Future of Business and Finance

Tamara Carleton
Shaun West
William R. Cockayne *Editors*

Inventing the Almost Impossible

Creating, Teaching, Funding, and Leading Radical Innovation

Springer

Teaching Imagination and Future-Shaping Skills: What Do Universities Offer Students to Help Them Imagine and Invent?

Ted Selker

Abstract This short chapter reviews some of the ways that imagination is promoted and sometimes learned in and around schools. Parents and teachers try to help children view, review, and work with things that expand their purview. At school, we are shown places, things, and techniques we never imagined before. Such envisioning can be done in ways that are prescriptive and limiting to the course's test on the topic. Today many are pushing to make the experiences more explorational by exposing students to possibilities, places, and things, and techniques can be used with material already known and motivation to find out more.

Imagination is visualizing things that do not but might or might not be possible. It is a critical part of creativity. Some worry that people are only creative when they are children. Some worry that only some people are creative. There is a rich literature around training people to think outside the box. If you listen too carefully to Sternberg's tomb on creativity, however, you will come to believe that creativity cannot easily be taught and is difficult to transfer across domains.

Actually, we all are creative in some areas. Somehow everyone can dance, be it building a shelter, the way you pack, the way you pick which shirt to wear, the way you fix the closure of a bag, everyone makes some creative decisions every day.

Outside classes, schools can offer mentoring experiences where students work with projects ranging from research or planning to constructive. Again, these experiences can be prescriptive such as asking a person processing a circuit board to dip the board in a solution. It can be exploratory, asking a person to learn how this equipment works so we can etch a circuit board.

More and more schools require some sort of internship for students, giving them as many chances to see how professionals do the work they are hoping to learn to do. Antioch specifically was known for being all about internships. Again, the internship can be prescriptive furthering the idea that you have to use procedures or it can be expanding, helping students discover new things in their projects.

Graduate programs again can be prescriptive: "Take these courses to get mastery and receive a MS degree." They can alternatively be thesis oriented, giving students

T. Selker (✉)
Palo Alto, CA, USA

T. Carleton et al. (eds.), *Inventing the Almost Impossible*, Future of Business and Finance, https://doi.org/10.1007/978-3-031-36224-8_7

a feeling that the courses and the lab work are in service of discovering a path that you want to pursue around a project that is sized well for that journey.

Schools should and try to work to expand students' abilities to imagine in new ways. They can include courses in design process, envisioning, entrepreneurism, and interdisciplinary activities such as are practiced at the Hasso Plattner Institute of Design at Stanford University.

Imagine games are typical and encouraged for babies and children. It might seem trite, but practice is helpful in everything. We get used to imagining the same things, should think about ways of imagining differently.

1 Projects as School

Hampshire College and Evergreen State College were both set up based on a book that showed that the curricula of Amherst College stifled the most creative students who wanted to take the initiative. The students are required to do a small project of their choosing approved by a committee of faculty in each of the college's five schools. This Division 1 project is then described and "defended" in front of a committee of faculty in the school it is designed to teach. They are then required to do two Division 2 projects that are much bigger in two of the schools. Finally, they get a degree when they complete and defend a Division 3 project, or thesis, in one area. The students at these colleges are encouraged to take classes only when they support the education. Other universities allow or require capstone projects for undergraduates too. These can be transformational experiences in helping a student develop their sense of goals, project, planning, and scoping.

2 Projects in Classes

A project-based class might include a project that a team works on together. Many times, the teachers start by teaching students the tools they will need; they then give the students the problem and describe several solutions. This focuses the students on the project-based issues of scoping, planning, coordinating with others, and presenting the idea. These are difficult and important skills to learn. Often these experiences are memorable for their bonding potential and intensity. In such a forum, the constraints work to focus the participants.

The same things that focus the participants can limit their learning of other things. The variability in engagement that occurs in many project-based learning activities led me to modify my project-based teaching approach. In a computer science class, we want everyone to learn the basics of computer science. In a design class, we want everyone to learn design. In a project management class, we want everyone to learn project management. If in a project one person takes on the project management role, another the coding role, and another the design role, they are each learning very different skills of management, coding, and design. I also found that a strong or even flashy personality in a group would limit the imagination, creative contributions, or ownership of the project by others. In a very real way, they took different classes. They should get credit for things they learned. Typically, we either think we can gloss over the differences in what different people learn in a group project and say they learned from the others or have two classes that collaborate siloed by topic. I sought a different model.

