User-Centered Design, Activity-Centered Design, and Goal-Directed Design: A Review of Three Methods for Designing Web Applications

Ashley Williams
Bridgeline Software
10 Sixth Road
Woburn, MA 01801
ashleyew@gmail.com

ABSTRACT
When conducting research with users in order to design web applications, the practitioner has a variety of methods from which to choose. This paper examines three such methods—User-Centered Design (UCD), Goal-Directed Design (GDD), and Activity-Centered Design (ACD)—in terms of their foundations, processes, and deliverables.

Categories and Subject Descriptors
H.5.2 [**Information Interfaces and Presentation**]: User interfaces – *evaluation/methodology, theory and methods, user-centered design*. D.2.1 [**Software Engineering**]: Requirements / Specifications. D.2.2 [**Software Engineering**]: Design Tools and Techniques – *user interfaces.* D.2.10 [**Software Engineering**]: Design – *methodologies*.

General Terms
Measurement, Documentation, Design, Human Factors.

Keywords
User centered design, UCD, activity centered design, ACD, goal directed design, GDD, user experience, user interface, usability, contextual inquiry, content strategy, information architecture, IA, personas, web design, interface design, interaction design.

1. INTRODUCTION
This paper examines three methods that are often used by researchers and designers as they help create web sites or applications. The methods are User-Centered Design (UCD), Goal-Directed Design (GDD), and Activity-Centered Design (ACD). Of course, web designers and researchers are not the only ones to employ these methods in the course of their work. These methods apply readily to such related disciplines and activities as technical communication, information architecture (IA), usability, Human-Computer Interaction (HCI), human factors, and content strategy, etc. Regardless of whether the given related discipline or activity is evaluative in nature (like usability) or generative (like information architecture), these methods—or at least *aspects* of these methods—can be brought to bear upon the work at hand. To better understand similarities and differences among these three methods in business settings, this paper reviews each method in terms of its foundations, processes, and deliverables.

SIGDOC'09, October 5–7, 2009, Bloomington, Indiana, USA.

2. USER-CENTERED DESIGN (UCD)
User-Centered Design may be considered a practice, field, craft, framework, philosophy, discipline, or method of designing tools for human use by involving humans in the design process. Regardless of how one couches UCD—framework, philosophy, etc.—the defining tenet of UCD concerns placing users at the center of design decisions. This is not to say that users themselves are producing the final design deliverables, even though users may engage in design activities as participants of design research. As Nielsen notes, "users are not designers, and designers are not users" [1:12-13]. Further, user-centered design is not about asking users directly what they want [8: 283, 284]—indeed, to do so would be to introduce bias. Rather, UCD is about a UCD practitioner (such as a user experience architect, interaction designer, information architect, etc.) profiling users and defining their behaviors of use of and preferences for various aspects of a given application, and using that information to then make design decisions about the web application. In this regard, UCD is a collaboration between designer and user.

2.1 UCD Foundations
User-Centered Design has multiple methods-, research-, and theory-related precursors from the 1980s and 1990s, including what was commonly referred to as Usability Engineering (e.g., see [1]) and Human-Computer Interaction (e.g., [2], [4], [5], [6]). Usability engineering serves as a solid foundation given its focus on the user, the user interface (UI), and the user's goals and approaches to the task at hand. Human-Computer Interaction (HCI) also makes sense given that it has historically approached computer use from the perspectives of cognitive psychology, computer engineering, and system design. However, there are lesser known terms to describe the foundations of User-Centered Design (e.g., "cognitive engineering" [3]).

Use of the term *User-Centered Design* can be traced to the 1986 publication, *User Centered System Design: New Perspectives on Human-Computer Interaction* [6]. However, if one were to pinpoint *the* foundation of UCD, one cannot overlook the role that Xerox's Palo Alto Research Center (PARC) has played. PARC

was established in 1970. By 1971, the "impact of the psychological advances on the human factors of how computers were used was not yet very great, though the potential was clearly there" [4]. In 1974, the Applied Information-Processing Psychology Project (AIP) was formed. The charter of this unit was to "create an applied psychology of human-computer interaction by conducting requisite basic research within a context of application" [4]. If Xerox PARC is said to be the foundation of UCD, then this charter is clearly its cornerstone.

