# THE HACK-CHARRETTE

## DESIGNING A MODEL FOR ENGAGING TEAMS IN TECH INNOVATION

by YARA SAFADI

# THE HACK-CHARRETTE

## DESIGNING A MODEL FOR ENGAGING TEAMS IN TECH INNOVATION

by YARA SAFADI

PUBLISHED BY

MiD | MASTER OF INDUSTRIAL DESIGN
THE UNIVERSITY OF THE ARTS

320 SOUTH BROAD STREET
PHILADELPHIA, PA 19102

COPYRIGHT © 2014

Copyright © 2014 by Yara Safadi, and the Master of Industrial Design program at The University of the Arts

Illustrations copyright © 2014 by Yara Safadi

Photography credits:
All photography created by © 2014 Yara Safadi unless otherwise noted.

All rights reserved. No portion of this book may be reproduced-mechanically, electronically, or by any other means, including photocopying-without written permission of the publisher.

Cover design by Yara Safadi
Book design by Yara Safadi

Master of Industrial Design at The University of the Arts
320 South Broad Street
Philadelphia, PA 19102

http://mid.uarts.edu

First printing: May, 2014

# THE HACK-CHARRETTE

## DESIGNING A MODEL FOR ENGAGING TEAMS IN TECH INNOVATION

by YARA SAFADI

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree Master of Industrial Design

in the College of Art, Media, and Design
The University of The Arts
Philadelphia, Pennsylvania

May, 2014

Thesis Committee:

Advisor: Michael Zarro, 5th Year Ph.D. Candidate at Drexel University's College of Computing and Informatics

Readers:
Sharon Lefevre, Professor, The University of the Arts
Jeremy Beaudry, Director, Master of Industrial Design, The University of the Arts
Chris Garvin, Director, Lamar Dodd School of Art, University of Georgia

*For the extraordinary people in my life*

*Anan*
*Dad, Mom, Keenan*
*Lydia, Adam, Jude, Marcel*
*My Grandmas, My Grandpa*

*For your great love and support*

# ACKNOWLEDGMENTS

MY SPECIAL THANKS ARE EXTENDED TO ALL THOSE WHOSE SUPPORT GUIDED THE SHAPING OF THIS THESIS. THANK YOU MICHAEL ZARRO, CHRIS GARVIN, JEREMY BEAUDRY, AND SHERRY LEFEVRE FOR ACTIVELY GUIDING MY WORK FROM BEGINNING TO END. THANK YOU MEREDITH WARNER AND JONAS MILDER FOR YOUR INVALUABLE FEEDBACK. THANK YOU MELISSA VISINTIN, CAROLEIGH DENEEN, AND KIRSTEN GAY FOR YOUR INDUSTRY EXPERTISE AND FOR HELPING ME REFINE MY IDEAS IN THE INITIAL PHASES OF THIS PROJECT. THANK YOU MATT VAN DER TUYN FOR YOUR ENCOURAGEMENT. THANK YOU BRANDON KLEVENCE FOR YOUR SUPPORT AT PILOTDC. A SPECIAL THANKS TO KEVIN LEE, MAYANK JAIN, AND ALL THOSE I MET THROUGHOUT THIS JOURNEY FOR YOUR TIME AND SIGNIFICANT INSIGHTS. THANK YOU JEFFREY WOLPER, PAMELA TUDOR, AND MID FACULTY FOR YOUR CONTINUOUS MENTORSHIP.

# ABSTRACT

CONSTANT DEVELOPMENTS IN THE FIELD OF INFORMATION TECHNOLOGY HAVE MANAGED TO SOLVE MANY COMPLEX PROBLEMS, AND MOBILIZE OUR ADVANCE INTO THE FUTURE. HACKATHONS, WHICH ARE INNOVATION EVENTS IN THIS INDUSTRY, CONTINUE TO GROW AS A HUB FOR SOFTWARE CREATORS TO COME TOGETHER, SOLVE PROBLEMS, CREATE, AND INNOVATE. THESE EVENTS FOSTER INNOVATION BUT EXCLUDE NON-TECHNOLOGY VOICES, AND LACK A STRUCTURE THAT SUPPORTS EFFECTIVE TEAM ENGAGEMENT.

THE HACK-CHARRETTE VIEWS HACKATHONS AS AN ENTRY-POINT FOR SUPPORTING A SHIFT THAT THE INDUSTRY IS UNDERGOING, NAMELY THE FOCUS ON HUMAN-CENTERED DESIGN IN SOFTWARE. THE PROJECT INTRODUCES DESIGN METHODS AND TOOLS INTO THE HACKATHON SPACE TO SUPPORT TEAM ENGAGEMENT, AND TO IMPROVE THE LEARNING EXPERIENCE. IT ALSO OPENS THE DOOR FOR MORE DIVERSE PARTICIPANTS TO JOIN IN DECIDING WHAT OUR FUTURE TECHNOLOGIES LOOK LIKE.

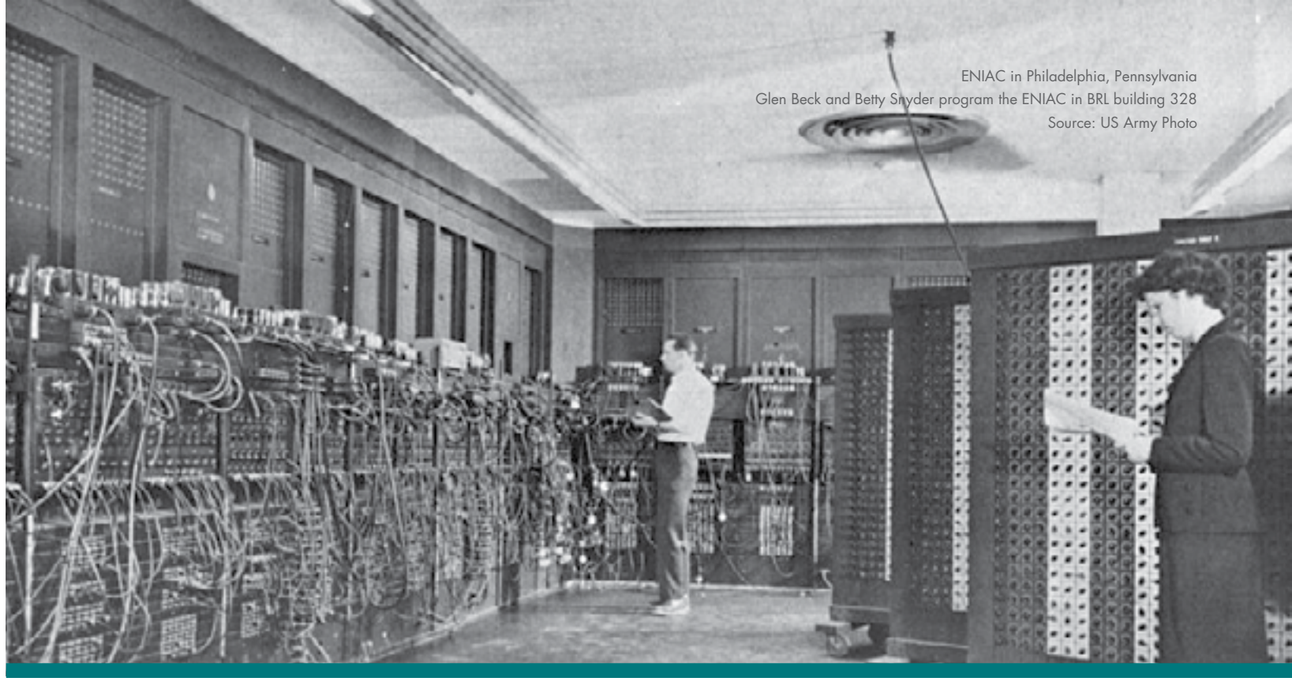
AT ITS EARLY STAGE, THE RESULTS FROM THE PROJECT HAVE SHOWN THAT THE INTRODUCTION OF DESIGN METHODS AND TOOLS CAUSED TEAMS TO BRAINSTORM MORE EFFECTIVELY, THOUGHTFULLY CONSIDER THEIR END USER'S NEEDS, MAKE COLLABORATIVE DECISIONS AND ESTABLISH CLEAR PROJECT GOALS.

# CONTENTS

# I. CONTEXT & DISCOVERY

# INTRODUCTION

## *Innovation In Information Technology*



ENIAC in Philadelphia, Pennsylvania
Glen Beck and Betty Snyder program the ENIAC in BRL building 328
Source: US Army Photo

### Information Technology And Its Impact

Technology supports and enables many aspects of human activity. Innovation in this field has been and continues to be crucial for human development. Considering that technology is such a vast field, my thesis project focuses more specifically on innovation in Information Technology (IT).

The impact of IT is best described in a 2012 study conducted by the National Research Council of the National Academies titled *Continuing Innovation In Information Technology*. IT plays a key role in many aspects of our lives like health care, education, and communications to name a few (National Research Council 12). While the study's purpose is highlighting the importance of research in IT, it collects key facts and figures on the industry's impact on the economy:

> In total, according to estimates for 2010 by the Bureau of Economic Analysis, the IT-intensive "information-communications-technology-producing" industries grew by 16.3 percent and contributed nearly 5 percent to the overall U.S. gross domestic product (GDP). A 2011 study by the McKinsey Global Institute found that in 2009 Internet-related activities alone contributed an average of 3.8 percent to the U.S. GDP […]. These substantial contributions to the economy reflect only the direct economic benefits of the IT sector and do not capture the full benefits realized from the application of IT throughout the economy (National Research Council 12).

The substantial impact of the IT industry on the economy has great implications for the relevance and importance of supporting innovation processes within this industry—which is what this thesis project addresses.

## Innovation In Context

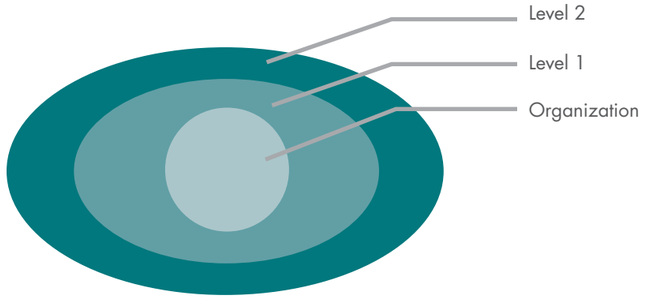
It is important to establish what is meant by innovation in the context of this thesis. The concept of innovation in the business world is often linked to newness or change (Guvenis 2). But it has long been defined in many different ways, and can mean different things in various applications. In 1953, Homer Barnett, an American anthropologist and thinker, defined innovation as "any thought, behavior, or thing that is new because it is qualitatively different from existing forms (Barnett *et al.* (1953) cited in Guvenis *et al.* (1989))."

Barnett's definition states that newness is not limited to a "thing" that can be measured but can also be a thought or an idea (Guvenis 2). This project considers thoughts and ideas as an extremely important part of the innovation process. They are an excellent starting point. But, they must be translated into forms that are visible and sharable so that they may be acted upon.

Another important definition of innovation comes from Everett M. Rogers in 1971. Best known for introducing the term "early adopter," Rogers defines innovation as "an idea, practice, or subject perceived to be new by an individual." He points out that "newness" can be expressed in the form of knowledge, an attitude, or even the decision to use it or not (Rogers *et al.* (1971) cited in Guvenis *et al.* (1989)). Furthermore, Gobeli and Rudelius in 1985 viewed innovation based on a company's competitive position. According to them, companies that innovate have competitive advantage (Gobeli and Rudelius *et al.* (1985) cited in Guvenis *et al.* (1989)).

Technological innovation, defined more specifically by Myers and Marquis, as a complex activity that proceeds from new concept or idea generation to solving a problem and then to utilization of a new product that has economic or social value (Myers and Marquis *et al.* (1969) cited in Guvenis *et al.* (1989)).

Innovation occurs within the social environment of an organization. An organization can be surrounded by two levels of social environments: the first is the industry or field in which the organization belongs, and the second level is broader in context and involves society or the general public surrounding the organization (Guvenis 4).



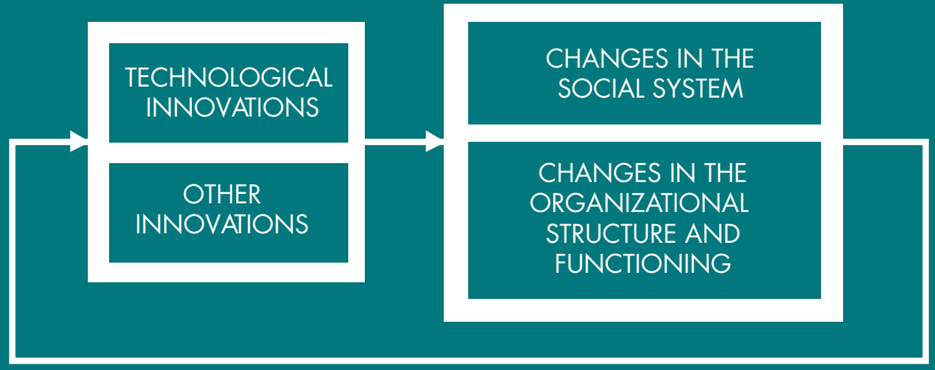

Levels of Social Environment Surrounding an Organization
Source: Guvenis (1989)



Innovation begins when an organization identifies that it needs to do more than what it actually does in order to achieve the change it envisions. This type of recognition may come from the first level environment (within the organization) or from external sources (Rogers *et al.* (1971) cited in Guvenis *et al.* (1989)).

