INTERNATIONAL CONFERENCE ON ENGINEERING AND PRODUCT DESIGN EDUCATION
8 & 9 SEPTEMBER 2011, CITY UNIVERSITY, LONDON, UK

EXPLORING COLLABORATION BETWEEN COMPUTER SCIENCE ENGINEERS AND VISUAL COMMUNICATION DESIGNERS IN EDUCATIONAL SETTINGS

Aaron GANCI, Dr Rajiv RAMNATH, Bruno RIBEIRO and R Brian STONE
The Ohio State University

ABSTRACT
Because of our increasingly technology-enabled society, computing-supported interactions are growing in both number and complexity. As a result, visual communication designers are now required to align themselves closely with computer science engineers. To create truly useful and successful software applications, a wide variety of specialties is now necessary. For the foreseeable future, design's success will be closely tied to the success of our computer science colleagues.
The recent proliferation of Agile environments has helped establish a dialogue between the fields. However, a lack of educational exposure to these collaborative environments can cause friction between the roles, ultimately affecting productivity in the professional arena. Open dialogue needs to be introduced earlier to allow students to practice working together towards a common goal.
In an attempt to start defining this critical space, graduate students from the Department of Design at The Ohio State University worked as consultants for development teams within two capstone courses in the Department of Computer Science and Engineering. Out of observations, surveys and interviews during the Autumn 2010 Quarter, recommendations were established to improve upon the traditional educational experience. These recommendations were used to define a new model of interaction for the students, now placing an emphasis on interdependency. In this redesigned classroom environment, students will now be required to work closely to integrate multidisciplinary perspectives into their solutions. By giving students the opportunity to work interdependently in an academic environment, institutions will help generate professionals that are prepared to work more integratively after graduation.

Keywords: Design, computer science engineering, application, development, agile, multidisciplinary

1 INTRODUCTION

Since the end of the last century, the increasing popularity of the Internet changed the way we interact with the world. A cursory example of this can be found in the way we simply look for and take in information. The world has transitioned away from newspapers and magazines toward websites and weblogs as the main source of information. Moving forward, not only media companies, but all companies are now rethinking their digital presence, creating new online services that require complex virtual environments. This is a big opportunity for visual communication design.
As a result of this transition, visual communication designers have had to redefine their skill set. In the early stages, designers learned simple programming languages like HTML and Actionscript in order to access new design opportunities. However, as the digital environment progresses, it is not enough to know HTML, XHTML, Flash, Ajax, or PHP. The time has arrived when visual communication designers must now closely align themselves with computer science engineers to produce complex programming solutions. It is no longer the designer's role to learn how to program and deal with APIs or complex databases. This is the responsibility of computer science professionals. As a result, for the foreseeable future, design's success will be closely tied to the success of our computer science colleagues.
This means that visual communication designers now need to be prepared for a shift in practice. Traditionally, designers had the terminal role in production. Even though they depended on the printing industry to produce their work, printers often took a passive role in the interaction. The

responsibility fell on the designer to check everything before sending it to print. When the focus moved to the web, designers became responsible for uploading files and fixing any problems that arose. But as preparation and production of our designs become more technically complex, engineers now have the predominant role in production. Although it does not necessarily mean less responsibility, it is a loss of control. Designers are not in command of the production and they will need to adapt to this new paradigm.

On the other hand, designers will now be able to approach more complex problems that used to be out of reach. They now have commensurate partners, everyone moving towards the same goal of making successful digital interactions. But, as we start to work in tandem, we must understand each other better.

As designers, the future of our discipline needs to be untied from the wrong idea that designers only "make things pretty". Aesthetics will still be a part of their role, but designers must advocate for their other skills. Similarly, engineers must promote their own skills and be willing to work jointly with designers. Only when we become aware of each other's strengths can we be truly successful.

