

D3 — Value Sensitive Design

Michael Brodskiy
College of Engineering
Northeastern University
Boston, U.S.A
Brodskiy.M@Northeastern.edu

Abstract—This document outlines an instance on the Northeastern University Boston campus where value-sensitive design was not taken into consideration; furthermore, it provides a possible solution to said design, in addition to restating the design as reformed problem statement.

Index Terms—technical, success, design, values, value-sensitive, technology, access, philosophy

I. INTRODUCTION

This document is prepared in accordance with the Institute of Electrical and Electronics Engineers (IEEE) specifications, and attempts to identify a value-sensitive design problem occurring in the context of Northeastern.

II. THE PROBLEM

A. Description

Quintessentially, computing relies on two factors: files and processes. Nowadays, the increase in complexity of computing systems has caused an abstraction from this straightforward dichotomy — words like “program”, “application”, and “tool” have obfuscated computing fundamentals (in most common cases, as exceptions do exist). For this reason, this document will lead an investigation into Northeastern’s overreliance on “applications” and “programs”, explain why this reliance leads to non-value-sensitive design, and propose a possible solution to said technical fix.

B. Significance

On a personal note, as a supporter of the Free Software movement, and one who tries to employ and adhere to the Free Software philosophy to the maximum extent possible, I was in awe of the amount of applications, with nearly identical functions¹, I was asked to install even prior to moving in on campus. Although I genuinely believed that some of the applications, such as one which checked in weekly on student mental health, had great motives behind them, whether I wanted to or not, I was barred from utilizing them. As Arnold Pacey, in his work entitled *The Culture of Technology*, puts it, “If it [technology] is to be of any use ...[this technology] must fit into a pattern of activity which belongs to a particular lifestyle and set of values” [1, pp. 3]. Thus, by utilizing Pacey’s definition of useful technology, in tandem with my aforementioned experience, it becomes

evident that Northeastern’s many digital applications can not be classified as either useful technology or value-sensitively designed. In this manner, it becomes a matter of finding a technological solution to this technical fix.

C. Defining Success

Throughout [1], Pacey provides examples like Eskimos utilizing snowmobiles and villagers using hand-pumps for water to show how communities and individuals developed technical fixes into full technological solutions through adaptation to their values. Furthermore, Pacey adds to this by stating, “...a ‘technical fix’ (not a technological fix) ... represents an attempt to solve a problem by means of technique alone, and ignores possible changes in practice ...” [1, pp. 7]. As the Eskimos adapted snowmobiles and the Indian villagers adapted hand-pumps through changes in practice, we, as the Northeastern community, and future industrial leaders, need a change in practice of the reliance on applications. This is best done by analyzing the existing scenarios: a technical fix can best be improved into a technological solution by allowing individuals, and, thus, communities themselves, to contribute, little by little, piece by piece, until a fully-functioning, value-based (not just sensitive) design is fabricated. It is then, and only then, that technology can become absolutely adherent to values, whether social, ecological, or anything in between.

III. VALUE-SENSITIVE DESIGN

A. Incorporating Value-Sensitive Design

As was stated before, the best method to generate a design that incorporates the values of those meant to use it is by simply giving people the ability to contribute. In doing so, rather than trying to create a “one size fits all” design, providing the infrastructure to someone to further build upon it and improve it, thereby incorporating their own values, allows for bettering the design for others. Extrapolating such a case to the Northeastern community, rather than providing many applications, with predetermined functions and capabilities, providing the infrastructure to fabricate a community application, with the support of the community itself, would allow for the values of all to be heard. This could be done in a git-inspired system, in which a base version of a community application, licensed under a freedom-respecting license (*e.g.* GPL, Apache, etc.) is created by Northeastern (*i.e.* the infrastructure). Users fork this to add their own modifications; some repositories will gain more traction than others due to more popular appeal

¹I am told this is a result of computer science students wanting to create something to add to their résumé

of features, and more and more users contribute to add their own features to an already-growing application. Such a process would be good in that, firstly, it allows for innovation, as, instead of having a small group working on implementing features, people can bring their diverse experiences and perspectives to add functionality; Secondly, it would allow users to port the application to any system (as they would now have access to the code), which would allow usability to all of those interested; Third, people can still contribute, and have a record of their contributions, meaning it is something they can still describe in a résumé. Finally, and most importantly, this would mean that people actually have their values taken into account, making the outcome an ever-growing, constantly-improving, community-driven value-based design.

B. Byproducts and Mitigation

In terms of byproducts, there are two which present themselves immediately, both quite minor. First and foremost, there is the possibility that some people in the community would find contributing to such a project a daunting task due to a lack of ability, which would cause them to refrain from getting involved in the community and even from using the application. This is easily mitigable in one of two ways: first, the bare application would still provide basic functionality, and, thus it can be used; or, second, if the client would want some kind of feature incorporated, they could simply be given the ability to request it from someone who has the ability to add it (a benefit of the git-inspired system). In such a manner, no one would be barred from making a contribution. A second byproduct would be a possible strain on Northeastern resources, as more support would be necessary to maintain such a community. This boils down to Northeastern's willingness to provide the necessary resources to allow the inception of such a network. In the short run, there may be a loss to the University, as overhead costs may increase, for example cloud hosting fees. In the long run, however, if such an application were to be fully functional, to the liking of the community, the University could benefit from a unified community. It is possible that such a movement could even grow beyond Northeastern, with many universities contributing to an application to benefit society. Such an outcome, that includes community-driven, feature-based adaptation, in addition to a free software license, is reminiscent of Canvas, the learning-management system, currently being used at Northeastern. This application is similar, albeit more specialized, than the presented idea.

IV. REDEFINED PROBLEM STATEMENT

The basic problem is that Northeastern needed a way to more easily connect their students to available resources. This could be expanded in the following way:

- Basic — Make an application for students to access resources
- Value-sensitive — Provide a basic application upon which students can expand to incorporate requested features and values

- Value-based — The problem is that Northeastern has many resources to which its students require quick, easy access to. This problem impacts all students, as, without the ability to directly interact with Northeastern resources, they will be unable to fully use what should be available to them during their time in college, which would mean they would not get the most out of their education. The students need to be able to access the necessary resources available to them, in a manner that is convenient and right for them and the university. The application needs to be designed in a way that allows for Northeastern to have some oversight over development, but also does not limit the students in adapting it to their needs and values. Such an application would be ever-evolving, structured in a git-inspired style.

REFERENCES

- [1] A. Pacey, "The Culture of Technology". Cambridge, Massachusetts: The MIT Press, 1985, pp. 1–12.