Referencing Rogers (1971); Zaltman, Duncan, and Holbek (1973); Kanter (1984); Gobeli and Rudelius (1985); Petroni (1985); Ebadi and Utterback (1984); Ettlie, Bridges, and O'Keefe (1984), Guvenis synthesizes the following:

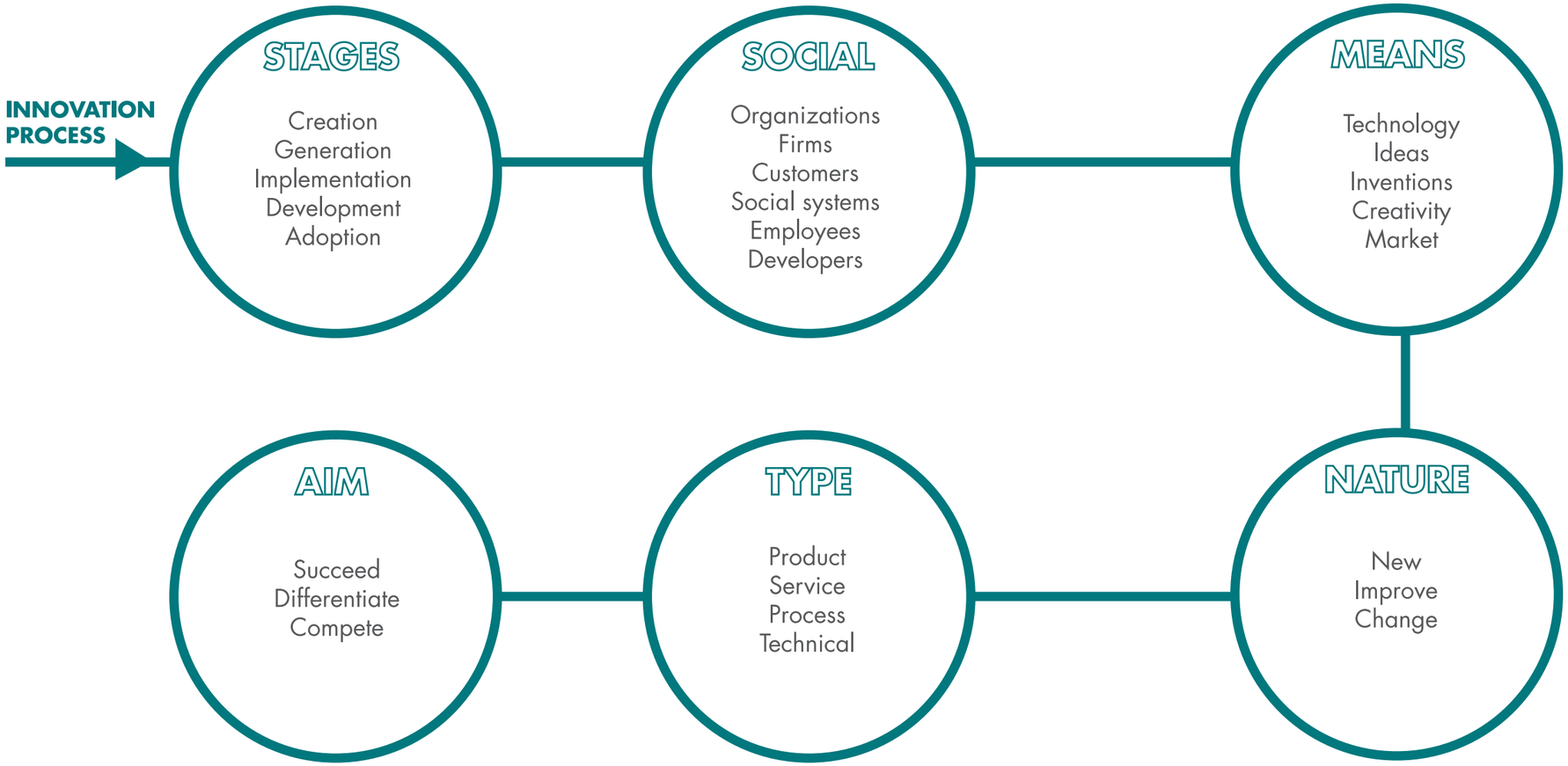
> The innovation can precede and cause social change, or it may be developed in response to needs created by social change. Technological innovation, being a subset of innovations in general, is not an exception to this routine. Together with social and organizational innovations they create a continuous and dynamic interaction (Guvenis 5).




Dynamic Interaction between Innovations, and the Social System and Organizations
Source: Guvenis (1989)

In 2009, researchers Anahita Baregheh, Jennifer Rowley, and Sally Sambrook from the UK contributed an article that sought to develop a multidisciplinary definition of innovation by conducting a content analysis of the various definitions that exist (Baregheh, Rowley and Sambrook 1323). The analysis was done on 60 definitions of innovation, from various disciplines. In this method, the researchers attempted to capture the "essence" of innovation in an integrative, cross-disciplinary definition (Baregheh, Rowley and Sambrook 1325). Their findings from the content analysis led to the development of the following model to capture the innovation process:

Source: Baregheh, Rowley and Sambrook (2009)




The researchers indicate that the 6 components of the model (represented in the circles) are not meant to describe only the process of innovation, but they also indicate various starting points within the process. The starting point on the diagram is determined by the backgrounds of the people involved (Baregheh, Rowley and Sambrook 1333).

The multidisciplinary definition of innovation that the research led to is the following:

> Innovation is the multi-stage process whereby organizations transform ideas into new/improved products, service or processes, in order to advance, compete and differentiate themselves successfully in their marketplace (Baregheh, Rowley and Sambrook 1334).

In summary, the key takeaways from the studies described in this section are that innovation, whether it is occurring on an organizational level or on a social level, is impacted by the environments that foster it. Both environmental levels are interconnected and influence each other. Another take away is that not only does innovation mean "invention" or something that has never been done before, but innovation in the context of this thesis can also mean processes and methods that people use to make a change in their environment.

# SOFTWARE TODAY

## *The Tech Industry's Paradigm Shift*

### The Shift

Competition in the computing industry has passed through a number of stages in the past few decades. The focus moved from fast and powerful computers in the 90s, to mobile devices that are slimmer and have brighter screens in the 2000s. Today the industry is shifting the focus to software design (Bilton, Tech).

With the focus on design, changes in software are the products of teams with a diversity of skill sets. In other words, the skill and creativity of a company's engineers and designers is what puts companies ahead. These skills cannot be copied easily because they address the soft side of technology or what is known to be the user experience (UX) (Bilton, Distrubtions).

To adapt to this shift in focus, major companies that rely primarily on developers are bringing in more designers to their workforce, and multidisciplinary teams that include designers and engineers are being formed for the creation of software products. This diversity in skill sets within teams is needed because it is no longer sufficient to create a product that has the technological capabilities, but does not meet user needs. Trying to adapt to this changing structure has proven to be challenging within most companies. Why so, is discussed in the following section.

### People and Processes

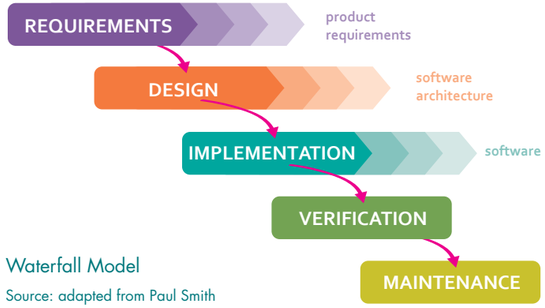
> *"The 1940s was a decade when fundamental questions were raised about [...] how human beings should interact with machinery. Do we construct machines that do what is technically feasible and adapt the human to their capabilities, or do we consider what humans cannot do well and try to construct machines that address those deficiencies?"*
>
> *Paul E. Ceruzzi*

Designers began having influence in the software industry in the 1980s and 1990s. In the beginning they approached software design in much the same way as they approached other materials they worked with, such as in product, print, and fashion design. In these other arenas, the designer was constrained by an expensive manufacturing process and had to have a completed design before the physical product, print, or garment could be made (Gothelf and Seiden 3).

Software provided designers with a different medium and posed new challenges. Upon entering the industry, the processes that designers used in creating software did not differ much from what they did within a manufacturing process, as they still had to use physical modes to duplicate and distribute their work (Gothelf and Seiden 3).

Today the playing field has changed. The Internet profoundly changed the distribution of software. Many software products have moved to online distribution methods. This shift has changed how both, designers and developers, approach software development. There is a lot of pressure on development teams to have faster and greater outputs because of ever growing competition in the industry (Gothelf and Seiden 3).

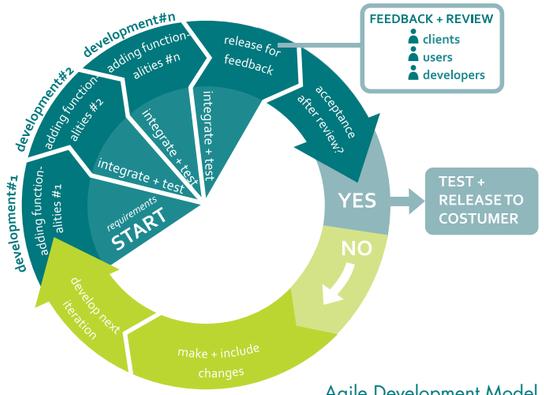
Not only is the design aspect a focus in software, but development processes have been shifting from older methods to newer ones as well. It has taken decades to perfect programming languages, and throughout its development different processes for creating software took shape. An older process, which is still in use today, is the Waterfall approach.




Waterfall Model

Source: adapted from Paul Smith

A newer and more iterative process, which is being utilized and adopted across the industry (and in many cases replacing the Waterfall approach), is called "Agile" development.



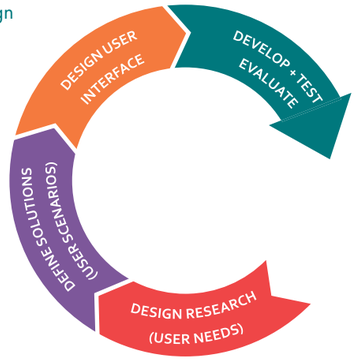

Agile Development Model

Source: adapted from a general understanding of multiple sources

User Experience (UX) Design on the other hand is also being adopted across these companies to make more user-friendly software, in response to user needs. UX uses design methodologies in conducting human-centered research that informs software interface design, as well as the service and the user experience of the software.

User Experience Design Process

Source: adapted from a general understanding of multiple sources




The problem, however, is that software has been created in a certain way for a very long time, and it is only recently that companies started to adopt UX design methods. Two important processes: software development, and UX research and design, have to somehow work together to create usable software. The integration has posed challenges especially when it comes to large companies where designers and programmers work separately most of the times. UX Research is slow in comparison with agile development sprints and release cycles. This change has posed questions about organizational structure, and the need for culture change within many tech organizations to adapt to a shifting industry.

But how do you change culture? How do you introduce more processes to an already process-heavy work environment? The reality is that many companies that are "developer-centric" sometimes fail to include their designer employees and vice-versa.

In other words: in many cases, different departments in a given company may use the most advanced techniques and processes to achieve efficiency in their production life cycle, but as Eric Ries puts it:

> The interconnections between them [the departments] are still mired in an antiquated industrial past (Gothelf and Seiden Forward).



## The Iceberg Analogy

A good way to describe the relationship between the various players in software creation was described by Kevin Lee, a Philadelphia based developer, in an interview for this project. Kevin explains the structure of software applications, especially mobile ones, in the Iceberg Analogy. In an iceberg, most of the ice is under water. The only thing you can see is the tip of the iceberg.

Similarly in mobile apps, the user can only see and interact with the interface of the app, or what is the tip in the iceberg metaphor. Whereas the underlying structure of the app, or what is called the "back-end" is very important and it's what makes the application work, and that could be the under-water part of the iceberg. So the end user—someone who uses technology but is not involved in creating it—does not care or pay attention to what is "below the water." Even though, both what is under the water and what is visible are extremely important and interconnected.

The analogy points out how both aspects of software—the technical and the interface—are crucial in software creation. Therefore, if any company strives to create products that are still relevant in this era, it must seek out and adopt methods that improve the tools and work environments for its teams to effectively collaborate.

Now that the IT industry is in context, I will explore how innovation, passion for building ideas, and pushing the boundary is pursued by the people involved within this industry.

# SIDE PROJECTS

## *Man and His Tools*

*"Tools are intrinsic to social relationships. An individual relates himself in action to his society through the use of tools that he actively masters, or by which he is passively acted upon. To the degree that he masters his tools, he can invest the world with his meaning; to the degree that he is mastered by his tools, the shape of the tool determines his own self-image. Convivial tools are those, which give each person who uses them the greatest opportunity to enrich the environment with the fruits of his or her vision. Industrial tools deny this possibility to those who use them and they allow their designers to determine the meaning and expectations of others. Most tools today cannot be used in a convivial fashion."* Ivan Illich

### Convivial Environments

As in the past, due to our globalized economy, fierce competition drives the market. Employees at companies small and large are pressured to keep up with the pace of innovation and to excel in performance and delivery. Many people caught in this cycle are deprived of their right to use their knowledge and expertise on passion projects and learning experiments, that bring them self-fulfillment, joy, and that lie outside of daily work tasks. It is thus important for this thesis to establish the relevance of "convivial environments" for fostering innovation. This theme is drawn mainly through the work of Ivan Illich.

In his book *Tools for Conviviality,* Illich wanted to offer a methodology that helps identify means (i.e. Technology), which have turned into ends (Illich 14). He argues that developing countries can bypass the industrial age by choosing to go directly to a post-industrial balanced mode of production, which the industrialized nations will be forced to switch to in the future. He submits a concept around the balance of human life, which he believes could serve as a framework for evaluating the relationship of people to their tools (Illich Introduction). What Illich highlights with regards to the relationship of people and their tools is still highly relevant to our time and in particular to the topic that this thesis explores.