2 AN AREA OF OPPORTUNITY

With this new collaborative understanding, we will be able to approach more complex interactions, which is rapidly becoming critical. Because of our increasingly technology-enabled society, computing-supported interactions are growing in both number and complexity. However, as Tulis points out, "this kind of increasing complexity and evolution of technology doesn't necessarily mean that...[they] are easier to use" [5]. This complexity has confronted the application development community with an interesting challenge. How do we not only ensure successful products, but also teach students to work in a new, more appropriate way in order to deal with this complexity?

The primary way that the development community can create more useful and effective products is to teach this more inclusive approach earlier. By bringing together several disciplines, including computer science, user interface design, user experience design, cognitive engineering and graphic design, we can approach complex solutions more effectively at all levels.

3 THE PROFESSIONAL EVOLUTION

The professional arena has been moving in a collaborative direction for a while already. As it shifts to this environment, the largest obstacle has been how each discipline defines success. The new collaborative model must be closely tied to an alignment of success metrics across all fields. Too often we get relegated into defining success through efficiency in computer science, and innovation or style in design. We need to start forming a common vocabulary for how we define success as a collaborative group. We must all be on the same page in terms of the vision, mission and goals of the application. One way to do this is to closely align our definition of success to the needs of the users within the scope of the project, a concept closely tied to user-centered design fields. In order to be successful in this context, elements of usability and design have to be integrated in the development process.

The recent adoption of Agile methodologies in the professional arena has helped the integration of multiple fields. The Agile Manifesto defined a new way to approach application development. This new approach favors "individuals and interactions over processes and tools, working software over comprehensive documentation, and responding to change over following a plan" [1]. The continued establishment and proliferation of these methodologies is an important element to uniting design and engineering.

However, this new unification in the professional space would not be possible without an alignment of goals and success metrics, especially when corporate funds are tied to that success. According to Riehle, Agile methodologies deal with success by redefining traditional value systems. He explains how a methodology and the success of its output are interrelated.

...users of the methodology must share the value system to make effective use of the methodology and its techniques. Only if these three pieces (a coherent value system, techniques that are compatible with the value system of a methodology, and users that share the values) come together harmoniously can a methodology be effective. [4]

The design community is able to interface with the development team more closely in these environments, encouraging interdependency. These methodologies have been utilized successfully in recent years, especially when user input is essential.

4 TRANSITION TO EDUCATION

Despite the adoption of a collaborative approach in professional settings, education has been slower to evolve. Agile methodologies and their benefits are widely taught and utilized by today's CSE students. However, the full experience of working on a collaborative, multidisciplinary team is not as common in either field. Without this inclusive experience, students become professionals without seeing the full benefits of the experience.

This problem is not relegated to one field specifically. Everyone involved in the development process has a serious lack of collaborative experience in the traditional educational setting. From the engineering perspective, Peslak reports that "...the importance of interface design and the topics of human-computer interaction have not risen to a high level in [the technological] sciences education." [3]. Similarly, Donald Norman best describes this issue from the perspective of design.

In educational institutions, industrial design is usually housed in schools of art or architecture, usually taught as a practice with the terminal degree being a BA, MA, or MFA. It is rare for in design education to have course requirements in science, mathematics, technology, or the social sciences. As a result the skills of the designer are not well suited for modern times. [2]

This is an area that needs careful consideration in the coming years as we become more dependent on complex computing-supported interactions. Students must be prepared to work in a collaborative way on graduation day, instantly becoming a valuable member of a development team. This idea will only work once we establish common success metrics within the context of education.

5 OBSERVATION OF THE ISSUE

In an attempt to start building this collaborative educational setting, several students and faculty at The Ohio State University have started to explore possibilities. During the Autumn Quarter of 2010, we, as graduate students from the Department of Design, enrolled in two capstone courses within the Department of Computer Science and Engineering (CSE). This was the first step in establishing a conversation between designers and engineers in this setting.