2.2 UCD Processes

The UCD process comprises three phases: design research, design, and design evaluation. During research, the designer's purpose is, among other things, to assess who the users are and what their needs are. The second phase, design, should be obvious: based on findings from design research the designer designs (the user interface, the document, the information architecture, etc.). Once a design is drafted, the UCD practitioner then evaluates the design with users and revises it as needed based on the results of the evaluation. These three activities are merely the core activities of the UCD process. The UCD practitioner may also contribute to sales and project management related tasks like defining project scope and schedule, and he or she may also engage members of the project team such as creative visual designers and web and software developers in order to explain the design and even revise it as project or technical demands dictate.

2.2.1 Phase 1 - Design Research

During design research, the designer's purpose is to assess who the users are and what their needs are. To say "understanding users and their needs" is to put it simply. At a high level, design research typically involves planning, conducting, analyzing, and reporting on research data—each of which itself typically involves multiple steps. Only select steps are in described in this section because detailing any of these steps could easily warrant a stand-alone paper in itself. However, because the central tenet of UCD is placing the user at the center of the design process, I provide in this section most detail about interviewing users.

2.2.1.1 Planning Design Research

Planning design research typically involves focusing on the business in order to identify up front their goals, constraints, and assumptions. By knowing the business' goals and assumptions, the designer can make informed decisions about what to research and how to conduct it—the two key activities of planning research. Planning research by focusing on the business involves:

Defining who the business stakeholders are. In generally, they are the business sponsors of the project and others persons in the company or organization who have something at stake in the project.

Interviewing the business stakeholders and articulating their needs. One could name the users stakeholders, since users obviously have something at stake in the application. However, it is important to distinguish between business stakeholders' needs and users' needs. For example, the goals of the marketing department as stakeholder are different from the goals of the prospective customer as user. In such a scenario, one goal of the marketing department is to increase sales. A goal of the user, on the other hand, is to purchase something. Furthermore, business stakeholders may make assumptions about what users want, or worse, make design decisions based on their personal vision for the website or personal preferences for interacting with a website. And stakeholder's visions and preferences are likely to be different from users'.

2.2.1.2 Conducting Design Research

Conducting design research typically involves the following:

Conducting background research as needed on the subject matter. If the designer doesn't already have knowledge of the subject matter, it is important to conduct background research before interviewing users so that the designer may be as prepared as possible for the user interview. For example, if the project is a business-to-consumer (B2C) effort, the subject may be an online record store, and depending on his/her experience, the designer may need to research trends in e-commerce and transactional sites. If the project is for a pharmaceutical company, the subject may be one or more aspects of early stage drug discovery, and the designer may need to conduct background research to become familiar with relevant terms and processes, etc. before interviewing users. If the project concerns financial services, the subject matter may be annuities or mutual fund management, etc.). Background research includes interviews with subject-matter experts (SMEs), an audit of the existing application if one exists, and literature research. Data sources for literature research vary by subject; one may research publications by Gartner or Forrester, publications in Pub Med, or publications produced internally, such as company white papers.

Assessing competitors' work. The designer evaluates or audits related or competing websites or applications, etc.

Interviews with users. User interviews can take multiple forms. The two types of interviewing techniques described here—in-person interviews and remote interviews—are popular; many lesser known but still effective interview techniques are available to the UCD practitioner, such as day-in-the-life studies, self-documentary or self-reporting studies, diaries, beeper studies, etc. One of the most highly effective interview techniques is the "in-person interview." In-person interviews themselves take multiple forms. (See [13] for more about interview practices and techniques.) The designer may follow a contextual inquiry approach (see [7]), by shadowing the user and observing their use of the application in the natural context in which they would use it. Observing their use of the application assumes that the project is for a redesign of an application. If the project involves design of a new application, the designer would shadow the user and observe the work they do in light of the application to be developed. Or, instead of conducting a contextual interview, the designer may conduct a formal, face-to-face interview, asking a mix of (A) planned and exploratory questions and (B) close- and open-ended questions of the user. During formal interviews and contextual inquiry, the designer may gather relevant source materials used by the interviewee, and they may engage the user in such data gathering methods as card sorting, divide-the-dollar, free listing, concept testing, or A/B testing, etc.

Remote interviews may be conducted by phone or web. The web-based remote interview involves both interviewer and interviewee using a web-based application that allows for such activities as screen sharing (such as WebEx or GoToMeeting) and other ways of interacting such as point or gesture with the mouse cursor,

click to select, or drag and drop, etc. (such as MindCanvas or WebSort).

To help record notes during either type of in-person interview or during remote interviews, the designer can take field notes using pen and paper or computer, take screen shots of a computer screen, take photographs, and/or record audio or video. Alternately, rather than have only one designer conduct and record the interview, a team may be employed, whereby one designer asks questions and in general facilitates the interview, while a second designer takes notes.