The model for my Android product development course was to enable each student to own their creative solution. As a graduate level class, I told them, "You know your computer science and I will not be teaching syntax of Java in this class." The whole course used the same tools and had to use the same programming environment, create posters from a template, create a one-minute video, and show their work as a demo. The bonding experiences were the similarity of tools and products they could help each other with and the group discussions about their progress. My bonding tool for the group was to set them up so with one button click they could all download a complete program development system with a phone simulator for the development phones I handed out. By everyone having the same and functional tools the first day of class, they had that to share. Everyone was to be responsible for their own project. Everyone had to come up with five ideas in the first week. I have never faced as much blowback as asking them to come back with ideas. I received multiple paragraph emails stating that no teacher had ever required them to produce ideas by themselves. They complained that to teach these skills I needed to give them programs to write that emphasized the skills.

Instead, I lectured the first day about all the apps that existed that used a special feature of the phone: the phone has so many sensors that are not found on a computer, the phone has multiple radios not found in a computer, and the phone can be used anywhere. I gave many examples of novel scenarios for phones. They had to come back having researched alternative things they could work on having researched what has been done, what cannot be done easily, and what could be done with less than 80 h of programming.

The second class was only getting reactions to the ideas people brought. Is it really too much for the time you will have this semester? Is it really interesting relative to things other entrepreneurs and researchers are doing? What are the tradeoffs of your five ideas? Pick two ideas for which you will try to prototype the hardest part this week.

By engaging the whole class, people experience a range of architectural, usability, scenario, business, and project sizing issues more deeply.

During week 3, students came back having asked help from each other and me. Their eyes are more open to the difficulties in their visions and the realities of implementation. The discussion of so many alternatives gives the class an overview of the programming issues in mobile development, ways of making decisions about direction, and scoping.

In week 4, the students need to come back with a mostly working version of a project. This includes a demonstration that all the hard parts of the project are being tested. They discuss alternatives such as the fast Fourier transform analyzer that one student used to tune a guitar. The exposure to other students' projects helps people know who to go to for help on the rest of the app, how to debug their app, and how to simplify and re-scope their project. The assignment now is to start making a poster and a video. I lecture about communication approaches in which I talk about the need in a video or for a viewer of a poster to be tantalized repeatedly and at least every 20 s of viewing.

Week 5 is spent talking about the poster: title, images, references, and results to get them thinking about the story. We show and discuss what can be done in a 1–2-min video.

We talk about the many problem apps that are in need of rescue.

During week 6, we are showing working apps. We are talking about sketches for videos; we are showing sketches of posters.

Week 7 is when we present to others.

The point of the short summer intensive smartphone development example is to show several ways I got every student to individually learn deeply the important parts of the curricula: idea generation and pruning, pivoting, scholarship, development tools and practice, collaborative problem solving, and presentation.

As a student I took independent study classes to try my ideas with a professor. It is much lonelier than my smartphone class. I did not have others around using the same tools, asking similar questions, and debugging the problems I was or there to present to. I also rarely had the luxury of changing my direction.

Research projects are even more constraining; they have a predetermined hypothesis that was formed before the project started. This does not create imagination.

The spurt of fun happens when people bring me problems. At theMIT Media Lab, Master Card brought the need for a secure RFID. I designed several. The immediacy of need of a request can help the imagination.

In my Industrial Design Intelligence class, I held an idea exploration session. This lecture consists of me bringing a problem into the class and soliciting solutions. The thing I did was classify and ask for more. So when an idea about a car door that will not hurt a bicyclist is soft, I put it in one column; when another person adds that it might break off, I put it in the same column. I then ask, what other ideas can we generate. Someone says the door can slide back, another says under another says over the car. The non-hinged door is now a category column. I ask for more approaches. One person says, "Let's use a camera to find the bike and lock the door." Another adds, "Let's signal the bike that the person is opening the door." By adding different category columns for different approaches, I was trying to inspire people to imagine the maximal alternative categories of ideas. The goal was to celebrate and judge an expressed idea as part of a column or to create a new column. Each new idea pushed people to try to imagine a new approach. These sessions were generative and exciting. They were among the best teaching experiences I have had.