Two terms he uses to support his argument for creating balanced modes of production that support the relationship of people and their tools, are "conviviality" and "tools." By "conviviality," Illich means:

> Autonomous and creative intercourse among persons, and the intercourse of persons with their environment; and this

> in contrast with the conditioned response of persons to the demands made upon them by others, and by a man-made environment (Illich 11).

In more simplified terms, Illich believes that when people embark on creative activities, they feel "joy." A society should be designed to have space for people to utilize their tools with creative freedom. By using the term "tools" on the other hand, he does not intend it to include only physical tools or cars and machines, but he also intends it to include environments, systems and infrastructures like:

> Productive institutions such as factories that produce tangible commodities like corn flakes or electric current, and productive systems for intangible commodities such as those which produce "education," "health," "knowledge," or "decisions." I use this term because it allows me to subsume into one category all rationally designed devices, be they artifacts or rules, codes or operators, and to distinguish all these planned and engineered instrumentalities from other things such as basic food or implements, which in a given culture are not deemed to be subject to rationalization (Illich 20).

If we relate Illich's concepts back to innovation, from the previous chapter, we can make the assumption that innovation occurs in environments that provide inhabitants with creative freedom and fulfill their sense of joy. Environments have a great impact on the relationship of people to their tools, be it physical or conceptual. Thus, environments that foster innovation should be designed for conviviality, which is what I define to be "convivial environments"—environments that support creative freedom and help people fulfill their sense of joy utilizing their tools.

## Experiential and Reflective Learning

Decades ago, renowned academic and user-centered design advocate, Don Norman, wrote about creating computing devices that conform to human needs and not the other way around. In his book *Things That Make Us Smart*, Norman goes into depth in analyzing cognitive artifacts, experiential and reflective cognition, as well as the three different types of learning (accretion, tuning, and restructuring). The key takeaway from Norman for this research is on experiential and reflective cognition in designing learning activities.

Experiential cognition is when information patterns are perceived and assimilated, and the necessary response is almost automatically generated without putting an effort and without delay. Experiential thought is reactive and relies heavily on a large storehouse of experience from which it retrieves its reactions (Norman 23). Experiential cognition may come from information that we already took a great effort to collect. In math for example, if you are asked "what is the sum of 5+2", you may unconsciously answer "7" (Norman 23).

Reflective cognition on the other hand is when a task requires considerable thought, decision-making, and planning (Norman 24). Unlike experiential cognitive processing, reflective cognition is slow and laborious. It is a type of thought that requires the ability to store temporary results, to make inferences from the stored knowledge, and make connections, and synthesize that information. The use of external aids helps the reflective thought process because they can act as external storage artifacts for the memory (Norman 25).

According to Norman, both of these cognitive modes are needed, and one is not better than the other (Norman 25). He emphasizes that these two modes of cognition don't capture all of our cognitive abilities, and they also aren't completely separate from each other, as sometimes they can be used at the same time for the same activity: one can enjoy an activity in an experiential mode and at the same time reflect upon it (Norman 26).



Furthermore, the relationship of these cognitive modes to designing tools (see Illich's definition of tools described previously), activities, or environments that support learning is explained in the following:

> Tools for experiential cognition should make available a wide range of sensory stimulation, with enough information provided to reduce the need for logical deduction. Similarly, tools for reflection must support the exploration of ideas. They must make it easy to compare and evaluate, to explore alternatives. They should not restrict behavior to the experiential mode. In both cases, reflective and experiential, the tools must be invisible: they must not get in the way (Norman 26).

What can be derived from this is that convivial environments should be designed to support both modes of cognition, experiential and reflective. Or in other words, environments that foster innovation should allow room for one to experience as well as reflect in the process of innovation. This can significantly improve the process by which we problem solve and create opportunities.

## Side Projects and 20 Percent Time

Many people have a natural urge to make, and create things to impact the environments they live in. But modern human lives are complex, and it is not always easy to satisfy that urge and to exercise ones creative freedom, especially within the structure of the modern work place (Tate 14). Successful or not, people work on side projects out of personal choice, and most likely for emotional reasons like self-fulfillment or to test a unique idea that they are passionate about.

In the last decade, the concept of side projects in the tech industry became popular. It was the reason for the rise of 20 Percent Time, hack days, and hackathons. Businesses started to take into consideration that in order for them to be relevant and to withstand the tests of the economy, they must provide their workforce with the creative freedom necessary to foster innovation (Tate 3).

HP and 3M were one of the first companies to provide their employees with time to work on side projects (Tate 5). Google, in 2004, set an example in the tech industry for popularizing this practice through their "20 Percent Time" policy, where employees were encouraged to devote one day per week for side projects of interest (Tate 3). Some of the notable projects that came out of 20 Percent Time at Google included AdSense, Google News, Google Reader, and most importantly Gmail—which gave rise to the practice in the first place (Tate 4).

After years of Google's support for side projects, however, its attitude over 20 Percent Time shifted until it finally put an end to it in 2013. Nevertheless, there are countless examples of companies that made their break through, out of tinkering with a side project. Whether or not practices promoting side projects succeed or die out at certain companies (for reasons far more complex than I can delve into here) the concept in itself is crucial for demonstrating how people need to have a balance in their relationship with their tools.

> *"If man is deprived of the use of this power, he becomes useless for work. Society can give shape to these personal activities, but it cannot appropriate the energy used on them for other tasks."* Ivan Illich

More importantly, the concept of providing creative freedom for employees led to the rise of what is known as Hack day, or hackathons, which is explored in the next section.

# HACKATHONS

*The Birth of a Phenomenon*


## History

Hack days, or hackathons as they are known today, are events often held for 24 hours straight where programmers are encouraged to turn their ideas into working software prototypes. The event started as an attempt to make 20 Percent types of projects, more affordable and they spread like wild fire amongst tech firms since they started in 2005 (Tate 60).

So how did it all begin? Atlassian, a startup in Australia, wanted to emulate 20 Percent Time but couldn't afford it. Its CEO Mike Cannon-Brookes thought of a compromise: Instead of 20 Percent on a weekly basis, he would let his employees have 20 Percent Time once (Tate 60). Atlassian went on to hold an eight-hour event where employees created a prototype and showed it to the company.

> It would be a pressure-cooker version of 20 percent time. Referencing the concept of one-day delivery, Cannon-Brookes called his event "FedEx Day— a mini, experimental, heavily bastardized version of Google's 20 percent playtime." He was thrilled with the results, he said, which included a task list maker, a flowchart generator, and tools for updating and debugging Atlassian's software suite (Tate 58).

Shortly after, another startup called JotSpot was going through a similar dilemma. It was seeking out a more affordable way to emulate 20 Percent Time. JotSpot's CEO, Joe Kraus, thought implementing a similar concept would not only empower his employees to pursue their passions, but would also boost workplace morale. Kraus learned about Atlassian's FedEx Day, and JotSpot went on to host its own version of Fedex Day, calling it a "hackathon." According to JotSpot, the impact of the event wasn't just great projects, but employee morale went through the roof (Tate 57).

So how did hackathons turn into a global phenomenon? It was started by Chad Dickerson, a yahoo programmer, who followed in the footsteps of Atlassian, and JotSpot. Dickerson organized the famous series of Yahoo! Hack Days, which were extended to the more common 24-hour hackathon, and are now emulated at tech companies like Facebook, Twitter, and eBay.

Dickerson believed that "deadlines catalyzed creativity." He also believed in what is called a "hacker ethic" which put an emphasis on programmer individual passions and the "cultural potential" of programming (Tate 61). Hack Day encouraged programmers to create a rapid prototype for the most essential parts of their ideas. The employees had a chance to put their ideas in front of managers, and managers had a chance to see potential ideas that could become actual projects (Tate 60). In other words,

> Hack Day was a way to say [to managers], 'You know what? In twenty-four hours I can build it and show it to you, and I think it's ridiculous that you want to spend eighteen months trying to throw a product together (Tate 63).'"

Most importantly, Dickerson organized the first Yahoo! "Open" Hack Day—an event open to the public. The Open Hack Day turned hackathons into a global phenomenon and allowed for people outside of tech firms to participate at the events (Tate 59).

To summarize the spirit of the phenomenon: hack days are a cheaper, condensed version of 20 Percent Time inside companies. Externally, they became a social, vibrant, innovation hub for the community of software creators. In both types, internal at companies and external, open to the public, the events offer an environment for people to innovate and give legs to their ideas. Participants get immediate feedback about their projects and learn from each other's skill sets (Tate 60). Hackathons also foster communities of practice—

> groups of people who share a concern or a passion for something they do and learn how to do it better as they interact regularly (Wegner-Trayner)

What the phenomenon offers is a gateway, or an open door for people to utilize their knowledge and expertise to express their creativity. It cuts through rigid structured work environments and equalized the playing field. At companies, hackathons allow employees to have a voice, and it empowers them. And, in external public hackathons, it creates a space for a community of practice to come together.

Hackathons, of course, have limitations as well. The two main down sides, which I explore in my research: these events foster innovation, but they: a) exclude non-technology voices, and b) lack a structure that supports effective team engagement. I will delve into the deeper level of the nuances at play in the next section of this research, but before doing so, it is at this point, that I would like to express how hackathons can serve as an entry point for supporting the shift that the industry is undergoing.

First, it is important to highlight that hackathons started as events held for developers/engineers. They grew out of the developer-centric cultures that are dominant at most tech companies today. Therefore, you will see very few designers attend these events, unless they are at a place where diversity and cross-disciplinary collaboration is encouraged.

> *"Hack Days separate doers from talkers."*
> *Chad Dickerson*

## How Do Hackathons Support Collaboration?

Collaboration is an essential characteristic of the modern workplace. A critical aspect of successful collaboration is effective communication. There are countless communication tools at a worker's disposal to effectively communicate with fellow co-workers. Tools like blogs, email, wikis, social networking platforms are some of the technologies used to communicate and collaborate (Turner, Qvarfordt, Biehl, Golovchinsky, and Back 841).

A yearlong study conducted in a small US corporation which focused on the communication technologies used within that company (Turner, Qvarfordt, Biehl, Golovchinsky, and Back 841). While the research analyzes the various tools used, how they are used, when and by whom, the main point to extract from this study for this project is on face-to-face communication.

Face-to-face communication comes from the media richness theory—a theoretical framework developed by Richard L. Daft and Robert H. Lengel. The theory defines the richness of information by its ability to change what a person understands from the information within a period of time. The longer a communication takes to transmit understanding the lower its richness, and vice versa (Daft and Lengel 560).

The yearlong study highlights key aspects of the media richness theory. It points out that:

> Face-to-face is the richest medium since it provides rich feedback, multiple cues such as intonation, body language variety, and a personal focus. The media richness theory predicts that communication will be more effective face-to-face than through other media (Daft and Lengel *et al.* (1987) cited in Turner, Qvarfordt, Biehl, Golovchinsky, and Back *et al.* (2010)).

A series of interviews for the study were conducted with regards to the media choice:

> From the interviews we found that face-to-face is a preferred communication channel in the studied organization. This is not surprising, as all media selection theories predict a preference for face-to-face. Respondents commented that face-to-face communications is good for relationship building, ideation, problem solving, and for keeping some issues "off the record." Thee comments resemble findings from other research. A recurrent comment in both the survey and in the interviews was the immediacy of establishing a face-to-face communication: "it's a very small office and I just walk over and talk to somebody, because that is immediate." (Turner, Qvarfordt, Biehl, Golovchinsky, and Back 847)

The weaknesses of face-to-face communication, as the study shows, is that in terms of efficiency it is more time consuming. Another weakness is in distance and location: people need to be at the same place and time in order to communicate.

In conclusion, the hackathon space fosters an effective and rich communication medium: it provides people with the opportunity to utilize face-to-face communication for effective collaboration because they are at the same place, at the same time.



# TECHNOLOGY FOR EVERYONE, BY EVERYONE

## *Why Technology Development Should Not Be Exclusively Left To Programmers.*

> *"[...] As such technologies come to characterize the future of the way we live and work, the people programming them take on an increasingly important role in shaping our world and how it works [...] That's why this moment matters. We are creating a blueprint together—a design for our collective future. The possibilities for social, economic, practical, artistic, and even spiritual progress are tremendous."* Douglas Rushkoff

Considering that digital technologies are programmed, then, they are undoubtedly biased to the people with the ability to write the code. Douglas Rushkoff argues in his book *Program or Be Programmed*, that if the majority of people don't learn how to program, or at least understand that there's code behind the interfaces of the software that they use, then they risk continuing to be at the mercy of those with programming skills, the company's that pay them, and the technology itself (Rushkoff, 1488/1762).

The Internet and computers have given us access to the writing, and sharing of information. But according to Rushkoff, the underlying capability of the era of computers is programming, something that the majority of people who use technology don't know how to do. We are satisfied with using the software that others have programmed for us, and most children are taught how to use these programs but not how to make them, which gives them access to what others are capable of but not the skills that empower them to create these technologies for themselves (Rushkoff 133/1762)

This idea is significant when considering how integrated the technologies we use are in our daily lives. Technology has its biases. If we understand these biases and become conscious and aware of how the technologies we use are made, we can reverse the trend. We can actually become active participants in shaping the programs we use rather than being mere observers and relying on a very small elite; an elite that has gained access to the tools that empower them to make those decisions for us (Rushkoff 125/1762).