Secondary forms of research. These include surveys; questionnaires; reviewing SEO data (such as key search terms entered into external and internal search engines), analytics data (such as abandonment rates and click-through rates), and log files (e.g., server logs, search logs, etc.); and gathering information from customer support, technical support, and/or marketing research.

2.2.1.3 Analyzing Design Research

Analyzing design research can take multiple forms, depending on the type of research conducted, the kinds of data gathered, and the overall purpose and goals of the project. Since UCD research tends to be qualitative, common analytic techniques include debriefing (with interviewees and/or with the UCD team, if there is one); listing (e.g., listing early guesses, key findings, and/or key recommendations, etc.); and clustering (reviewing the data and noting trends such as similar needs between different types of interviewees).

2.2.1.4 Reporting on Design Research

Reporting on design research involves writing and presenting findings about the users and their needs and recommendations about the design direction, etc. The content and style of the report depends on the audience and their needs for the report. While business stakeholders, visual designers, technical architects, and engineers may need findings and recommendations based on UCD design research, each audience type has its own needs and will use the results in different ways. For more information about reporting the results of design research, see [12].

2.2.2 Phase 2 - Design

The design phase involves brainstorming and conceptualizing and sketching initial drafts of the design based on findings from design research. Little is written in the UCD literature on how practitioners tactically bring research findings to the design process; such information could benefit the field tremendously.

After one or more sketches are produced, the UCD practitioner uses a relevant software program to produce one or more drafts of various key design deliverables. Please see section 2.3 for a list of design deliverables.

2.2.3 Phase 3 - Design Evaluation

Evaluating the design typically involves testing it for usability. Usability itself has a rich history and a large body of literature and is considered a discipline unto itself. See [9] and [32] for more information about usability—but note for [9] that while the title of the book *(Measuring the User Experience: Collecting, Analyzing, and Presenting Usability Metrics)* suggests that the authors will detail how to present usability findings, there is actually little information provided on presenting usability findings. Beyond formal usability testing, other design evaluation techniques include heuristic or expert reviews, satisfaction questionnaires, walkthroughs, etc. For more details, see [11: 614-622]

2.3 UCD Deliverables

This section lists common deliverables produced during each of the three UCD phases. By "deliverable," I mean a document submitted formally to a client, customer, or key stakeholder. During the UCD process, many other kinds of documents are produced which are not formally shared with others outside of the members of the UCD team or project team. These other kinds of documents include field notes taken during interviews, various documents and spreadsheets used during data analysis, sketches drafted during the early stages of the design phase, and usability test scripts and observation coding forms prepared for use during the design evaluation phase.

2.3.1 Design Research Deliverables

During design research, the deliverables shown in Table 1 are typically produced. This list is by no means exhaustive.

Table 1. Sample deliverables produced during the design research phase of the UCD process

Deliverable	Description
Written report of findings and recommend-ations	This is typically formal in nature. Depending on the scope of the project, it may be long and detailed, including such sections as an Executive Summary, Table of Contents, Overview, Background, Methods, Audience profile, User Needs or Task Analysis, Usage Scenarios, User Requirements, Business Requirements, and Recommendations. It may be delivered in one or more formats, e.g., portable document format (.pdf), or web (.html).
Presentation	The presentation is different from the report in terms of the display of information. It is typically produced in a software program like PowerPoint.
Personas	These are user archetypes that capture users' needs, etc. The "look and feel" of this deliverable varies widely. For UCD purposes, an effective persona will describe a given user, his/her level of knowledge, context of use [10], goals, tasks, constraints, and requirements (where a requirement isn't merely a given system feature, it's also the related task to be performed by the user via the feature), etc. Note that UCD has adopted Personas from GDD.
Process Flow	Also called Task Flow, Wireflow, Activity Diagram, or Decision Tree. Its purpose is to represent the flow of a process such as a decision-making process, an information gathering process, or a particular activity, etc.
Usability Test Plan	If usability tests are planned, the best time to draft test plans is during the design research phase, when results are fresh in the mind of the designer.

2.3.2 Design Deliverables

During the design phase of UCD, the following deliverables are typically produced by the UCD practitioner. Again, this list is not exhaustive, and many other kinds of documents are produced during design.