Imagination is not vague; it has a story.

Imagination is not isolated; it is context.

Imagination is not singular; it comes in relation to many possibilities.

The maker movement may have come out of the old world of books like The Boy Mechanic, projects in periodicals like Popular Science, and a whole genre of creation-based education exemplified by making toys like erector sets and LEGO. Erector sets, chemistry labs, and Legos dominated the childhoods of many post-war children teaching them about creating. As a child, I ogled the Heathkit catalog. It was filled with kits you could put together to create every electronic gadget on the market from the components. Most of these project-based toys did not have deep pedagogy (but aspired to). Project-based activities with computers require more concentration and planning than other project-based activities. If the programmer has not thought through their program, it does not work. The only way to make a program work was to imagine it, design it, build it, and debug it. Until every aspect of the idea was fleshed out and translated into the syntax of the computer, it did not work.

I like to think that the maker movement followed this ideal. You must understand the parts and their relationships to make something with a laser cutter or 3D printer. You must at least understand the interfaces and instructions of Sparkfun parts to make something from them. The maker movement may not have embraced invention, in that scholarship was not the point. It embraced creating a system from parts. The idea of makerism is that we can make almost anything with parts and software. The idea was for this to spur imagining more.

The missing ingredient in many "how to make anything" courses and maker shops is needs finding. Without a good understanding of what is needed, it is hard to have a vision of what should be. If there is one thing that I criticize most in most curricula on makerism, it is the way that scholarship and needs finding is done. Many curricula celebrate asking people what they want. This is great for learning a bit about an area you know little about. It is not great for becoming an expert. My whole life I have walked through buildings designed by people that did not know what it is like to use the buildings. Many famous buildings echo loudly, making people feel small and disoriented. How is it that some companies want every entrance and building to be more important and impressive than you should deserve? How is it that most hospitals are mazes of confusing corridors you typically traverse when already stressed and disoriented?

Anything we make should be guided by an appropriate vision for its use, not just the presentation of the project. This approach requires context and empathy. These ideals are different from clipboard lists recorded with experts. The experts remain the knowledge sources, but you must internalize by putting down the clipboard and being the expert to know what is needed.

In classes, we use role playing, scenario creation, props, mockups, and prototypes to get a feel for ideas. This is the stuff that feeds imagination. Often missing are the alternative roles, scenarios, props, mockups, and prototypes that can make us reflect on options as we finalize a solution.

One of my first successful project-based educational experiences was getting 30, 3-person teams to each make an AI-based computer programming help system that adapted to the user's demonstrated knowledge. I started the course by spending 6 weeks teaching many AI approaches to representation, reasoning, and learning. I told the students they had to use some of these ideas in their resulting system. I created and gave them a window-based UX structure to present the information about

LISP in. By the assignment constraining the context, the 30 groups all made systems that made sense to each other. They all made different kinds of solutions that helped them all see different possibilities. The various groups collectively built a landscape of alternatives that became more interesting than any one system could have been. The resulting lessons from the alternatives were strong: an AI learning system that responds immediately is more engaging than one that might change things in the future. This example showed me the value of alternative ideas to help form a better understanding of the problem trying to be created.

My approach is to always be learning every tool in case it might be useful. When I find a problem I ask, "Do I have some things I know of that can help solve this? Can I put together techniques known to a person in the art to solve this?" Then I ask, "What if I had new techniques, how would that improve the solutions I can create?" That is the moment of serious imagination; "What are the futures that allow me to build better solutions?"

Tamara Carleton and Bill Cockayne's Moonshots courses and teaching enable students to focus on modern horizon thinking, taking imagination in a different direction. The idea is to think about what could be done now, a bit in the future, at a time when a novel invention would be needed to do it and at a time when most of today's constraints are no longer constraints. This takes imagination in a very different landscape of expanding scopes that stretch people out of their what they know can be done cocoons.

