# Soft Skills for Digital Designers

Shelby Doyle, Nick Senske | Iowa State University

## Introduction

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 [1] to a firm belief that these technologies will revolutionize the way architects practice and think about design. [2] 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 digital design instruction, as tools are misunderstood and misappropriated by students and teachers alike.

The aim of this paper is to take control of the pedagogical agenda for digital design in architectural education by debunking the myth of the digital native and by defining a new set of soft skills for computational design and digital representation. This paper is a discussion of architecture, design, and education; not an argument for software and computer use in design. Soft skills provide a framework for learning and understanding digital skills which in turn support the development of technical skills. These base proficiencies in turn facilitate the development of sophisticated digital architectural designs.

## Soft Skills and Fostering Learning Habits

Computer use in digital design 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. Professions such as business and information services have cited employees' lack of soft skills as one of the biggest reasons why projects fail. [3] For students, developing soft skills is equally as important, if not more important, than learning technical skills.

This paper proposes that a set of complementary "soft" skills is missing from most discussions of digital pedagogy and that teaching these skills can improve student outcomes and the integration of digital technologies into architectural pedagogy.

*Fig. 1 Knowing how to operate a smartphone does not necessarily make one an effective computer user.*

While soft skills have a role to play in professional education and practice, they are not to be confused with professionalism. [4] Professionalism is a social construct about social behavior in a professional setting. At their core, soft skills support and activate learning. 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.* [5]

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 and digital tools. [6] 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.

## 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." [7] 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. [8]

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 digital foundations from which future digital advancements will emerge.

## Traditional vs. Digital Soft Skills

The kind of soft skills described in this paper are not entirely the same as soft skills introduced earlier. 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 promote 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.

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. [9] 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. [10] 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. [11] This affects not only 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. Unfortunately, very little time, if any, is given in digital design curricula to the explicit cultivation of 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.

***1. Communications Skills***

Communicating clearly with others is a critical set of soft skills for digital designers. 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.

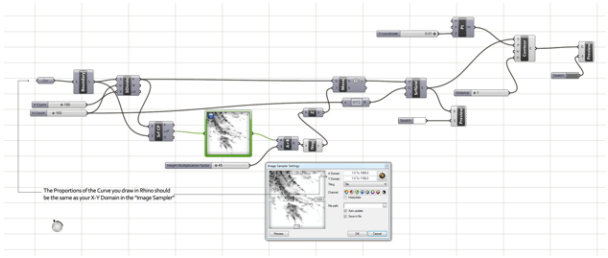

*Fig. 2 Example of a downloaded Grasshopper definition. Working digitally demands questions of authorship and intellectual property be discussed with students.*

- **Collaboration** - The ability to work with others digitally, particularly at a distance. One aspect of this is organizing files and sharing them across a digital platform.

- **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** - Designers should be able to seek, locate, and pursue support for software and technical issues, many of which might exceed the abilities of the instructor of 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.

*2. 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.

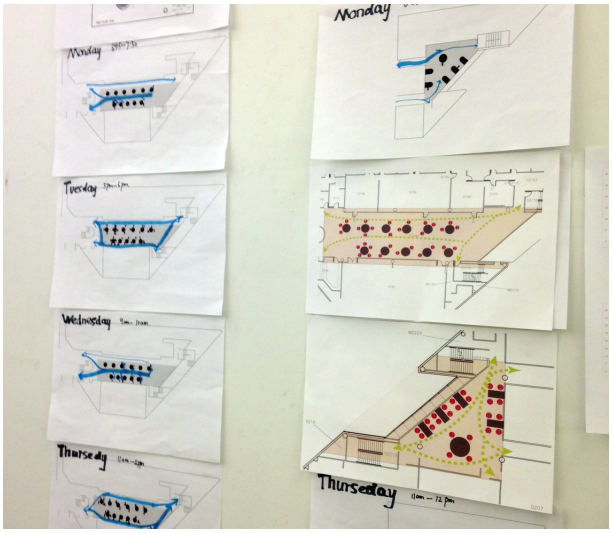


*Fig. 3 Example of a response to a large print which failed. Soft skills encourage students to anticipate such failures and to develop alternatives, such as printing on 8.5x11 paper, creating analog versions, and using a projector.*

*3. Time Management*

Digital design projects 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.