Table 2. Sample deliverables produced during the design phase of the UCD process

Deliverable	Description
Sitemaps	These are also called Site Diagrams. They provide a "birds-eye" view of the site via hierarchical diagram of the pages of a website or application.
Wireframes	Wireframes are page-level schematics or "blueprints" for any given page of a web site or application. They are typically annotated to explain various aspects of the user's experience, such as how a user interacts with a given widget. They may also annotate social objects (tagging, sharing, social bookmarking, etc.) and opportunities to optimize the page for search engines, etc.
Process Flows	The process flow is produced during this phase only if it was not created during the design research phase.
Prototypes	Prototypes are working models of the application. They may be low- or high-fidelity. Low fidelity prototypes (such as paper prototypes) tend to look like wireframes. High fidelity prototypes tend to be interactive, that is, clickable via computer mouse.
Content Strategy	This deliverable may consist of multiple documents. It can include findings from an audit of existing content, recommendations on how content should be used on the site or application, a map or diagrammatic flow of how content is generated, edited, and published, a list and definition of content-related roles such as content owners and contributors, etc. Related deliverables include Content Inventories, Taxonomies, and Controlled Vocabularies.

2.3.3 Design Evaluation Deliverables

The two most common deliverables produced during the design evaluation phase of the UCD process are the Usability Test Plan and the Usability Test Report. Again, this list is not exhaustive, and many other kinds of documents are produced during design evaluation which are not formally shared as deliverables, e.g., usability test scripts and observation coding forms.

The usability test plan is produced during this phase only if it was not created during the design research phase. Alternately, if it *was* created during design research, it may be updated during this phase. The usability test plan includes goals and objectives, a description of test participants, the test environment and protocols, test scenarios, scope, and details of what will be measured when testing for performance (that is, whether it's usable) and what will be assessed when testing for preference (that is, whether it's useful and enjoyable to use), etc.

3. ACTIVITY-CENTERED DESIGN (ACD)

3.1 ACD Foundations

ACD has roots in a variety of theories and disciplines. Like UCD, ACD has foundations in human-computer interaction [14]. But it also has roots in distributed cognition (e.g., [33]), computer-supported collaborative work (CSCW) (e.g., [14], [34], and see [18]), and the highly democratic, Scandinavian Participatory Design (which itself has a large body of literature). ACD can also be traced to "postcognitivist theory and practice from the fields of sociology, communication, education, and organizational studies, as well as from science and technology studies […], and computer-supported collaborative work" [18].

However, its most notable theoretical roots are in Activity Theory, which is richly documented in a large body of literature. The foundations of Activity-Centered Design can be traced through Activity Theory to Russia, in Marx's Theses on Feuerbach, and in a Soviet-era Russian theory of psychology (particularly the work of Vygotsky [16] and Leont'ev [17]) [15]. Marx is mentioned here only because his concept of labor informed Leont'ev's concept of activity.

Despite the rich and complex foundations of ACD, some authors [20] attribute Don Norman with founding ACD because of Norman's publication of [19]. The occurrence of mis-attributions such as this warrants critical analysis in itself, and Spinuzzi has offered some critical insight into why crediting Norman with founding ACD is misguided [22].

Moreover, Gifford and Enyedy were writing about ACD in 1999, in a paper in which they focus on theory, learning, and computer-mediated educational environments and seek to develop a new theoretical framework for computer supported collaborative learning (CSCL) [35]. So as early as 1999, scholars are positioning ACD as an approach that "emphasizes the design of computer-mediated environments to support and structure the interactions and interdependencies of an activity system" [35]. In this work, Gifford and Enyedy offer a useful account of ACD, and anyone seeking to articulate ACD's value to the design of web applications has much to learn from this early work on ACD. For example, the authors distinguish ACD from Learner-Centered Design (LCD), which grew from UCD. (Please see the conclusion for further discussion of the differences between ACD and UCD). Still, despite its rich foundations, ACD has no authoritative text among the industry-oriented publications intended for practitioners who design web applications (compare with the ample publications about UCD that are aimed at practitioners).

3.2 ACD Processes and Deliverables

In light of the recent enthusiasm for ACD among leading practitioners in the field [19], [20], [21], and given that ACD has formally been known and called ACD for at least 10 years), it is surprising that no definitive text yet exists that profiles the processes, methods, and deliverables that are to be used or produced by the ACD practitioner in business, or workplace

contexts. Consequently, this section provides less information about ACD than the other sections provide about UCD and GDD.