In the capstone courses, CSE students approach real projects that have been initiated by a business sponsor. The sponsors submit their concepts to the CSE department and are subsequently selected by faculty in accordance with relevance and complexity. Within the ten chosen applications, there was an impressive scope of unique challenges for both design and development. Examples of the projects included a math book for tablet computers, a sonar-based treadmill application, a GPS-based iPhone game, a visualization tool for gene mutations and their phenotypes, and a web-based development team management system.

Because the projects have real-world implications, the sponsors often have a wide range of specifications. Depending on the sponsor, the students may have specifications on technical issues, visuals, use and interaction, timelines, or deliverables. This provides a very complex exercise for the students.

The CSE students then embark on a rigorous investigation for a solution. We saw four distinct phases of the students' development process. (Figure 1). Starting with a definition phase, they established deliverables for themselves, specify development environments, map out their process, build use cases, and envision risks. After the initial work, they then moved into a user interface (UI) development phase. This was the area we were predominately approached for input. Because of a presupposition that we were primarily graphic designers, the majority of our input lied in Phase Two.

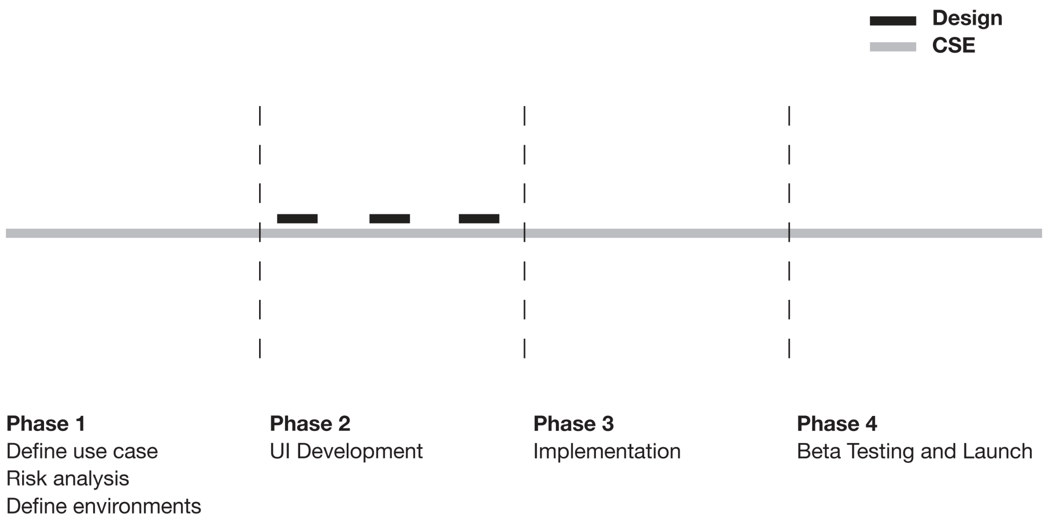

Figure 1. Current model for interaction between Design and Computer Science students

After the UI was relatively finalized, the team would move into an iterative implementation phase. The further the team progressed into Phase Three, the less likely it became that we were asked for input. The risk of having to revise already completed work because of a design recommendation seemed to deter some teams from interacting. Despite that reluctance, this was the second most prevalent phase for us to provide input. Finally, after development had progressed to a defined point, the team would functionally test their product, and possibly launch it in the end.

In an attempt to reduce the complexity of the projects, and supplement the experience of the engineering students, we worked as consultants on all the projects, ten in total. Taking the users' needs into account, we worked with each team to revise their applications, mostly in regard to user interface design.

Regardless of development phase, the teams usually approached us in an iterative process, predominantly in the second half of the course. Each team shared concepts that varied in both detail and level of completion. The media of the concepts all varied, ranging from hand-drawn sketches to interactive wireframes.

During these sessions, the team described their progress and then we got a chance to propose ideas for improvement. This would usually lead to a fairly open conversation in which a balance was reached, bridging design ideals and development capabilities. Because the interactions were so varied for each team, it was difficult for us to stay invested in all projects throughout the ten-week course. However, this was the most productive time due to our ability to work simultaneously towards a common goal.