The key take away from this theme is that it's crucial for us to begin the conversation around becoming participants in making a technology that has more human voices than the other way around. The reason for this being a critical moment is that digital technologies are built on complex systems and networks. This means that in the same way that decisions made in the past still influence the programs being built today, the programs built today are shaping what the future technologies will look like for mankind.

But how do we join the movement? How do we begin this trend? I consider external hackathons, ones that are open and free to the public, to be an entry point for other disciplines to join the conversation as well. They are an opportunity for others to have a voice in the community of makers. Not only should designers and engineers take advantage of improving their processes together in this forgiving environment, but also others should participate and take advantage of an event that fosters learning along with a supportive community.

> *"Tools foster conviviality to the extent which they can be easily used, by anybody, as often or as seldom as desired, for the accomplishment of a purpose chosen by the user. The use of such tools by one person does not restrain another from using them equally. They do not require previous certification of the user. Their existence does not impose any obligation to use them. They allow the user to express his meaning in action."* Ivan Illich

Image credit: UArts Applied Design Lab 2012

# A DESIGN APPROACH TO TECH INNOVATION

## *How Design Methods and Tools Can Foster Tech Innovation*

The shift that the tech industry is undergoing is a tremendous and slow effort. Companies are doing massive culture change to their corporate structures to adapt and continue to be relevant. I believe hackathons offer an entry point to supporting this shift. But first, the down sides of these environments must be addressed.

These events could be more inviting to non-tech voices, and they could also use a structure that would support effective team engagement and collaboration. This thesis utilizes design methods and tools to accomplish this task. It seeks to introduce a design thinking approach to tech innovation. Design thinking is

> Innovation powered by …direct observation of what people want and need in their lives and what they like or dislike about the way particular products are made, packaged, marketed, sold, and supported…[It's] a discipline that uses the designer's sensibility and methods to match people's needs with what is technologically feasible and what a viable business strategy can convert into customer value and market opportunity (Tim Brown *et al.* (2008) cited in Gothelf and Seiden *et al.* (2013)).

Design thinking is an important concept to software design. It encourages product teams to consider a holistic approach to what they create (Gothelf and Seiden 6).

### What is a Charrette?

One of the design methods utilized in this project is called a design charrette. A charrette is a design method in the form of a workshop. It offers a space for collaboration, for designers and others alike to explore, share, and generate ideas in order to solve design problems. The format of the charrette is meant to help ideas flow and encourages participants to build on each other's ideas. It is usually low tech, and a moderator is present to help teams run through the process. The technique encourages teams to generate many ideas, and narrow down, ending in the creation of quick and rough sketches/prototypes for their design solutions (Martin and Hanington 58).

### Thesis Questions:

I set out to ask, and answer the following questions:

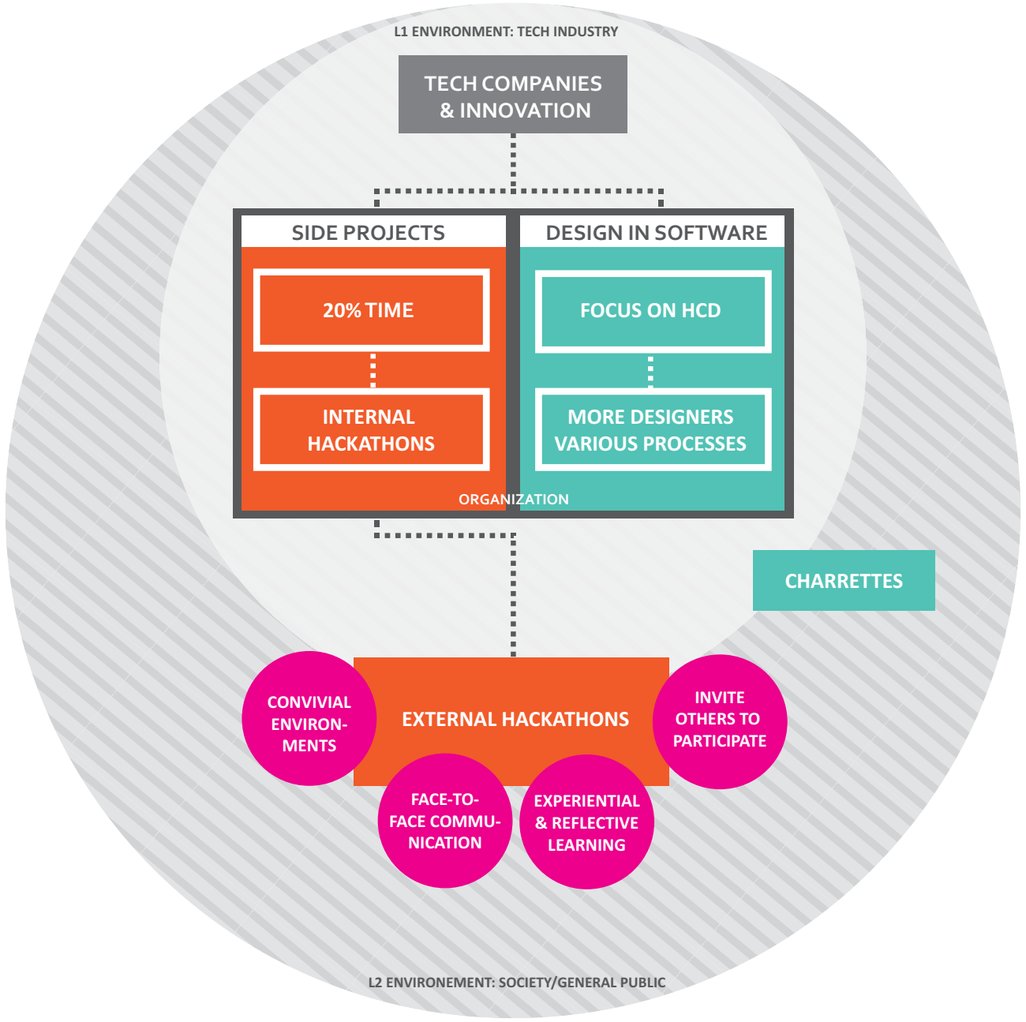
*How can design help teams, especially multidisciplinary teams have an engaging learning experience?*

*How do you invite other disciplines to events that are dominated by one community of practice?*

*How might design methods and tools play a supporting role in tech innovation events like hackathons, an environment that supports the creation of many of the technologies that impact our daily lives?*

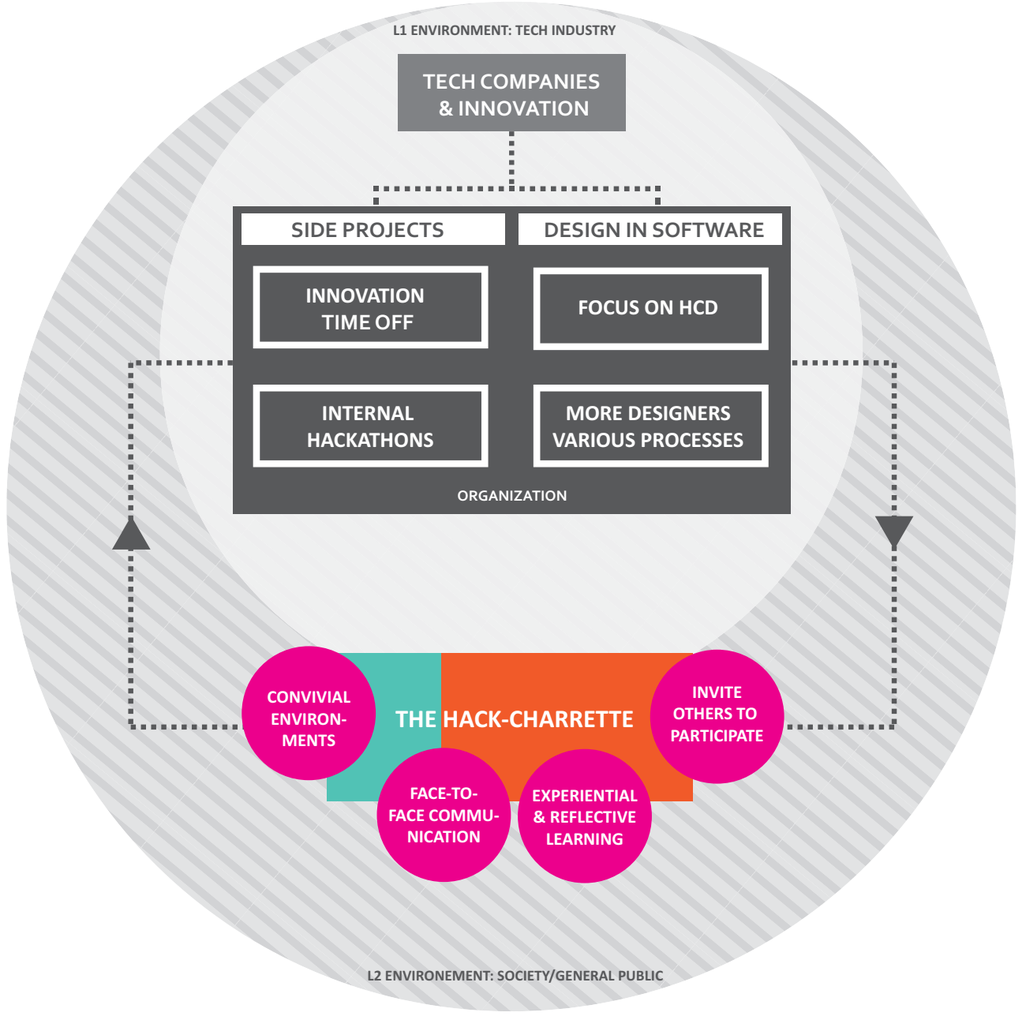
## THESIS STATEMENT

SOFTWARE COMPANIES THAT STRIVE TO BE INNOVATIVE NEED TO PROVIDE CONVIVIAL ENVIRONMENTS, SUCH AS HACKATHONS, THAT SUPPORT MULTIDISCIPLINARY TEAMS. DESIGN TOOLS CAN FACILITATE METHODS THAT WOULD FOSTER ENGAGEMENT WITHIN THESE ENVIRONMENTS. THESE METHODS SHOULD BE DESIGNED FOR EXPERIENTIAL AS WELL AS REFLECTIVE LEARNING. METHODS SHOULD ALSO CONSIDER THE CULTURE OF EACH COMMUNITY OF PRACTICE INVOLVED WITHIN THE TEAM.




Visualization based on research showing current model for how companies approach innovation.
● 4 themes identified which could potentially be supported by hackathons.






Visualization of the Hack-Charrette thesis supporting a future model.
Incorporating design methods and tools into hackathons can support tech innovation processes, enhance team engagement and finally support the 4 themes. This model will create a “continuous and dynamic interaction” between L1 and L2 environments and innovation.

# II. ENTER THE HACKATHON

# UNDERSTANDING THE HACKATHON ENVIRONMENT

*How Does This Thing Work Anyways?*

As indicated in the past section, there are two types of hackathons that grew from the spread of the phenomenon: a) internal hackathons, or what is also called hack day (events that happen inside tech firms), and b) external hackathons, which I define to be hackathons that are open to the public, and mainly occur in the form of "civic" hackathons, or within educational environments like high school or college hackathons.

I chose to apply my thesis in the context of external hackathons, with a focus on events occurring within educational environments.

## Event Structure

One major observation about hackathons, is that they lack structure. The events are this way by design. It all comes from the idea that, at a hackathon you are supposed to have full creative freedom, so imposing any small amount of structure may cause participants to act negatively. Since my focus is on college and high school hackathons, below is a timeline of how a typical 24-hour hackathon may look like:

2pm to 1pm Saturday: Registration, receive swag, event kick-off—opening remarks for getting participants excited about the event; Sponsorship information: resources, and prizes
1pm Saturday: Begin hacking
Meals and snack breaks throughout
11am Sunday: Final hours, prepare for demo
1pm Sunday: End hacking
1pm to 2pm: public expo/pitching your hack to judges
2pm to 3pm: judging, and finally, prizes.

This format may seem simple, but conducting interviews with organizers of college and high school hackathons proves them to be otherwise.

Meet Katy, a college hackathon organizer. Below is Katie's partial to-do list:

Logistics:

- Lead team of organizers and divide tasks
- Find sponsors
- Secure venue and technology
- Order food+ swag
- Set up web site
- Market event + get people to sign up
- Recruit Mentors
- Plan hourly tasks and shifts
- Financial
- Prizes
- Judges
- Organize workshops around the clock

Things to Keep in mind:

- Get people excited about CS + Tech
- Encourage learning over winning

It is no doubt a great task to take on organizing a hackathon, especially when you are expecting anywhere between 500-1000 students at your event. A lot can be learned from hackathon participants-turned organizers. These are students who are passionate about what the event can foster. They themselves had a great experience at a hackathon at some point and decided "hey, I'm going to bring this to my school." Interviews with organizers exposed that the real goal for most of them is to get people excited about learning computer science—a field described by a compute science graduate, as well as a professor in computer science, to be stuck in its academic roots.

Furthermore, hackathons support learning because you can learn by doing. And projects are team based, so you can learn from the people around you as well. The events also invite mentors who are professionals in the tech field. The mentors volunteer their time to come help out and provide feedback to students.

## Profile: Organizer

Mayank Jain is a student at the University of Illinois where he majors in computer science. He attended his first hackathon on campus at UIUC mainly because he walked by and saw ads for the event, in his own words:

> "I had always been interested in technology and entrepreneurship, and realized that this was a perfect combination of the two."