At a high level, it seems that ACD presents a dramatic, if merely a surface-level shift from UCD in that ACD asks not what tasks or activities the user must *perform with* the application but what tasks or activities must be *enabled by* the application/tool/system. In the industry-oriented publications that are available on ACD, the move is from "understanding your users as people" to "understanding them as participants in activities" [31].The shift seems surface-level and almost nominal at best, since UCD processes include both contextual inquiry and usability techniques—two approaches that emphasize a person engaged in an activity. Norman [21], drawing from cognitive anthropology [23] calls for practitioners to use not only taxonomies by "taskonomies"—however, these are not unlike UCD's task flows, focusing on task, activity, and goal.

Some authors [24], [25] have noted that Activity-Centered Design remains largely theoretical in nature. It is likely that the very foundation of ACD—rich, complex, and largely theoretical—simply does not translate readily to the day-to-day practices of design in industry. Chi writes: The translation of theoretical into practical takes a lot of time, and may be overly difficult (or not that meaningful)" [25]. To the contrary: while translating the theoretical into the practical will certainly be challenging, it should also be meaningful given that prominent figures such as Norman and Hoekman already tout the potential of ACD in practice. But more work is needed in order to delineate what an ACD process might look like when applied to the practice of designing web applications. Doing so is sure to be a worthy endeavor that will advance the field.

4. GOAL DIRECTED DESIGN (GDD)

4.1 GDD Foundations

Goal Directed Design was developed by Alan Cooper et al. over a span of years between 1983 and 2000. One author writing about GDD notes that GDD was "developed entirely through practice in the real world," a statement which implicitly raises the question, what user research methods were *not* created from practice in the real world? (27: 7). The primary texts ([26], [27]) that write about GDD are texts authored by those who either are attributed with developing this method, or contributed to its development in some way. Although no theoretical foundations are explored in the primary texts on this method, one can clearly see anthropological and ethnographic underpinnings to the processes involved. In fact, Cooper himself notes that the research phase of the Goal Directed Design process employs ethnographic field study techniques [26: 20]; however, like the primary texts on UCD published for use by those working in industry, theory is not discussed at any length.

4.2 GDD Processes

The processes of GDD are not that different from the processes of UCD. Cooper himself lists the following design research activities as being most useful: stakeholder interviews; SME interviews, user and customer interviews; user observation / ethnographic field studies; literature review; and product / prototype and competitive audits" [26: 52].

According to [27], it takes a year for even a skilled designer to master only the *fundamental* techniques of GDD, a year which includes taking classes and an apprenticeship. It would take "two years or more before they can take full advantage of the method's potential" [27: 7]. In light of the rigor which one must undergo to become skilled at the GDD process, it will not be possible to do justice to delineating that process here. So I offer a high level overview: According to [27], GDD comprises four components, of which process is merely one component:

1. Principles
   Basically, these are best practices which any given seasoned practitioner should know, e.g., design ample white space, avoid clutter, etc.

2. Patterns
   Goodwin suggests that patterns are pre-fabricated solutions [27: 9], perhaps not unlike Yahoo's well known UI Patterns. The patterns of which Goodwin writes are not to be confused with the "behavior patterns" of which Cooper writes. Behavior patterns emerge from analysis of the data from the user research. [26: 20].

3. Process
   This component involves the planning, conducting, analysis and modeling of research, and the specifying of requirements, etc. Because the activities of the "Process" component of the GDD method are most similar to UCD processes, I will focus on the GDD Process step in the remainder of this section.

4. Practices
   This component concerns organizational communication practices that would support GDD approaches and applications in the workplace.

4.2.1 The GDD Process Component

Each of the following steps of the GDD process is itself a multi-step approach, none of which differs significantly from UCD's multi-step processes and sub-processes. While two recent primary texts on GDD ([27] and [26]) differ slightly in their explanation of the GDD process, I describe here the process as put forth by Cooper [26] since he is considered the founder of GDD. This process includes research, modeling, requirements, framework, refinement, and support.

4.2.1.1 Research

Like UCD, GDD research may be quantitative or qualitative, but is usually qualitative. At a high level, GDD research for a given project includes a team ramp-up, a definition of scope, stakeholder interviews, competitor and literature review, a product audit, and user interviews and observation [27: 51 and 26: 24].

4.2.1.2 Modeling

The modeling phase of the GDD process includes synthesizing stakeholder and user findings, and an iterative process among analyzing data, drafting findings, and modeling the data via various deliverables, most notably personas.

4.2.1.3 Requirements

Like the modeling step, the requirements step is also analytic and iterative. GDD requirements include personas—which capture users' needs (see additional discussion in the GDD Deliverables section below)—and business requirements, which capture the goals and objectives of the business.