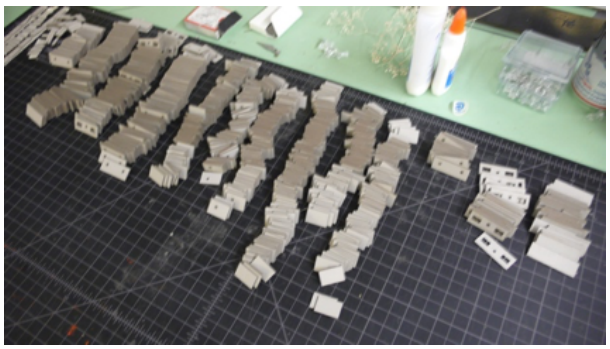

*Fig. 4 Example of a time management and workflow issue common in digital production. It must be reiterated that the computer is not automatic nor is digital production in and of itself 'fast.' A tedious laser file will become a tedious model to assemble.*

- **Estimation** - Determine the full amount of time needed to complete a task or processes (e.g. milling, printing, rendering).
- **Sourcing** - Identify the most effective tool and process for the development of the idea and in relation to the time available for production.
- **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.

*4. 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.

*Fig. 5 Example of a back up protocol. Soft skills enable students to feel confident that computers will fail and that they are empowered to seek alternatives.*

- **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 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. [12]

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 them 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. [13] 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) how 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.[14] 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.[15] 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. [16, 17] 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. [18, 19] 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. [20] 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 students' 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". [21] 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.

## 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 professional behaviors. Soft skills support the goal of not only working well with technology, but together with other people in technologically-supported ways. Attitudes, habits, and workflows not only shape one's process, but one's goals and outcomes, as well. Soft skills impact design and so they should be of interest to anyone who values good design.

Incorporating 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 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, soft skills can and should be taught in foundation design.

## Notes

[1] Pullasmaa, Juhani. The Eyes of the Skin: Architecture and the Senses. John Wiley & Sons, Inc., 1996.

[2] Kieran, Stephen and James Timberlake. Refabricating Architecture: How Manufacturing Methodologies Are Poised to Transform Building Construction. McGraw-Hill Education, 2003.

[3] Bancino, Randy, and Claire Zevalkink. "Soft Skills: The New Curriculum for Hard-Core Technical Professionals." *Techniques: Connecting Education and Careers (J1)* 82.5 (2007): 20-22.

[4] Professionalism accounts for the skills, good judgment, and polished behavior crucial to professional success. These traits include: consistent academic preparation, actively engaging in discussions, meeting deadlines, collaborating courteously with classmates, giving and receiving respectful academic criticism, incorporating design feedback, managing time effectively, respecting the course space and building, and communicating in a polite and considered manner.

[5] Boyer, Ernest L. and Lee D. Mitgang. Building Community: A New Future for Architecture Education and Practice: A Special Report. Jossey-Bass Inc., 1996. (Preface xvi)

[6] 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.

[7] Prensky, Marc. "Digital Natives, Digital Immigrants." On the Horizon. MCB University Press, Vol. 9 No. 5, October 2001.

[8] Enrique Dans, "The Absurd And Unfounded Myth Of The Digital Native." Jun 4, 2014 https://medium.com/enrique-dans/the-absurd-and-unfounded-myth-of-the-digital-native-45d1ff397785#.t959mvxza Accessed January 9, 2016

[9] Sheil, Beau A. "Coping with complexity." *Office Technology and People* 1.4 (1983): 295-320.

[10] Blackwell, A. "What is programming?" *14th workshop of the Psychology of Programming Interest Group*. 2002.

[11] Pea, Roy D. "Logo programming and problem solving." (1987).

[12] Perkins, David N., and Gavriel Salomon. "Are cognitive skills context-bound?." *Educational Researcher* 18.1 (1989): 16-25.

[13] McCombs, Barbara L. "Alternative perspectives for motivation." *Developing engaged readers in school and home communities* (1996): 67-87.

[14] White, B. Y., and J. Frederickson. "Inquiry, modeling, and metacognition: Making science accessible to all students." *Cognition and Instruction* 6: l. 1998.

[15] Bransford, John D., Ann L. Brown, and Rodney R. Cocking. How people learn: Brain, mind, experience, and school. National Academy Press, 1999.

[16] Mistrell, J. Teaching Science for Understanding. pp129-149 in Resnick, Lauren B., and Leopold E. Klopfer. Toward the Thinking Curriculum: Current Cognitive Research. 1989 ASCD Yearbook. Association for Supervision and Curriculum Development, 1989.

[17] Bransford, John D., and Barry S. Stein. "The IDEAL problem solver."1984.

[18] Scardamalia, Marlene, C. Bereiter, and R. Steinbach. "Teachability of reflective processes in written composition." *Cognitive Science* 8.2 (1984): 173-190.

[19] Simon, Herbert Alexander. Problem solving and education. Carnegie-Mellon University, Department of Psychology, 1978.

[20] Ericsson, K. Anders, Ralf T. Krampe, and Clemens Tesch-Römer. "The role of deliberate practice in the acquisition of expert performance." *Psychological review* 100.3 (1993): 363.

[21] Vye, Nacy J., et al. Cognition and Technology Group at Vanderbilt. "SMART environments that support monitoring, reflection, and revision." *Metacognition in educational theory and practice*: 305-346. 1998.