He is now very active and involved in hackathons and thinks that they have a great incentive in the possibility of creating your own startup, and making your projects come to life. He also sees that the core value in these events is that they offer a great method for learning how to solve problems through making and creating your own application.

Prior to entering hackathons, Mayank thought the event was about "hacking," or breaking into something until his first hackathon where it turned out to be the exact opposite. He described it to be about

> "building something new rather than destroying something that can be exploited."

Excited about the opportunities lying in these types of events to make something good, Mayank co-founded a non-profit organization called Pilot. Pilot hosts hackathons across the country for high school students. Through Pilot, Mayank hopes to expose beginners to programming and to provide them with a learning environment to test and create with technology. One way that Pilot tries to attract newbies is by pushing away from using the word "hackathon" or "hack" in their marketing, and focusing on the learning aspect of the event.

For organizers, like Mayank, organizing an event and helping teams form is one thing, making sure participants have a positive experience and get the most out of it is another. There is the stress of taking care of logistics like securing a venue, contacting sponsors, ordering prizes, taking care of technology, security, food, and containing the chaos of 500 high school students.

## What Do These Events Look Like?

The following images depict what a typical hackathon looks like.


*From Observing the events, I found many teams just want to start coding immediately.*


*Depending on the size of the event, there might be a final expo. This image depicts the PennApps 2014 expo which generated over 200 submissions.*


*The social aspect is incredible as well. For high school hackathons for example, telling students they can stay up all night with friends just makes the event much more appealing.*

The images show how hackathons are vibrant, social and have a collaborative spirit. However, their simple structure can easily be dismissed. There is a lot that goes on in a 24-hour time span.

There are defining moments throughout the event, like when a team is deciding on an idea to pursue, when they are trying to make the code work, and even when they are pitching their hack to the judges. The nuances and interactions that take place during these moments can make or break a team project.

Some organizers, like those of Pilot, try to divert attention away from "winning" or the competition aspect of the event. This is excellent but it is sometimes hard to achieve at college hackathons, for example, where prizes and sponsors recruiting the future IT workforce offer an incentive for students to join and compete.

Moreover, a bad experience can lead a participant to reject the whole concept behind hackathons rather than take advantage of what the event has to offer. The next section explores the nuances at play.

# UNDERSTANDING THE HACKATHON EXPERIENCE

*The People*

How do you measure experience—something so qualitative and best measured by the person experiencing it? Using methods such as active participation, observation, and surveys, I developed concepts that helped me qualitatively measure hackathon experiences, and thoroughly investigate areas for improvement.

The concepts include: the impact of the social environment, the degree of which the environment supports learning, the impact of incentives, the size of the event and sponsorships, the ways in which teams form, how open ended the prompts are for the creation of hacks, the importance of constraints like time, the cultural dominance of one community of practice over another and its impact on other disciplines joining the event, and finally the impact of judging methods on the competitive aspect of the event.

This section describes in detail the methods used, the research findings and synthesis of the concepts for measurement.

## Profiles: Participants

Name: Kevin
Age: 21
Background: Electrical Engineering major
Attendance: 2 Hackathons
Team: meets teammates at events

Kevin's first hackathon was incredible. Initially, he wasn't sure what it would be like and was ready to leave 2 hours in because he couldn't find a team to join and felt awkward. But on his way out as Kevin explains it: "holding my arduino, I bump into another guy holding his arduino. He looks at me, I look at him, then he says "Come on!" and we team up. It was amazing."

Kevin joined a hackathon for the second time, excited to meet new people, socialize, be creative and make something awesome using his tech savviness. At the event he was paired up with a team of 3 other participants whom he did not know and who came from different disciplinary backgrounds. They all had a great deal of skills in their personal toolbox, but the 4 of them had to collaborate and create something as a team within the given timeline.

A few hours in, two of the team members were paring and not cooperative with the group's activity. Collaboration was difficult. Those two team members disappeared after a short while and went on to work alone. Kevin was frustrated with this but still wanted to go through with the experience. He and his remaining teammate continued to work together but ultimately did not submit anything for the competition. Kevin lost his excitement after his team collapsed and wished for a better experience. He regretted that he didn't have something he could show off.

Name: Luis
Age: 19
Background: Computer Science major
Attendance: newbie
Team: comes with teammates/friends

Luis and his 2 friends joined up a hackathon after hearing about it from their friends. They were very excited especially since they were going in together, they knew that by putting their brains together they will come up with something great. When asked by another hackathon participant during the first few hours of the event what were they building, Luis explained their brilliant idea and the participant was excited and cheered them on.

Around 14 hours later, and with very little sleep, Luis and his teammates were on their 5th idea. They couldn't decide on what they would like to build and so they would start an idea and give up on it after a few hours. They lost their energy halfway through the hackathon and their excitement to learn and create something new faded away.

From Luis's experience, I learned of another aspect, that is: the process used by most is not one that suits teamwork. This

demonstrates how hackathons, not only reflect the status quo of the tech industry (i.e. demographics: hackathons are known for having more male participants than female), they also reflect the status of computer science education in general. According to 3 computer science graduates interviewed for this thesis: most computer scientists are not taught how to work in teams like they are asked to do at hackathons, or more importantly like they will be required to do when they join tech firms.

Computer science majors are taught how to focus on complex problem solving through programming, alone with their machine. That is why hackathons offer the space where they can actually collaborate with other people. But the process still doesn't work for team-based projects. Jumping immediately into the code may seem productive at the beginning but if teams don't get lucky, they will have to back track at some point to figure out the big picture.

This is unlike a design thinking process, where collaboration is supported by the methods and tools used. A designer has to first explore the big picture, and then narrow down to the details—a process of divergence and convergence. The design process if used by teams at hackathons, may end up saving a lot of time in the long run, and teams can actually spend most of the time programming.

## Experience is Qualitative:

## Love/Breakup Letters Survey

To understand the experience of hackathons further, I participated in two hackathons myself. My own participation helped me understand hackathons better—for sure because I was consciously studying my every feeling. However, my own experience wasn't enough, and I wanted to capture stories from other participants. So I designed an Online survey, and put it out into the world, disbursing it via the connections I made throughout my research.

The survey was titled Hackathon Love/Breakup Letters. I took a design method that asked participants to write a love letter to a product, and changed it to that of an experience.

The survey asked participants to personify a hackathon experience and write a love letter first, and then a break up letter. Prompts were given for each question. A total of 68 surveys were started. Most filled out the demographic multiple choice, or check-box questions but when it came to writing about their experience exited the survey. 18 survey-takers wrote a love, and/or breakup letter. This was a learning point for me in designing future surveys. It is risky to have your main questions be in text box format. But luckily I still received responses that further supported my learning. Some responses came from organizers as well.

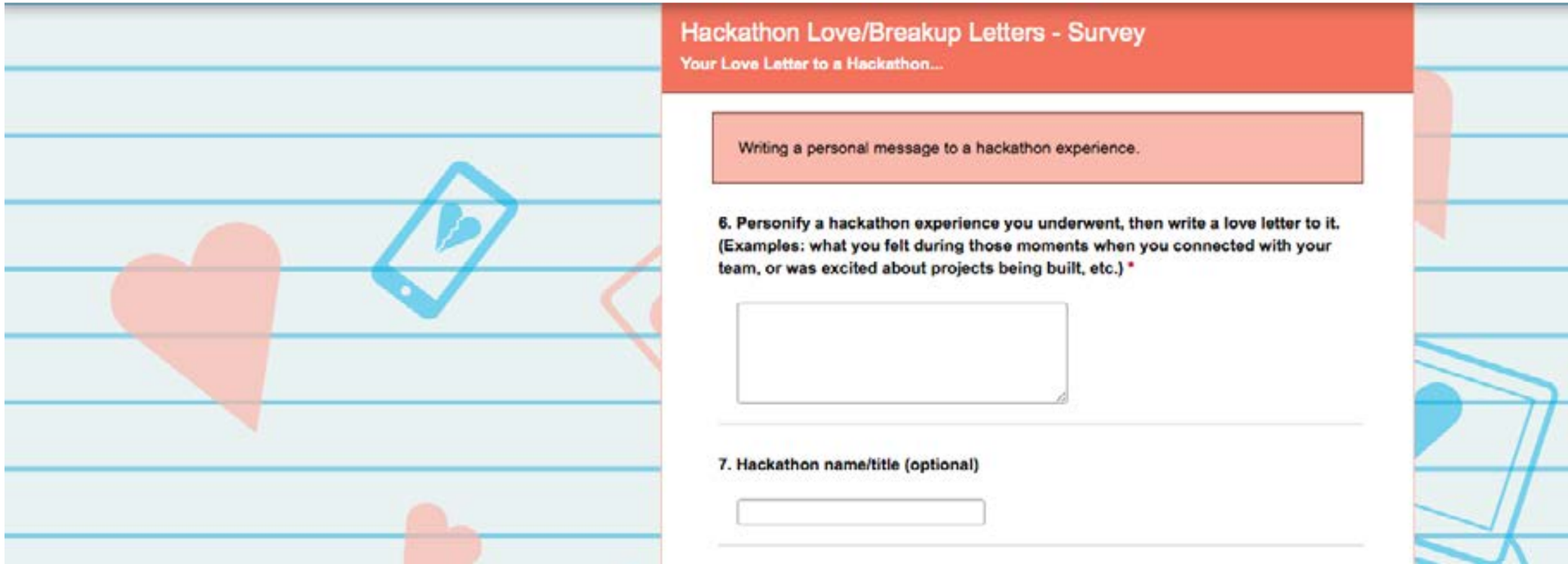
Interestingly a majority of responses were dedicated to a first hackathon experience: 13 out of 18 in the Love Letters and 9 out of 16 were dedicated to a first hackathon in the Breakup Letters.

Sample of responses to the Hackathon Love Letter:

*"Dearest PennApps, Thank you for being my first. I had a lot of fun. I loved how you just opened up and let me in. You were so welcoming. I really felt like I was in the place I belonged when I was inside you. You taught me several valuable skills that I look forward to applying when I meet others just like you. I really wish I had more then 40 hours with you. I look forward to seeing you later this year. I will never forget you. With love, R."*

*"My first ever hackathon was one that I organized at my high school, and I fell in love from that moment on. Seeing so many students learn so much, so quickly was like watching an artist pick up a paintbrush for the first time, before then going on to paint a masterpiece! It was thrilling to watch, exciting to organize, and overall one of the highlights of my time in high school!"*

*"Very difficult to get along. You would let me come up with ideas and when you did you ended up not working for hours. Once we got the idea working, we went ham on you and 15 hours later (out of a 36 hour hackathon) my baby was born. Best 15 hours of labor and the best feeling when you put your heart and soul into something to create something out of nothing."*



*"A Hackathon I organized and attended was CodeDay Denver, which was an unforgettable experience. Although I didn't expect it, we had a lot of inner-city kids show up who had no prior experience programming, and although it didn't result in fancy hacks, it resulted in a large number of students learning programming and building skills for the future. It was also fun pulling an all-nighter, and watching ourselves slowly lose sense."*

*"At Blueprint, my team just started randomly talking, after we sat down at the same table. We called ourselves "Team Serendipity," because we just got along. I loved that."*

*"Felt good when I won. Also felt good when I was learning alongside experienced coders."*

*"I f***ing love you."*


The love letters word cloud above highlights the words used to describe feelings towards a good experience. Words include love, fun, serendipity, good, bliss, to name a few. The descriptions provided by the survey takers help in measuring participant experience.

Sample of responses to the Hackathon Breakup Letter:

*"I understand that you want to keep working at 4 am, but I really have to sleep. It's over, I'm sorry."*

*"You didn't get along with me and did your own thing. We argued over simple implementation and ended up giving up because nothing was working"*

*"I wish I could've finished."*

*"No food? Not cool."*

*"I'm sorry it didn't work. They said long distance was hard. They were right. I feel like I haven't talked to you in forever."*

*"At MHacks III, my team struggled on which idea to settle on. It took quite a lot of debating before we came to agree, and that was definitely my least favorite part."*

*"Dear hackDuke, What the heck people, don't be jerks and hoard food, harass others, be disrespectful to faculty and whine about things. Sometimes, I feel like the hacker community has a few righteous and privileged jerks who ruin the awesome for everyone."*

*"A minor issue we ran into was that the attendees, without permission, took nerd guns from the office and started playing with them. It was fairly irritating, but we managed to clean everything up."*

*"No one on our team really cared about finishing or even the idea. One guy literally left the hackathon and I haven't seen him since."*

*"When the other members kept on trying to mix up the presentation, we weren't coming to agreement and it annoyed me."*


In a similar matter as the Love Letters, the Breakup Letters word cloud also highlights the words participants used to describe frustrations they had with a hackathon experience.

In conclusion, considering that experience is something intangible, the descriptions provided by participants utilized the writing medium to communicate good as well as bad hackathon experiences. In both experiences, good and bad, the participants interactions with others, including teammates were dominant in forming the event experience.


# KEY FINDINGS

## *Research Synthesis*

Part of my design research was done through observing and participating at hackathons. My very first observations and understandings of hackathons started as a mentor in PilotPhilly as well as a participant at Philly Hacks Cancer. The latter was a very small event in comparison to high school or college hackathons. But it allowed me to gain my first insights into the nuances at play when a team comes together in a shared space to create something they deem valuable.