Throughout the courses, we were also observing the process the students went through to approach their deliverables. Through methods including observation, surveys and interviews, we took note of areas that could facilitate better collaboration. By asking the students about their perception of design, and collaboration in general, we were able to gather some very useful insights. Out of these methods, we have defined a more ideal engagement.

6 OUR PROPOSED SOLUTION

Studio classes involving real projects from sponsors, like the CSE capstone courses, can be a great laboratory for designers and engineers. However, we observed several areas that need to be addressed to make the experience truly successful. Based on our experiences, we have several recommendations that will help foster collaboration and create a more holistic experience. In doing this, the result will be a better overall learning experience and subsequent product.

First, the area that needs dramatic improvement is an understanding of our colleagues' strengths. Both design and CSE students need to have a better understanding of how they can collaborate with each other. One of the biggest obstacles we saw for the interaction was a lack of understanding about each other's role or strengths. As designers, we did not know how engineers work, and engineers had a misconception of design in general. In a preliminary survey administered to the involved CSE

students, a wide variety of responses were recorded. In response to the question “What is your perception of the benefits designers bring to a project?”, the students’ responses ranged from “[designers have] knowledge of different types of frameworks so there is less overhead...” to “...[they] help with the user interface and general ‘look’ and ‘feel’ of our application”.
We saw this fractured view as a cause for friction in the initial phases of a project. Before the project starts, designers and engineers need to be exposed to each other’s disciplines. This clarification could be done before classes start or during the first week of the course, but it needs to be done early in the process.
Next, the projects and their teams should be carefully crafted to maximize efficiency. Before starting the project, sponsors’ proposals will be analyzed, and teams will be assigned. Depending on the scope of the application, the ideal is to have groups with two or three designers paired with four or five engineers. Then, designers and engineers would meet the sponsor together to have a full understanding of the goals of the product they will be developing. It is important to leave this meeting with a written proposal from the sponsor, detailing the expected goals for the project. After this introduction, the team will move into a series of updated phases that now stress interdependency (Figure 2).

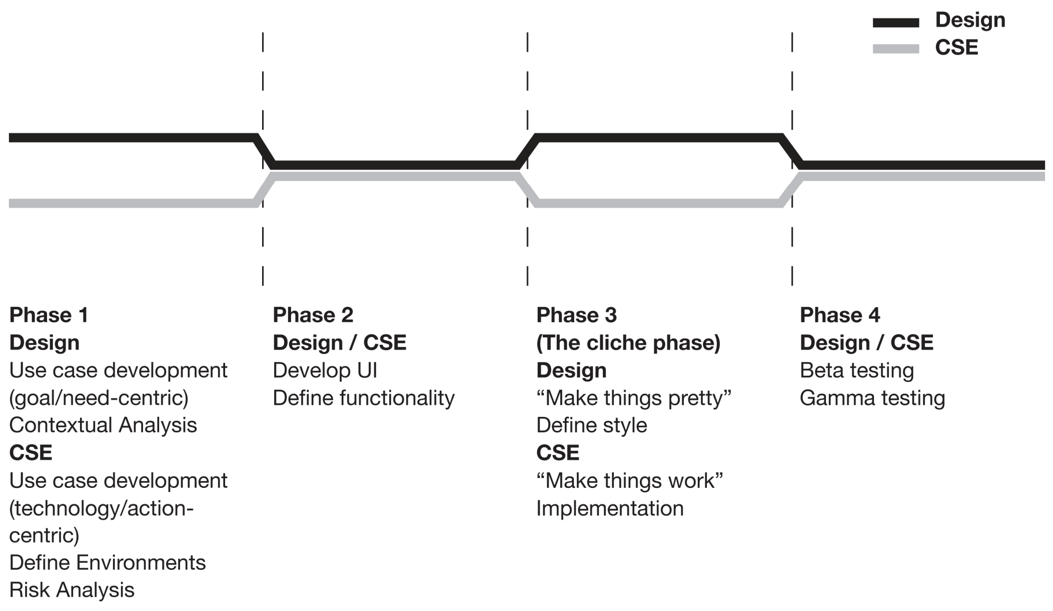

Figure 2. A mutually beneficial interaction model keeps both disciplines engaged throughout the educational experience