4.2.1.4 Framework

The framework step of the GDD process is similar to the Design Phase of the UCD process—it is the step during which the GDD practitioner defines the "form and behavior" of the application [27: 11].

4.2.1.5 Refinement

This step is a refinement of the framework step insofar as GDD practitioner further refines details of the form and behavior of the application during this step.

4.2.1.6 Support

During the support step, the GDD practitioner provides support to other members of the GDD team who must rely on the deliverables produced by the GDD practitioner in order to do their own work, which can include the technical architecting and developing of the application.

4.3 GDD Deliverables

GDD shares many of the same deliverables as UCD. The most well known deliverable of a goal-directed design approach is personas. As a deliverable, personas are so well known in fact that they have been adapted by practitioners who do not strictly follow the GDD process in their work.

In this section, I provide a high level overview of the deliverables produced by the GDD practitioner in each of the steps of GDD's Process component. Again, the steps are research, modeling, requirements, framework, refinement, and support.

4.3.1 Deliverables of GDD's Research Step

Because the focus of the research step is to define project scope and asses the needs of both the business and the end user, the primary deliverable is a Statement of Work that profiles the scope of the design/development project.

4.3.2 Deliverables of GDD's Modeling Step

During the modeling step of the GDD process, GDD practitioners may develop any of the following deliverables. Note that the first deliverables—Personas—is the key deliverable.

4.3.3 Deliverables of GDD's Requirements Step

The deliverables of the requirements step—Personas and a User and Domain Analysis document—articulate user, business, and system requirements. See section 4.3.2. for a description of Personas. The User and Domain Analysis document is not as narrative or story-like as Personas. Instead, it details "functional and data needs, user mental models, design imperatives, product vision, business requirements, and technology" [26: 24].

4.3.4 Deliverables of GDD's Framework Step

During the Framework phase, the GDD practitioner will produce a Form and Behavior Specification, which is a document that details the design. Other deliverables produced in previous steps inform the creation of the form and behavior specification, including personas and scenarios.

Table 3. Sample deliverables produced during the modeling phase of the GDD process step

Deliverable	Description
Personas	As mentioned above, these are user archetypes that capture users' skills, environments, behaviors, and goals, etc. In the GDD process, they also include what are called "scenarios": descriptive, narrative stories about how personas engage or use the application. All of these aspects of personas are meant to capture what are commonly called "users' needs."
Decision Tree	Like UCD's Process Flow, the Decision Tree represents a flow or sequence, but its focus is on questions that need to be answered or decisions that need to be made.
Mental model	This deliverable is a representation of the user's perception of a given situation or process. It includes the different concepts or pieces of information that are important to the user in light of the application. In representing the user's mental model, the designer typically applies the terms, words, or phrases generated by the user during interviews.
Taxonomy	This deliverable is a map or list or outline (usually hierarchical) of the user's perception of how any given concept or piece of information is related to another.
Affinity Diagram	This shows the results of organizing trends in the data, by highlighting clusters that can occur across multiple data points.

4.3.5 Deliverables of GDD's Refinement Step

During the refinement phase, the GDD practitioner will refine the Form and Behavior Specification, usually by adding details about the application's "appearance, idioms, interface, widgets, behavior, information, visualization, brand, experience, language, [and] storyboards" [26: 24].

4.3.6 Deliverables of GDD's Support Step

During this last step of the GDD process, the GDD practitioner will finalize the Form and Behavior Specification, based on any needs introduced by technical constraints of the development process. For example, a given design decision may not be feasible given development's technical environment, scope, or schedule, etc., and the design decision may need to be revised and finalized in the Form and Behavior Specification.

5. CONCLUSION

5.1 Overview of UCD, ACD, and GDD

While UCD focuses on the user (and considers goals and tasks in so doing), ACD focuses on the activity, and GDD focuses on the goal of the user. Where ACD asks what tasks or activities must be

enabled by the application/tool/system, GDD asks why the user must perform those tasks or activities, in order to understand what value, purpose, or meaning the tasks or activities have for the user. In these two regards, GDD is more like UCD and less like ACD, insofar as goals—being broader, higher level, and more encompassing—are different from tasks or activities, which are more concrete, granular, and specific: they are the specific steps that enable a person to achieve a goal. UCD, meanwhile, considers who the users are and their level of knowledge (how familiar are they with the application? how familiar are they with its purpose?), their context of use [10] (what are the technical, physical, and organizational environments like in which users use the application?), their reasons for use (why are they using it?), their performance patterns (how do they interact with it?), and their preferences (do they enjoy using it?).