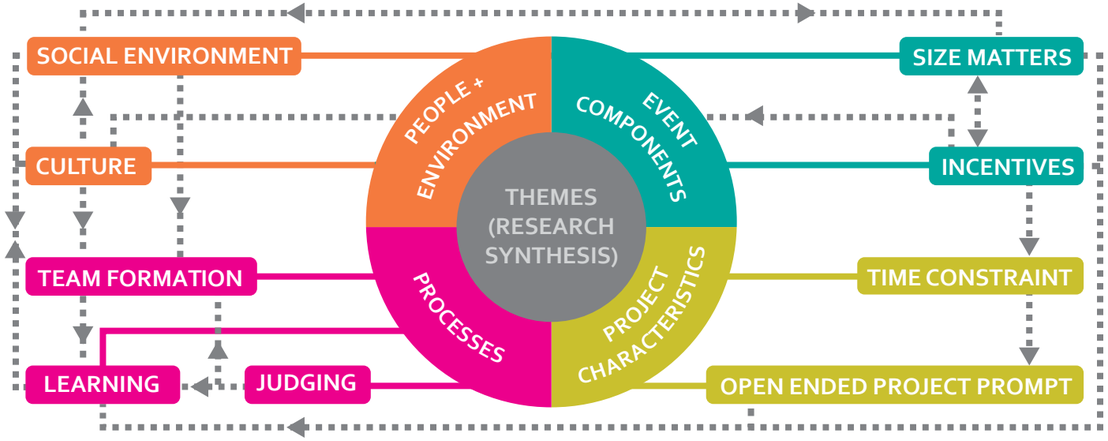
Team dynamics play a huge role in the experience at these events. At Philly Hacks Cancer, a civic hack, there was a clear goal to the event: create apps that may support cancer patients and care givers. While this type of hackathon is not the focus of my thesis, I still gained insights into the overall experience. Attendees had all come because they cared about the cause that the hackathon sponsors. So there was already a shared goal and bond that helped my team collaborate effectively. Participants were also very mature and all working professionals, so there was already experience in working in teams.

PilotPhilly was different. It was more like a pure format of hackathons in terms of how open ended it was. The event was my first gateway into what really goes on at hackathons. It is the opening hackathon of hackathons for many of the attendees as well. I was there as a mentor, and spent a lot of time doing fly on the wall observations of team interactions along with trying to absorb what was going on around me.

Some teams were more receptive than others to receive advice from me in their ideation phase. One particular interaction stuck with me: just like Kevin from the previous chapter, 2 hours into the event, a student was packing up his bags. I asked him why he was packing, and he answered "this isn't what I expected. I'm a designer, I come up with ideas. I don't know how to code like them. My mom is outside waiting to pick me up."

## Findings and Design Opportunity

Up until now, I have laid out the hackathon environment, and the experience based on my research and observation. The findings have informed my decision to focus on the nuances that shape the hackathon experience. But first, below is the synthesis of the themes from my research:




Connections between themes, and potential impact on each other

**SOCIAL ENVIRONMENT:** hackathons provide the social atmosphere that invites people to its space. It is an opportunity to get people excited about computer science.

**CULTURE:** Fitting in, may feel difficult for people coming from other disciplines as the overwhelming majority come from similar backgrounds. This leads to lack of diverse voices in the technologies that are created. It also reflects an industry dominated by males.

**TEAM FORMATION:** teams form in various ways. Some come with their teams already formed, some meet people through social networking sites and then form into a team, and others come to the event without a team, and try to join a team there.

**LEARNING:** people learn by doing. The hackathon provides a safe place for experimentation where it's OK to fail

**JUDGING:** presentation skills are needed for making pitches to the judges. Some ideas may not be presented in their best light. Or worse, some ideas may not be presented at all from fear of being judged.

**SIZE MATTERS:** the larger the attendance, the more publicity and sponsorship it gains. Competition grows more dominant, and takes the focus off learning. This may also intimidate new attendees.

**INCENTIVES:** there are incentives like prizes, and opportunities for recognition. These give people incentive to continue with their projects, but they also increase the competitive spirit



**TIME CONSTRAINT:** helpful for turning an idea into something visible—a minimum viable product.

**OPEN ENDED PROJECT PROMPT:** participants aren't given prompts. The only helpful constraint is time. Teams may get lost trying to navigate their project and narrowing down their focus, especially with the lack of structure.

The summary of research findings shows an opportunity for utilizing design methods and tools to improve the hackathon experience. My AHA! Moment came in analyzing the nuances of the experiences collected via observations, interviews, and surveys where I saw the high impact of team dynamics on the overall experience of participants.

If people have a good experience collaborating with their teams, than it's a domino effect: it improves the overall learning outcome for participants. It would also take the competitive edge off a participant's mind, especially those new to the event. It would refocus the event on learning, and collaboration. It would also become more relevant and inviting for other disciplines to join.

> *"I think space can be adapted to, but if the people there don't mesh well together then you can lose the benefits of bringing people together in a shared space, when they could just be learning Online or in a different format."* *Mayank Jain, Organizer*

## Design Intervention Planning

With a much more narrowed focus, understanding the nuances of the experiences was crucial to how I planned my design intervention. As I mentioned earlier, I immersed myself in researching hackathons in two modes:

1. **As participant, and**
2. **As mentor**

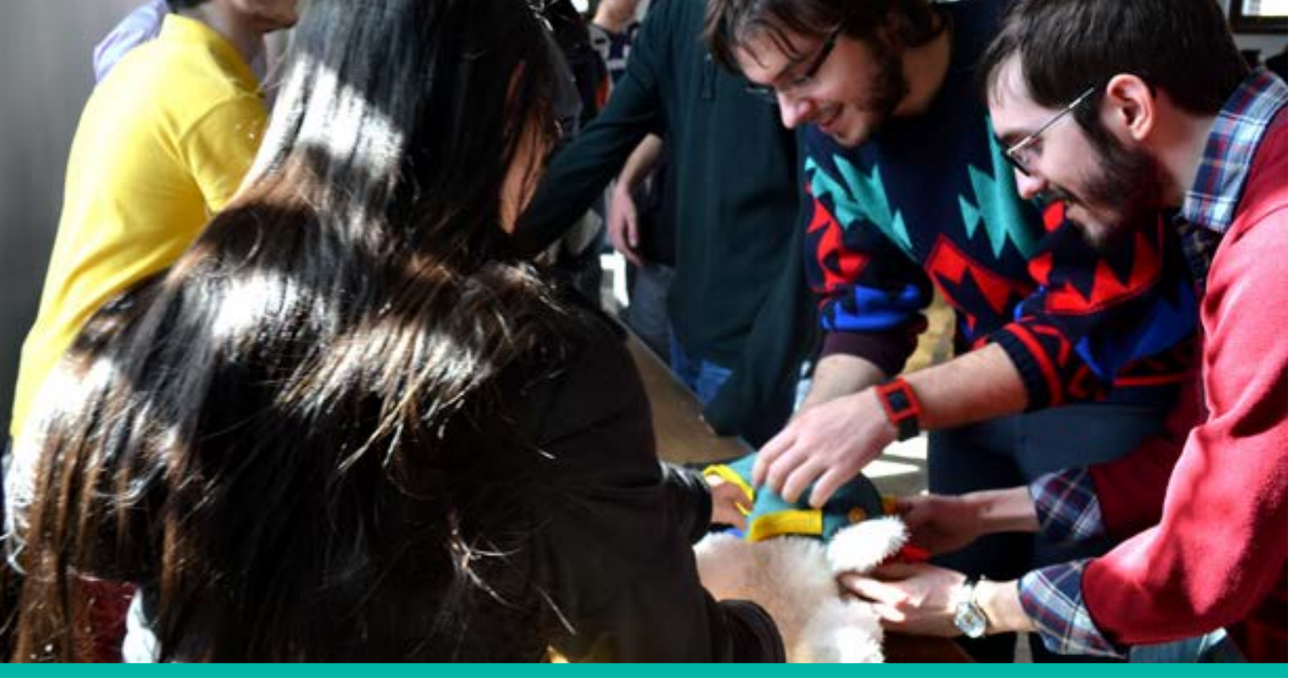
When thinking about facilitating a design intervention that will foster engagement and learning within teams, both modes had to be considered. Based on my findings, key points that needed to be considered for my design were:

**a) Focus on "invisible" facilitation when intervening as participant: considering that these events lack structure, it is important to keep teammates from believing there is a constraining process they have to follow. And,**

**b) Utilize design tools for facilitating design thinking activities that cater to hackathon participants. This can be done through using rough brainstorming and paper prototyping methods, and using language that is void of jargon and speaks to a diverse audience**

I had multiple opportunities within my project timeline, to join additional college and high school hackathons, and put my thesis to the test.

# III. THE CHARRETTE HACKS THE HACKATHON



# PROTOTYPE + TEST

*Parallel Prototyping*

## PennApps: Design Intervention in Participant Mode

PennApps is one of the largest college hackathons in the U.S. It is hosted once a semester at the University of Pennsylvania in Philadelphia. Over 1,000 college students from across the country, some international, attend the event each semester. Like most college hackathons, it is student run. It has managed to build a strong base of sponsorships since it started in the fall of 2012.

In Spring 2014, I recruited two fellow designers to attend the event with me. The goal was to introduce other designers to hackathons. We were all designers, but we had different skill sets and came from different backgrounds. I went in anticipation that I will be on a different team, but my fellow designers and I were teamed up with a bioengineering major from UPenn.

I had never worked with the designers on my team prior to the event which was important for me in terms of testing my intervention. Our team formation represented one of the common methods of teams forming at hackathons. Most of my team members had met before the event, except for one who experienced the hackathon like many others. She met her team at the event.

We had the opportunity to work in a hardware hacking space called "Stitch Fest." I understood then that hardware hacks are becoming increasingly popular and present at hackathons, which are mainly events for creating software. Even PennHacks, a hardware hacking event recently integrated into PennApps.

The goal of my design intervention was to facilitate an idea generation session with my teammates in a simple seamless format—considering experiential and reflective learning methods. I took with me paper rolls, sticky notes, markers, and tape—facilitation tools to help externalize our process, especially for reflective activities like idea generation. We selected a space, and I started hanging the paper on the wall. When my teammates saw me doing that, they were excited about the materials I provided and started organizing the hardware for our hack and labeling it.

I simply started to give an idea for what we might be able to start with. For example I suggested we think of areas or themes first, and would immediately follow my suggestion with an invitation to their thoughts on that suggestion. This proved to be key for guiding through seamlessly. The process ended up being guided not just by me, but by all of us. It was important for the process to be flexible but at the same time provide a guiding light.

We utilized design methods like the KG method, affinity mapping, personas, and scenarios. The facilitation tools offered a space for the push and pull between teammates to discuss concepts and ideas. It was difficult at the beginning to get team members to draw or write their ideas down, but after seeing the value of it, it turned into a communication and reference tool. Every thought was visualized and placed out in the open.

When we finally narrowed down and decided on an idea to pursue, I encouraged the team to consider our constraints in terms of skill and time. We had two who knew how to code, and two of us who knew how to design, visualize and make things.

The hardest part was getting the code to work. As a team, we had a solid idea, but we were almost hesitant to submit our idea at the end because it simply wasn't working. However, our engineer decided we should submit it anyways, and we did. We stood waiting for the refusal for our submission to come because it wasn't functioning, but it never came! We were given a space to showcase our hack at a final expo, and we didn't have something functional to show but we had a solid idea and when people came around to our table our pitch was strong and people were still excited about the concept.

Following the experience, I asked my teammates to evaluate the impact of the process and methods we used at the beginning of the event and they thought it was very helpful especially having our ideas on the wall and the ability to reference them The engineer stated that engineers typically blurt their ideas out and jump into working on them, but this process helped us stay organized and explore different options.

In conclusion, PennApps was an attempt at implementing a prototype that was so loosely planned, and open for change. I was able to practice, as a designer, speaking in a way that speaks to our diverse group. I refrained from using jargon that I knew none of my teammates would understand. I simply stuck with explaining the process in plain natural language. I even tried to get the ball rolling by turning a few simple steps into an agenda. And most importantly, since the facilitation tools became part of our conversation, our working space was transformed—it was a convivial environment by all means. We owned the space.

## Philly Codefest: Design Intervention in Mentor Mode

Philly Codefest held at Drexel University is younger and smaller in scale than PennApps. The event is not a typical college hackathon. It is open to students and professionals alike and is organized by faculty and students. The event offers some structure around team formation and problem areas that the hacks should solve for. It is described as a "two-day coding competition to transform data into real world solutions."

I attended this hackathon as a mentor. Unlike at PilotPhilly, I had more understanding of the event at this point and of mentoring. There were around 150 attendees who formed into 32 teams. Considering the event was open to the public, the attendees were mature, and knew they had to create an app that would utilize a data set to solve a problem. The competitive aspect of the event was evident and dominant.

As a mentor you are supposed to roam around and ask if teams have questions and need any support. Considering I had no technical expertise, I mentored at the early part of the event to support teams in their ideation phase.

I struggled a lot at the beginning to find a team that could use my help. It felt awkward to ask adult participants if they needed my assistance. Finally a conversation with a few team members led to them asking me what I do. I explained my design skill sets and that I could support them with ideation. So I followed them to the remainder of their teammates and I found they were a large group of 6 participants coming from various backgrounds: engineering, medical assistance, and graphic design, of which they were split into 3 students and 3 professionals.

The team knew how to create software, but according to one of their engineers, none of them created software for a living. Nevertheless, they had a brilliant idea utilizing data in healthcare to help doctors specify treatments for a very specific disease.
The first thing I noticed when coming to their workspace was that they had no means of externalizing their information. Most teams had a whiteboard nearby, but not this one. So I immediately supplied them with markers and a paper roll from my supply bag. They were happy I provided them with the materials and devised a way to hang the paper roll on the wall.