Still, we go back to literature about creativity and productivity. Mihaly Csikszentmihalyi taught us that we should work on things at a level in which we know enough about what we are doing and how to do it to make progress. He calls this state Flow. Jeff Dow did a series of demonstrations that show that considering multiple solutions in parallel outperforms interactive improvement in design. Sternberg wrote a whole handbook on creativity that covers many experiments that show that people can improve any skill they work at in a specific domain. He showed much worse results in teaching people to be creative across domains. I find the concern that people do not transfer creative skills seems cynical. II see encouraging people often spurs them to new successes.

Universities are places where we can learn anything. A catalog of courses serves hundreds of different degree alternatives. The catalogs of universities spurred my imagination as I began my university studies. Then the reality comes through: an ABET approved engineering degree has so many required courses that a student's schedule is full of physics for engineers, statics, systems, and other classes that do not sound so imaginative. Business students take none of these but are busy learning how to do accounting, marketing, sales, and so forth. So where is the imagination in such a university? As well as the Antioch and Hampshire colleges described above, Stanford's Hasso Plattner Institute was another answer to this, bringing people across curricula for exercises in problem solving. This is great at least at getting people to see that there are other ways of thinking about things. I dream that we are starting to move farther.

The space between engineering and business curricula is technology invention. When we understand and get better at teaching the skills of entrepreneurship and technical evaluation of new ideas, we will have a special class of skilled workers. These workers will be able to evaluate engineering solutions for the many steps needed to move them from a prototype to something that can be sold, built, supported, and improved to make a continuing and productive enterprise.

We naturally enjoy and support imaginative exploring play in children. Learning to play a piano is not about imagination; it is about skills, craft, and mastery. Learning to dance specific ways is also craft and mastery. These crafts do require envisioning and concentration. What they do not easily support is the imaginative expansive self-guided creative exploration. Interpretive dance or creating new dances requires imagination. We naturally gravitate to focusing on being thorough and dependable in learning the trades in our young adulthood. Lest we lose the ability to form novel problems and solutions we must work to keep parts of the imaginative exploration in our schedule always.

Recommended Reading

Burleson, W., & Selker, T. (Eds.). (2002). Creativity and interface. *Communications of the ACM (Special Issue), 45*(10), 88–90.

Carleton, T., & Cockayne, W. (2023). *Building moonshots: 50+ ways to turn radical ideas into reality*. John Wiley & Sons.

Carleton, T., Cockayne, W., & Tahvanainen, A. (2013). Playbook for strategic foresight and innovation.

Kaufman, J., & Sternberg, R. (Eds.). (2019). The Cambridge handbook of creativity (2nd ed., Cambridge handbooks in psychology). Cambridge University Press.

McKim, R. H. (1972). *Experiences in visual thinking*. Brooks/Cole Publishing Company.

Resnick, M., Myers, B., Nakakoji, K., Shneiderman, B., Pausch, R., & Eisenberg, M. (2005). *Design principles for tools to support creative thinking.* Report of Workshop on Creativity Support Tools. No. 20.

Selker, T. (2005). Fostering motivation and creativity for computer users. *International Journal of Human-Computer Studies, 63*(4–5), 410–421.

Selker, T. (2008). Fostering motivation and creativity for computer users. In *Proceedings of NSF CreativeIT Workshop at Arizona State University*.

Shneiderman, B., Pausch, R., Selker, T., & Eisenberg, M. (2005). *Creativity support tools: A workshop sponsored by the NSF* (pp. 25–36).

Ted Selker is an American computer scientist known for his user interface inventions. Ted spent 10 years as an Associate Professor at the MIT Media Laboratory where he created the Context Aware Computing group, co-directed the Caltech/MIT Voting Technology Project, and directed the CI/DI kitchen of the future/design of the future project. Ted's work strives to demonstrate considerate technology, in which people's intentions are recognized and respected. He is well known as a creator and tester of new scenarios for working with computing systems. His design practice includes consulting to help dozens of startups and large companies, speaking engagements, and innovation workshops. His successes at targeted product creation and enhancement led to his role as IBM Fellow and director of User Systems Ergonomics Research at IBM. He has served as a consulting professor at Stanford University and taught at Hampshire College, University of Massachusetts at Amherst, and Brown University. He worked at Xerox PARC and Atari Research Labs.