On the other hand, GDD can be said to be like ACD in that it focuses on the user's goals, which are not to learn or explore the web application but something larger such as to buy a book for a relative's birthday. In this regard, the web application becomes the tool that enables the user to achieve his/her goal.

5.2 Problematizing UCD and ACD

Some authors have problematized UCD [8], [19], [20:40-41], [29], [30], [31], [35]. Others argue that ACD as a method is superior to UCD [19], [29]. However, the methods of ACD have not been clearly articulated in literature aimed at practitioners. I would argue though that the two are not so different, that each, in fact, informs the other. ACD clearly emphasizes activities and tools, but because UCD's focus is the user, UCD inherently considers activity and tool because one cannot separate user from activity or tool when researching, designing, or evaluating the user experience of websites, web applications, help systems, documents, or genres, etc. Earlier in this paper, in the review of the foundations of UCD, I noted the charter of PARC's AIP and went so far as to say that if PARC were the foundation of UCD, then AIP's charter is its cornerstone. I restate the charter here in order to point out its operative words: The charter was to "create an applied psychology of human-computer *interaction* by conducting requisite basic research within a *context* of application" [4]. The operative words here are *interaction* and *context*, and I call them out because such criticisms of UCD as [19], [21], [29], [30], [31] seem to overlook the foundations of UCD, wherein we find not merely a static, descriptive focus on who a user is but a dynamic focus on a *user interacting with a computer in a given context*.

5.3 Need for further work

The review provided in this paper of these three methods is by no means exhaustive. It will be valuable for researchers and practitioners interested in these methods to examine in more detail the theories that inform these methods. For example, by examining UCD through the lens of Activity Theory, one may produce formidable objections to the criticisms of UCD, or one may further problematize UCD. Alternately, by documenting and developing the practice of ACD through the lens of Activity Theory and related theories and frameworks, one may advance the theory, study, teaching, and the practice of designing digital tools for use by humans.

Other possibilities for further investigating these methods include delineating what each method involves (vs. the high level overview provided in the short span of this paper), articulating its level of difficulty in learning and mastering for novice practitioners and students, uncovering its pedagogical implications, identifying the constraints and affordances it lends to any given design project, defining the advantages and limitations of each method, and exposing what, if any, epistemological, axiological, and ontological assumptions underlie each method. To investigate these methods in these ways can help practitioners and scholars. For example, such investigations may help practitioners determine when one method is better suited than another for a given design project, and they may help scholars inform and advance any of these methods through theory and research.

6. ACKNOWLEDGMENTS

The author would like to thank the 2009 ACM SIGDOC program Committee for its review of and comments on this paper's proposal and first draft.

7. REFERENCES

[1] Nielsen, J. 1993. Usability Engineering. AP Professional.

[2] Booth, P. 1989. An Introduction to Human-Computer Interaction. Lawrence Erlbaum Associates.

[3] Norman, D. 1983. Design rules based on analyses of human error. Comm. of the ACM. 26, 4 (Apr. 1983), 254-258. DOI= http://doi.acm.org/10.1145/2163.358092.

[4] Card, S., Moran, T., and Newell, A. 1983. The Psychology of Human-Computer Interaction. Lawrence Erlbaum Associates.

[5] Bleser, T., and Foley, J. 1982. Towards specifying and evaluating the human factors of user-computer computer interfaces. Proc. ACM Conf. Human Factors in Computer Systems. ACM Press, New York, NY, 309-314. DOI=http://doi.acm.org/10.1145/800049.801801

[6] Norman, D. and Draper, S. 1986. User Centered System Design: New Perspectives on Human-Computer Interaction Lawrence Erlbaum Associates.

[7] Beyer, H., and Holtzblatt, K. 1998. Contextual Design: Defining Customer-Centered Systems. Morgan Kaufmann.

[8] Bødker, M., and Nielsen, J. 2008. Vision Labs: Seeing UCD as a relational Practice. Proceedings of the 20th Annual Australasian Conf. on Computer-Human Interaction: Designing for Habitus and Habitat. (Cairns, Australia, December 08 - 12, 2008). ACM Press, New York, NY, 283-286. DOI= http://doi.acm.org/10.1145/1517744.1517781

[9] Tullis, T., and Albert, B. 2008. Measuring the User Experience: Collecting, Analyzing, and Presenting Usability Metrics. Series: Interactive Technologies. Morgan Kaufmann.