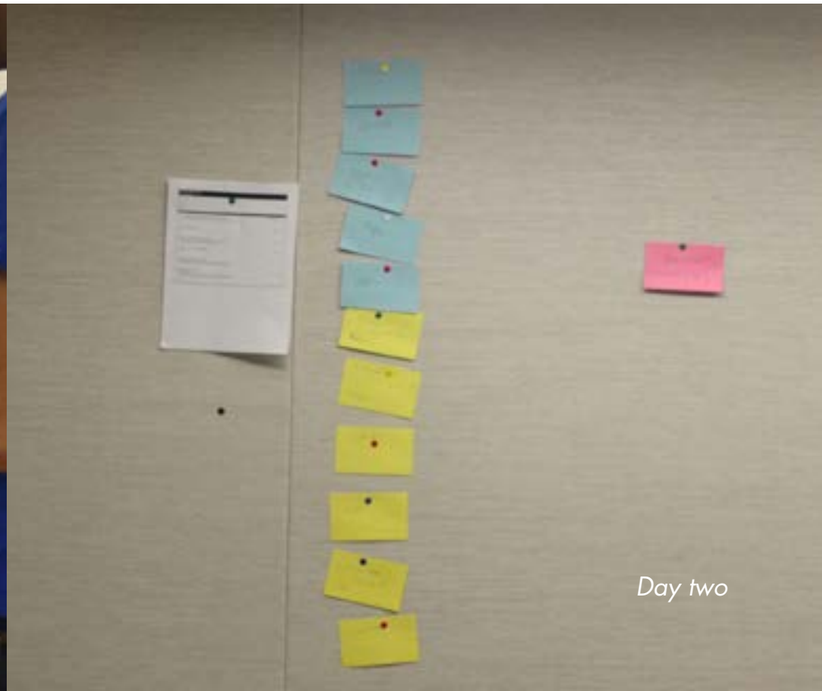
Unlike my experience with PennApps, I had difficulty applying a seamless design intervention as a mentor. For one, the participants viewed me as a more knowledgeable person then they were, which is not the case. And second, I wasn't with the team the whole entire time. I didn't have the advantage I did with PennApps. Here I was an outsider to the team. With my Design Thinking Cap on, I had to improvise.

The paper roll as a facilitation tool was my resort. The team was trying to explain the idea to me, and so I encouraged them to write it down, draw it, and start creating a persona for their main user. One engineer wasn't excited about the idea and said he needed to start coding as there is no time. But another member decided to write it out, and as she started writing with the help of another member explaining it, one member blurted out "I thought we were doing something else."

The whole dynamic and reaction to the paper roll suddenly changed. They discovered after externalizing their idea that they weren't even on the same page. I was thrilled because once again, design methods proved to provide the support needed for engaging teams. I was excited for the team and left them to get on track, and went on to check-in with other teams, and of course socialize.

The next morning I returned to Philly CodeFest to check-in with the team that I'd met the previous day. Most of them had gone to rest after a sleepless night, but I was thrilled when I saw their space. After I left the previous day, they went out to an art store and bought supplies to help them externalize their work and organize their process.

*Day one*



*Day two*

## THESIS STATEMENT

*Revisited*

SOFTWARE COMPANIES THAT STRIVE TO BE INNOVATIVE NEED TO PROVIDE CONVIVIAL ENVIRONMENTS, SUCH AS HACKATHONS, THAT INVITE VARIOUS DISCIPLINES AND SUPPORT TEAM ENGAGEMENT. DESIGN TOOLS CAN FACILITATE METHODS THAT WOULD FOSTER TEAM ENGAGEMENT WITHIN THESE ENVIRONMENTS. THESE METHODS SHOULD BE DESIGNED USING COMMUNICATION MEANS THAT CATER TO ALL COMMUNITIES OF PRACTICE INVOLVED WITHIN A TEAM.

# IV. THE BIRTH OF THE HACK-CHARRETTE

# ITERATE + IMPLEMENT

*PilotDC and the Idea Generation Workshop*

Informed by my research and my previous experiences, the Hack-Charrette developed as a prototype that was implemented in the form of an Idea Generation workshop at PilotDC—a high school hackathon. The Hack-Charrette is comprised of a short team building activity and 5 simple steps to get students started on their projects: Choosing a focus area, defining the user, generating ideas, affinity mapping ideas, and finally deciding on an idea to pursue.

The remaining documentation explains prototype design, implementation and workshop outcomes in detail.

## Pilot DC: Idea Generation Workshop

Attending PilotPhilly helped me connect with the organizers. I was presented with the opportunity to facilitate a workshop at another Pilot event being held in Washington DC. I was given a 1-hour time slot and so I decided to facilitate an Idea Generation Workshop.

I had the opportunity to iterate and improve upon my interventions at PennApps and Philly Codefest. I planned a workshop that utilized a design-charrette like process. It was designed with the following structure (Students were provided with all the materials at the workshop):

- Brief introduction to the process that the student will use for idea generation. I gave them the metaphor of a funnel–starting out with many ideas and then narrowing it down to one.

**- Team building:** I used a story telling activity where teams are given one piece of paper with the prompt "once upon a time…" and are asked to pass it around. Each team member adds one word to the story.

**- 5 step Idea generation process:**

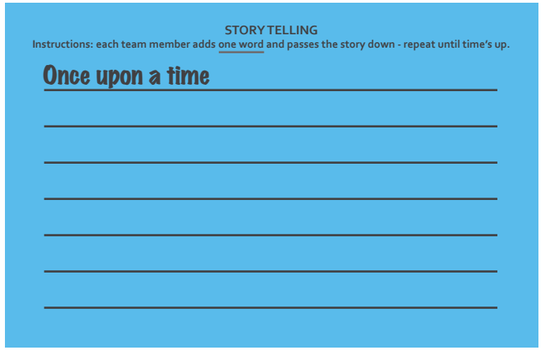
**Step 1.** Choosing Focus Area: I designed cards on each of which I listed a problem or opportunity area, along with examples of sub areas. This task was asking the students to collectively vote on what area of interest they want to generate ideas for their app. I also provided empty cards for them to suggest their own interest areas.

**Step 2.** User/Audience Definition: after selecting the area of focus students would then move on to defining a user for that area. I gave the students a template to begin with and showed them an example

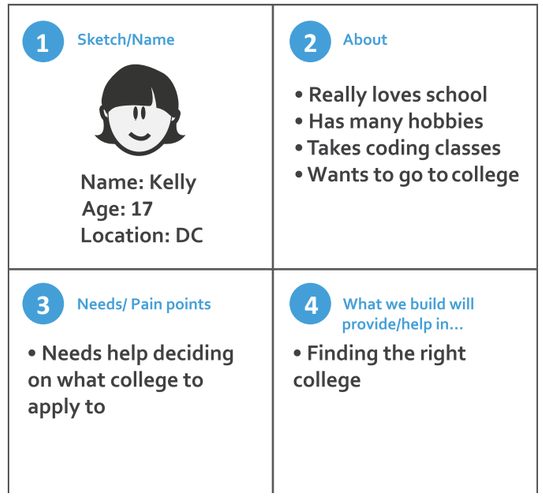
**Step 3.** Idea Generation: I gave the students sticky notes and markers and asked them to brainstorm ideas, feasible or not, for an app they would create for their user, within their focus area.

**Step 4.** Affinity Mapping: I showed the students how to take the ideas brainstormed in the previous step and cluster them based on relationships.

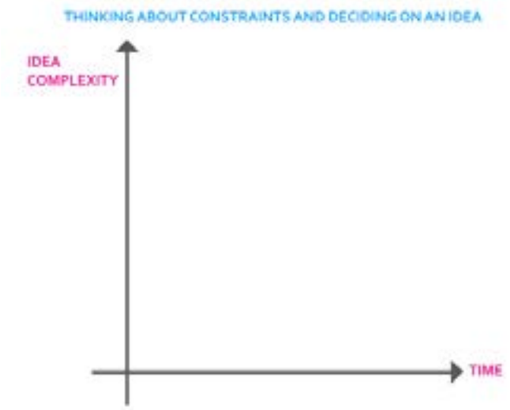
**Step 5.** Decision Matrix: the final step was simply providing the students with a method to come to a decision on what they want to pursue as a team. Here they were asked to place their ideas from the previous step in a matrix (shown below) to help them determine what is feasible within the constraints of event time, and idea complexity.




Top: Story Telling/Team Building Prompt
Bottom: Sample Focus Areas


Left: User/Audience Definition
Right: Decision Making Matrix



# V. PROJECT IMPACT

# WORKSHOP OUTCOMES

## *The Hack-Charrette at PilotDC*

*"We came here wanting to make a game, but after looking at the problem areas, we now want to solve for a problem."*

*PilotDC Idea Generation Workshop Participant*

For its first implementation, the Hack-Charrette was a huge success. 100% of participants showed interest in the process and wanted future Pilot events to host a similar workshop. The workshop also exposed 80% of participants not familiar with design thinking to design methods and processes. One participant even asked in an anonymous survey conducted after the workshop, if he/she could get a hold of the workshop process to utilize in their tech club at school.

The workshop was a success not only because participants showed interest in the process, but also, because what happened in the span of one hour made a positive impact on their experience throughout the event–this was expressed by close to half of the participants in the survey.

### The Hack-Charrette In Action

Overall, the workshop ran smoothly. The team building exercise at the beginning lightened up the mood and allowed teams to become familiar. One team read their story out load, sparking laughter. The process began with the area of focus and it was incredible to observe how teams started to converse around what path they wanted to pursue. For some teams it was an easy enough task, but for others it took an entire 9 minutes.

Participation in the workshop began with 32 students, and ended with 16. I was well aware that some students already had an idea but were struggling in narrowing it down. I encouraged them to stay, and make use of the steps that would help them narrow down their ideas. I allowed flexibility around teams staying or leaving when they reached that point. That was an important nuance to take care of, because I knew students came

in at different levels. If the idea generation session didn't support the project then that would be imposing a constraint upon them.

Creating a user/audience sparked a deeper level of thinking about who their app is for and what problem it is solving. The conversations became more dynamic. One participant whose teammates didn't attend the workshop alongside her asked my permission to go seek their opinion for the persona she was creating.

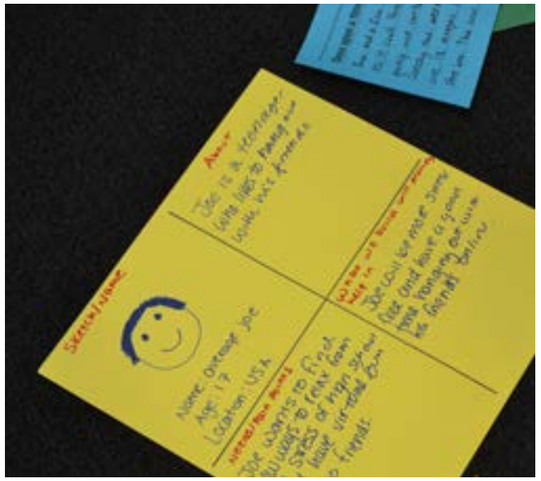
When arriving at the brainstorming step, some teams asked if they could leave because they had an idea they wanted to pursue. Affinity mapping prompted teams to organize their ideas into clusters based on relationships. This helped teams start the process of narrowing down the idea they would pursue.

At this point in the workshop, the last two steps began to bleed into each other and I, along with other mentors who came for support, began to work with teams one on one. The decision matrix was encouraged but was not used by everyone left. One team was stuck on clustering ideas, not because the task was difficult, but because when looking at the ideas they had brainstormed, they simply didn't think they were interesting. And as one team member stated: "we came here wanting to make a game, but after looking at the problem areas, we now want to solve for a problem."

After the time was up, I wandered around to further mentor other students. A few hours in, I found some of my workshop participants had been utilizing the methods they learned, and materials they generated at the workshop

Later on in the event, a participant who attended the workshop without her teammates came to ask for guidance. Her team, who had already started coding and had an empty web app with a title already in progress, said they didn't know where to go from here. She had already developed a solid user in the workshop, and I encouraged the team to utilize what she had defined and back track before going into the code. Her teammates gave me empty stares. If only they had attended the workshop as a team, they would have understood the purpose and utilized the methods more effectively.

Image credit: Brandon Klevence (BK)



## Post Workshop Survey

To understand the impact of the workshop on the entirety of the event, I designed a survey and sent it to participants following PilotDC. The key questions were around understanding how the workshop changed their experience, and if they used any of the learning throughout the event. The workshop outcomes are described below.

Total responses: 15 completed, and 3 partial
For 17 out of 18 it was their first hackathon

On how they met their team:

8 survey takers out of 17 knew each other in person before the event. 2 attended the event without a team and formed without any help, and one also came without a team but someone helped him/her find a team. Finally, one participant remained without a team.

On workshop attendance:

The workshop is most valuable when an entire team is present. Based on the survey, 7 participants attended the workshop with their entire team, and 3 attended the workshop alone. Equally, 7 members attended the workshop with a few members from their team. Participants who came alone or with their partial team were further asked whether they thought the missing team member(s) would have benefited by joining the workshop, 7 said yes, and 2 said no.

If and How the workshop helped them in developing their ideas:

Over 50% (9) of the survey takers said that the workshop helped them in shaping/modifying an already existing idea. Close to 30% (5) indicated that the workshop helped them come up with a new idea with their teammates. 1 indicated that the workshop encouraged them to change an already existing idea. The students described the ways in which the event helped them:

"The brainstorming and idea generating activities helped modify our initial idea into something the entire group felt would be successful"

"Not necessarily "changing", but improving a preexisting idea. Basically, we didn't go straight into coding and actually talked about things and layouts and stuff before hand, and organized things better."

