives and soft skills for digital design can help to bridge these gaps.
Incorporating learning objectives and soft skills into existing digital instruction may require more work from both the instructor and the students, but the benefits are lasting. Becoming more aware of one’s process and developing good digital habits pays off, no matter what software or tools one encounters. Ultimately, teaching learning objectives and soft-skills is about making students more independent and self-directed learners. With the rapid pace of technological change, students need to be comfortable with and capable of learning, relearning, and integrating new programs and tools throughout their career. For these reasons, learning objectives and soft skills can and should be implemented throughout digital design education.

Learning objects and soft skills support the goal of not only working well with technology, but together with other people in technologically-supported ways. Knowledge, abilities, attitudes, and habits not only shape one’s process, but one’s design goals and outcomes, as well. Soft-skills and learning objectives impact design and so they extend beyond pedagogical or semantic arguments. They should be of interest to anyone who values how technology supports good design.

REFERENCES