[10] ISO 13407: Human-centered Design processes for Interactive Systems. 1999. Geneva: Int'l Standards Organization.

[11] Maguire, M. Methods to support human-centred design. 2001. Int. J. Human-Computer Studies. 55. 587-634. DOI=10.1006/ijhc.2001.0503

[12] Kuniavsky, M. 2003. Observing the user experience: a practitioner's guide to user research. Morgan Kaufmann.

[13] McDowell, E. 1991. Interviewing Practices for Technical Writers. Series: Technical Communications. Baywood Publishing Company, Inc.

[14] Nardi, B. 1996. Context and Consciousness: Activity Theory and Human-Computer Interaction. MIT Press.

[15] Engeström, Y., Miettinen, R., and Punamäki, R. 1999. Perspectives on Activity Theory. Series: Learning in Doing: Social, Cognitive, and Computational Perspectives. Cambridge University Press.

[16] Vygotsky, L. 1978. The Mind in Society. Harvard University Press.

[17] Leont'ev, A. 1981. Problems of the Development of the Mind. Progress Publishers.

[18] Gay, G., and Hembrooke, H. 2004. Activity-Centered Design: An Ecological Approach to Designing Smart Tools and Usable Systems. Series: Acting with Technology. MIT Press.

[19] Norman, D. 2005. Human Centered Design Considered Harmful. Interactions. 12, 4 (July + Aug. 2005), 14-19. DOI= http://doi.acm.org/10.1145/1070960.1070976

[20] Hoekman, R. 2007. Designing the Obvious: A Common Sense Approach to Web Application Design. New Riders.

[21] Norman, D. 2006. Logic versus usage: the case for activity-centered design. Interactions. 13, 6 (Nov. + Dec. 2006), 45, 63. DOI= http://doi.acm.org/10.1145/1167948.1167978

[22] Spinuzzi, C. 2005. "Don Norman and 'Activity-Centered Design.'" Website: Computer Writing and Research Lab, Dept. of Rhetoric, UT Austin. (Aug. 12, 2005). http://www.cwrl. utexas.edu/node/180

[23] Dougherty, J., and Keller, C. 1985. "Taskonomy: A practical approach to knowledge structures." In J. Dougherty (Ed.), Directions in cognitive anthropology. 161-174. Univ. of IL Press.

[24] Otwell, A. 2005. "Activity Theory and User-centered Design." Blog: heyblog: A Space for half-formed Thoughts. (July 29, 2005). http://www.heyotwell.com/heyblog/archives/2005/07/activity_theory.html

[25] Chi, T. 2005. "Activity Centered Design." Website: ok/cancel. (Aug 5, 2005). http://www.ok-cancel.com/archives/2005/08/activity-centered-design.html

[26] Cooper, A., 2007. Reimann, R., and Cronin, D. About Face 3: The Essentials of Interaction Design. Wiley Publishing, Inc.

[27] Goodwin. K. 2009. Designing for the Digital Age: How to Create Human-Centered Products and Service. Wiley Publishing, Inc.

[28] Yahoo! Developer Network Design Pattern Library. 2009. Website. http://developer.yahoo.com/ypatterns

[29] Schweikardt, E. 2009. User Centered is Off Center. Interactions. 16, 3 (May + June 2009), 12-15. DOI=http:// doi.acm.org/10.1145/1516016.1516019

[30] Constantine, L. 2004. Beyond User-Centered Design and User Experience: Designing for User Performance, Cutter IT Journal. 17, 2. (Feb. 2004.)

[31] Constantine, L. 2006. Designing Web Applications for Use. Website: User Interface Engineering. (Dec. 11, 2006). http://www.uie.come/articles/designing_web_applications_for_use/

[32] Rubin, J., and Chisnell, D. 2008. Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests. Wiley Publishing, Inc.

[33] Hutchins, E. 1995. Cognition in the Wild. MIT Press.

[34] Christensen, H.B., and Bardram, J.E. 2002. Supporting Human Activities: Exploring Activity-Centered Computing. UbiComp 2002: Ubiquitous Computing. Series: Lecture Notes in Computer Science. Springer.

[35] Gifford, B.R., and Enyedy, N.D. 1999. Activity Centered Design: Towards a Theoretical Framework for CSCL. Proceedings of the 1999 Conf. on Computer Support for Collaborative Learning. (Palo Alto, CA, December 12-15, 1999). International Society of the Learning Sciences. 22-es.