"It helped me decide what our application would do and what its purpose would be. I think it saved me quite a lot of time and really helped me narrow my focus."

"It separated the team into two groups but helped my partner and I come to an agreement quicker and easier."

On evaluating specific steps:

As for the workshop process and steps, survey takers were asked to choose the most helpful steps from the workshop to developing their ideas as a team and to improving their communication. Step 1 (choosing a focus area) received 10 out of 17 votes. Step 2 (User/Audience Definition) received 9 out of 17 votes. Step 3 (Idea Generation) was voted 7 times. Even the story telling activity received 5 votes.

Students were then asked if they used any of the methods following the workshop and throughout the remainder of the event. Over 40% (7 out of 17) said yes. They described which methods they used:

" The sticky notes!"

"We thought how we could fit the audience"

"We used the user/audience definition"

"We were able to look back on our ideas that we brainstormed"

"We were able to use the audience four-square and the post-its for the rest of the 24 hours"

"When our final group was created after combining with another group, we used the idea generation part of the process. After spending a really long time thinking about what our app will be on, we decided to just stop, and write all our ideas on the white board. It helped us a lot so we could finally get working on an app."

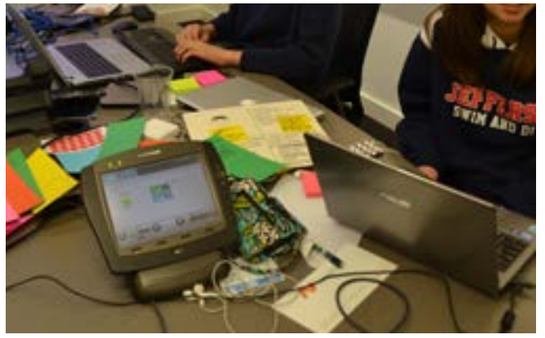
"We used the focus area to expand on our idea to address other concepts to incorporate to our project."

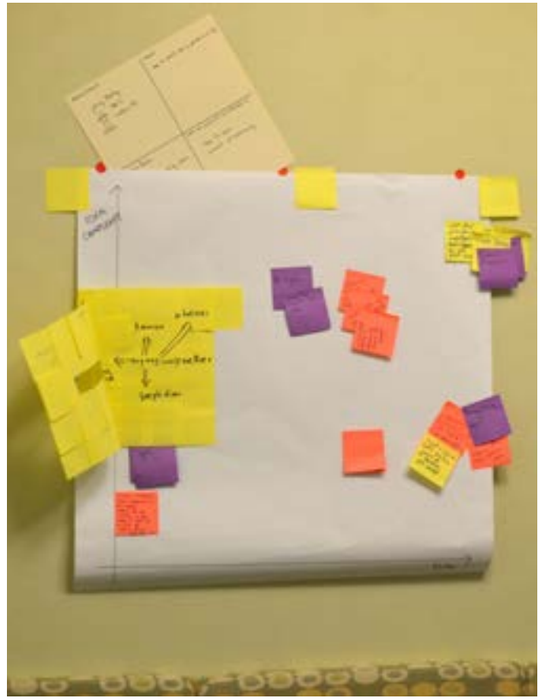
On future implementation of the workshop

100% of the participants indicated in the survey that they would use this kind of process again in the future.

Workshop impact in introducing design thinking into the hackathon environment:

Participants were asked in the survey if they have ever heard of design thinking before. 80% have not. They were then asked if they would use this kind of process for generating ideas in the future and 100% again indicated that they would.





In conclusion, the survey responses helped prove that the Hack-Charrette made an impact on participant experience. There is always room for improvement and the concept is at its early stages, but with participant feedback and the research conducted thus far with the small testing grounds at PennApps, Philly Codefest, and most importantly Pilot DC, participant feedback was coming from experiences at real hackathons.

At this point I am confident that the efforts and ideas behind this thesis project will withstand the test of time, and will slowly grow to become a part of the hackathon phenomenon. The Hack-Charrette demonstrates that design methods and tools can foster team engagement and ultimately support innovation processes.

Image credit: Brandon Klevence (BK)

# REFLECTION

## *How Will The Hack-Charrette Live On?*

Upon concluding the documentation for this thesis, and reflecting on my project process and outcomes, I realize that this is only the beginning.

I have managed to introduce design methods and tools into a movement that continues to take shape as future generations enter the tech field. In the hackathon context, the hope is that my methods will ultimately help in refocusing the event on learning, preparing students for a future of collaborative efforts, and last but not least, making it a truly open movement for others to join in.

As for the larger context established at the beginning of this book over tech innovation and the people who take part in it, the Hack-Charrette is a stepping stone in supporting tech industry culture change and the ways by which innovation and collaboration are approached. It also supports adopting human-centered design as a behavior and a mindset to solving problems not just limiting it to a process that needs to be integrated. Through exposing hackathon participants to the value of design, my project supports tech innovation, incrementally, one hackathon at a time.

On a more personal level, as a designer I learned a great deal from pursuing this project. My initial inquiries were far too broad. At times I felt that I was stuck, but my trust in the design process helped me manage and overcome the roadblocks. I didn't know what to expect upon attending these hackathons—a risky route for a thesis to begin with. But I believe that that part is what made me persist on making meaning out of all this. It's the designer in me, that actually went towards uncertainty, rather than avoid it. And to that I'm grateful.

My willingness to investigate and utilize this project as a learning opportunity, opened up doors of opportunities. I didn't know any of the people that I connected with in my project before, yet one connection led to another. The most recent one being an invitation from college hackathon organizers out of Virginia Tech, seeking advice on how to invite other majors at their school into the event.

And finally, there is the beauty of the design process. Not knowing what the end result will look like but learning to adapt and to navigate through the uncertainty. This project is the conversation starter for a long-term relationship. Like code that lives on the Internet for people to utilize and expand upon, I now officially declare the Hack-Charrette to be *open source*.

# VI. END MATTER

# POSTSCRIPT

The feedback received from my committee was around three points, all addressing future steps to build upon the success achieved in the early stages of this project.

The first point was with regards to communities of practice and how the project opens the door for designers and others to join. A key question was raised: what can a designer do at hour 12, for example, when the idea generation phase is over? The first step that was taken in this project was introducing design methods and tools into the hackathon space to open the doors for others to feel more comfortable when participating at hackathons—I was attending these events as a designer myself. The next step would be in addressing the specific and important question of what a non-developer is to do throughout the remainder of the event. My initial idea, as the hackathon is slowly transformed, is that down time would be the ideal time for exposing non-developers to programming.

The ultimate goal is not for everyone to become professional programmers, but for participants from other communities of practice to learn and understand the ways in which these technologies function in order to have a voice in their creation. In the larger context, this would allow for non-developers involved in software development to communicate and collaborate more effectively within their project teams. Whether it's a designer having a basic understanding of code, or a developer utilizing design thinking in her process, a degree of understanding as to what the other does can ultimately support teamwork.

The second point was on allowing room for negative feedback within the tools that I design in order to advance my learning, especially with regards to where more work needs to be done to improve upon future iterations. This is an essential and crucial point for me to consider. In its introductory phase, the PilotDC survey showed that the Idea Generation Workshop supported many participants in shaping their projects. But, there remain the students who were not engaged in the process. In future iterations it would be important to understand why those students are not engaged, and how that can be addressed. This point is connected to the third and final point that was provided by my committee.

The final point raised was with regards to how the project is taken to the next level in supporting tech innovation in the larger context. How might the Hack-Charrette be introduced to tech organizations and spark their interest?

My thesis committee acknowledged the compelling argument established in this thesis with regards to supporting tech innovation. They also acknowledged the great potential that lies in taking the project to the next level. As my thesis aims for incremental change, the final point is linked to the previous one because it hints at addressing how companies introduce design thinking to their culture. Introducing design thinking to a company's workforce is one thing, changing the culture to a design culture is another.

Further research would be needed to address the final point at that level, and the Hack-Charrette provides an excellent foundation for me to build upon.

# GLOSSARY

**Affinity mapping:** a method used to group information based on relationships

**Agile software development:** An approach to software development. The term is used to describe several software development methodologies that are characterized to be iterative, incremental, and adaptive. The methodologies under agile development promote incorporating iteration, continuous planning, continuous feedback, integration, and testing into the software development cycle. The approach also emphasizes collaboration and face-to-face communication within development teams.

**Code (*n*):** a system or language of rules and symbols that is used to provide a computer with instructions to perform specific tasks.

**Code (*v*):** writing such a system or language.

**Computer hardware:** the physical parts of a computer.

**Computer software:** the intangible parts of a computer in the form of symbolic languages and programs that are capable of performing various tasks. Computer hardware relies on software for directing its operations.

**Culture:** is the shared ways by which an integrated group of people behaves, thinks, and adapts in a social environment.

**Data:** quantitative or qualitative information that can be communicated, processed and interpreted.

**Developer:** someone who specializes in creating computer programs

**Digital:** any type of media that a machine such as a computer can read and process.

**Environment:** Is the intangible cultural and social states and circumstances that surround and influence an individual or community.

**Hard-coding:** A practice used in software development that describes inserting behaviors or data values into a program, in a location where modifications are difficult to apply.

**Human-centered design:** a design approach/process that puts the user and their needs at the focus of each stage.

**Innovation time off**: an organization giving its employees time off to work on side projects

**KG method:** a process that helps teams solving problems priorities their data and reach consensus.

**Methodology:** a system or collection of tools, methods, and principles that are used in a discipline to guide in achieving a specific goal or objective.

**Persona:** a fictional character that is created to help define types of users and their behavior towards a specific product, service or system. Creating a persona is a method that helps in addressing the specific needs of a user type.

**User:** a user is anybody who uses a computer or a system without completely understanding how the computer or system functions from a technical perspective.

**User experience:** the user's behaviors and experience interacting with a certain product, service or system. It includes how the user perceives the usefulness and value of the product, service or system.

**User-friendly:** a program or system that a user can easily use and interact with.

**User Scenario:** is a narrative with a specific goal that is meant to explain a possible interaction between a user and a product or a system.

**Waterfall model of development:** an approach to software development that uses a sequential process, starting from collecting project requirements, designing and implementing the software, verification and finally maintenance. Like a waterfall, progress in this type of model is perceived to flow downwards.

# BIBLIOGRAPHY


Baregheh, Anahita, Jennifer Rowley, and Sally Sambrook. *Towards a Multidisciplinary Definition of Innovation*. Management Decision. Vol 47, No 8, 2009. Print.

Bilton, Nick. *Tech Moves to the Background as Design Becomes Foremost*. The New York Times. Bits: The Business of Technology, 18 Jun. 2013. Web. <http://bits.blogs.nytimes.com/2013/06/18/tech-moves-to-the-background-as-design-becomes-mainstream/>

Bilton, Nick. *Distruptions: Mobile Competition Shifts to Software Design*. The New York Times. Bits: The Business of Technology, 16 Jun. 2013. Web. <http://bits.blogs.nytimes.com/2013/06/16/disruptions-mobile-competition-shifts-to-software-design/>

Ceruzzi, Paul E. *Computing: A concise History*. Cambridge, MA: The MIT Press, 2012. Print.

Daft, Richard L. and Robert H. Lengel. *Organizational Information Requirements, Media Richness and Structural Design*. Management Science, Vol 32, Issue 5, Organization Design May 1986. 554-571.

Gothelf, Jeff and Josh Seiden. *Lean UX: Applying Lean Principles to Improve User Experience*. California: O'Reilly Media, Inc., 2013. Print.

Guvenis, Moris M. *The Influence of Technological Innovations and Organizational Change on Facility Planning*. University Park, PA: Penn State Computer Integrated Construction Program, 1989. Print.

Hanington, Bruce and Bella Martin. *Universal Methods of Design: 100 Ways to Research Complex Problems, Develop Innovative Ideas, and Design Effective Solutions*. Beverly, MA: Rockport Publishers, 2012. Print.

Illich, Ivan. *Tools for Conviviality*. USA: Harper & Row Publishers, Inc., 1975. Print.

National Research Council of the National Academies. *Continuing Innovation in Information Technology*. Washington DC: National Academies Press, 2012. Kindle file.

Norman, Donald A. *Things That Make Us Smart: Defending Human Attributes in the Age of the Machine*. New York: Basic Books, 1993. Print.

Tate, Ryan. *The 20% Doctrine: How Tinkering, Goofing Off, And Breaking The Rules At Work Drive Success In Business*. Harper Collins, 2012. Kindle file.

Turner, Thea. Pernilla Qvarfordt, Jacob T. Biehl, Gene Golovchinsky, Maribeth Back. *Exploring the Workplace Communication Ecology*. CHI 2010: User Characteristics and Large-Scale Tracking. April 10-15, 2010, Atlanta, GA, USA. Print

Wenger-Trayner, Etienne. *Communities of Practice: A Brief Introduction*. Web. < http://wenger-trayner.com/theory/

# ABOUT THE AUTHOR

Yara has worked on a range of experience design and design research projects in both large-scale organizations such as Penn Medicine, and non-profit organizations such as Liberty Resources, Inc. and the Committee of Seventy in Philadelphia.

Currently involved in user experience design in software, Yara continues to expand and explore her interest in utilizing her skills to solve problems in technology.

Born in the Golan Heights, Yara self-identifies as a World Citizen. This feeds her passion for using a human-centered design approach to making a positive impact in people's lives. She enjoys a whole bunch of little things that make a big difference, like planet earth in the vast universe with all its sights and sounds.

Visit www.yarasafadi.com to explore other projects.

MiD | MASTER OF INDUSTRIAL DESIGN
THE UNIVERSITY OF THE ARTS