Finally, the process that these multidisciplinary teams work within needs careful attention. After meeting with the sponsor, designers and engineers will start working on the product. In the first phase, each discipline will now work parallel to one another. Designers would search for similar applications to form a contextual analysis. In addition, designers will need to develop use cases based on the user’s goals and needs for the product. Meanwhile, engineers would define development environments, analyze potential technological risks, and develop use cases based on actions needed to complete tasks within the product.
At the second phase, designers and engineers will realign. They come together to define functionalities and develop the user interface. This is a negotiation phase, when both teams will try to define ideals and then compare those with what is doable. Because they will be working together, both the user’s point of view and technological limitations will be integrated.
At the next phase, designers and engineers will work in parallel again. While engineers will implement the code for the product, designers will work on the visual style. We will label this the “cliché phase,” when engineers make things work and designers make things pretty. In this phase, each discipline will work hard to deliver a beautiful, functional and efficient solution that has been informed by the previous phase’s efforts.

The last phase is reserved for testing. The ideal is to do functional testing separately from user testing. However, in an academic environment, administering Beta testing and usability testing together can work nicely. While design students determine the quality of the application against design heuristics, engineering students can verify functionality. This phase will give the students valuable exposure to a variety of testing methodologies. The degree to which the test findings are implemented will be determined by the course's time constraints. At the very least, the students will benefit from reporting on their findings and making recommendations for future development.

7 MOVING FORWARD

Based on the approach we have defined here, the next step becomes the development of a fully realized course environment. Careful attention must be paid to the development of course goals, learning objective and accompanying assignments. Due to the multidisciplinary nature of this environment, new paradigms for educational goals and evaluative processes must become a priority. Crafting a system that allows the students to be evaluated in both their discipline, and as a collaborative team, is essential. A joint evaluation will stress interdependency and help impact the students to work together to create valuable solutions.

The presence and support of instructors from both fields is essential in facilitating a productive learning experience. Complementary to this is the establishment of a collaborative classroom environment. Students, under the guidance of faculty, work together in a studio environment, which may be a new concept for many engineering students.

Once a new framework is established for this experience, a pilot test will be essential to ensure success. Creating a transition plan and building a segue to the current capstone course will help all stakeholders evolve progressively. This new approach might cause friction amongst faculty or students, so easing everyone into the process will help aid in acceptance.

8 SUMMARY

Future development in this area will benefit the educational experience of both design and computer science and engineering students. By staying abreast of state-of-the-art software development methodologies, higher education can provide a unique environment for students to create valuable software solutions. The multitude of resources that are provided by collaborative efforts at the college level should be leveraged to prepare students for the work place. Ultimately, we believe that the integration of this new educational setting will help its stakeholders stay relevant while also generating top-tier professionals.

REFERENCES

[1] Highsmith J.A. *Agile software development ecosystems*, 2002. (Addison-Wesley, Boston).

[2] Norman D. Why Design Education Must Change. *Core77 / Industrial Design Magazine Resource*, 2010. (http://www.core77.com/blog/columns/why_design_education_must_change_17993.asp)

[3] Peslak A. A Framework and Implementation of User Interface and Human-Computer Interaction Instruction. *Journal of Information Technology Education*, 4, 2005, pp. 189-205.

[4] Riehle D. A Comparison of the Value Systems of Adaptive Software Development and Extreme Programming: How Methodologies May Learn From Each Other. *Extreme Programming Explained*, 2001. (Addison-Wesley, Boston).

[5] Tullis T and Bill A. *Measuring the User Experience: Collecting, Analyzing, and Presenting Usability Metrics,* 2008. (Elsevier/Morgan Kaufmann, Amsterdam).