Personas as a Design Tool for Interactive Systems

Nathalie Barthe

A Thesis

in

The Department

of

Computer Science

Presented in Partial Fulfilment of the Requirements
for the Degree of Master's of in Computer Science at
Concordia University
Montreal, Quebec, Canada

August 2003

Your file Votre référence
ISBN: 0-612-83896-X
Our file Notre référence
ISBN: 0-612-83896-X

ABSTRACT

Personas as a Design Tool for Interactive Systems

Nathalie Barthe

This thesis investigates the concept of personas, which are narrative representations of specific individuals of an interactive system's target audience. Personas aim at involving users in the design process to improve the usability of an interactive system before its development and deployment. This thesis first highlights the differences between design as defined in engineering and interaction design. Interaction design, which has its foundations in the human computer interaction community, is identified as an activity of researching, planning and modeling the software behaviours and services from the user's perspective. After discussion of user involvement in software development, we introduce a 3-steps model of the design process and review interaction design methodologies and their way to "know the users". Models of users' knowledge promoted in those methodologies are exposed and the persona model, rationale and lifecycle are described in details. Personas' uses and benefits are explained within the software engineering lifecycle and interaction design practices. Also, personas' mechanisms and effects are discussed in context of the design activity in terms of innovation, engagement and reflection-in-action. Then, with the help of two studies, a survey and an ethnographic report, personas are found useful and usable for software designers but their contribution in the overall software development lifecycle is questioned. Therefore, the need to formally assess the quality in use of personas and the challenges this tool faces in the software engineering community are identified.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 Introduction

1.1 Motivation and Justification

CHAOS, a survey and case studies report on software projects' failure produced by The Standish Group (1995), states that:

- the number one factor for the success of software projects is **user involvement** (15.9% of responses),
- the number one factor that causes software projects to be challenged is the **lack of user input** (12.8% of responses)
- the number one factor explaining why software projects are impaired and ultimately canceled is **incomplete requirements** (13.1% of responses),
- the number two factor for impaired projects is again **lack of user involvement** (12.4% of responses).

In software product development, marketing research usually brings the users' needs list into the software lifecycle. But obviously, software project success calls for more than market research data. Software practitioners need operational users' involvement and inputs that marketing, which is "primarily concerned with making business decisions, forecasting sales and quantifying business models.", cannot deliver (Gilmore, 2002).

1.2 Research Goal and Methodology

Personas, which are narrative representations of typical users of an interactive system, are becoming very popular as a tool for software design and have been receiving more and more attention in the human-computer interaction community in the past few years.

Personas are a hot topic on mailing lists like *chi-web* or *u-test*, where many designers are writing about their interest and use of this technique. In the "Design corner" of her InContext company's website (http://www.incent.com), Karen Holtzblatt reports that in a recent panel discussing and comparing user-centered design methodologies at the ForUse 2002 conference, all the panellists chose personas as the technique they would use across all methodologies.

As one can see, there is a lot of talk around personas. However, beyond the buzz, what are they all about? Where do they come from? On what grounds are they standing on? Which user input are they bringing to designers, software design methodologies and processes? More specifically, this thesis follows these goals and methodologies.

- From a review and analysis of the formats proposed so far, this thesis looks at how personas are built and what is the information generally included in a persona.
- Using a model of the design process, this thesis discusses how and at which stages of the software design and development process personas are better fitted and useful.
- With the help of an empirical survey with a sample of designers, this thesis explores if personas are a usable concept for software practitioners unfamiliar with usability techniques?

1.3 Thesis Outline

This thesis explores *personas,* as archetypal models of users acting as proxies for user involvement and input in an interactive systems design process. In the first section, the paradigm adopted in this thesis, human-computer interaction, is presented in parallel with

software engineering and software quality. After, interaction design methodologies are reviewed. In the second section are described the design applications of knowledge about users in terms of user profile, context of use, cognitive models and user classes. The third section details the concept of personas, from its origin to its lifecycle. The fourth section looks more specifically at personas as a design tool in the software design practice. It talks about where personas fit within the software lifecycle and how they contribute to the design activity in itself. Finally, this thesis investigates the usability and quality in use of personas as a tool for the design of interactive systems and questions its input in the HCI body of knowledge.

2 Design and User Involvement

What is software design? What is software product quality? What knowledge and activities provide software designers with a basis to create software product of quality? What user involvement or knowledge is necessary for that activity? Those questions are at the heart of software design engineering activities. Many scientific paradigms, from natural sciences to human sciences, have been engaged to explore and bring light to these design engineering questions.

In the next three sections we present the software design theoretical framework that is employed for this thesis on *personas* as a software interaction design *tool*. First, software design and software product quality are defined and discussed in terms of usability and quality in use. Secondly, interaction design for such quality is reviewed from a human-computer interaction (HCI) perspective. Also, the chosen model of the design activity is presented. Third, some HCI design methodologies are summarized. Finally, HCI-oriented models of knowledge about the users are described. This background and related works will then allow the investigation of the persona tool in relation to interaction design activity.

2.1 Software Design

2.1.1 Design in Software Engineering

From the perspective of a waterfall lifecycle model, software design is the development activity taking place between requirements analysis and implementation in order to create software. Software is "all or part of the programs, procedures, rules, and associated

documentation of an information processing system" (ISO/IEC 2382, 1993). The activity of software design can be broken down into two areas: internal design and external design (Löwgren, 1995). *Internal software design* focuses on the conceptualization of software "objects": system interfaces, algorithms, databases, classes, routines, variables, etc. for software construction. Internal design has mostly to do with researching, planning and modeling in terms of software coding. *External software design* focuses on the conceptualization of software "functions and their delivery": behaviors, services, dialogues, information displays, appearance, help documentation, etc. for software achievement. External design has mostly to do with researching, planning and modeling in terms of user-software interaction.

Having made that distinction, we note that the boundaries between internal and external software design are permeable since one necessarily is in need of, and influences the other. This thesis will concentrate only on external software design, called *interaction design*, due to its focus on the design interactive systems.

2.1.2 Software Design Quality

As defined in ISO/IEC 8402 (1986), *quality* is "the totality of characteristics of an entity that bear on its ability to satisfy stated and implied needs". For software, those quality characteristics are: functionality, reliability, efficiency, maintainability, portability and usability. Considering that software users can be developers, maintainers as well as end-users, those quality characteristics applies to both internal and external software design productions. In practice, usability quality is mostly looked at from the end-users perspective. ISO/IEC 9126 (1991), gives a definition for *usability* quality, "a set of attributes that bear on the effort needed for use, and on the individual assessment of such

use, by a stated or implied set of users". The usability attributes are: understandability, learnability, operability and attractiveness. Also in ISO/IEC 9126, the usability quality characteristic is accompanied and extended by the concept of *quality in use*, which is "the capability of the software product to enable specified users to achieve specified goals with effectiveness, productivity, safety and satisfaction in specified contexts of use. It measures the extent to which users can achieve their goals in a particular environment, rather then measuring the properties of the software itself". The quality in use attributes are: effectiveness, productivity, safety, satisfaction. The following figure summarizes those two quality characteristics and their attributes.

Quality Characteristic	Attributes
Usability	*Understandability*: The capability of the software product to enable user to understand whether the software is suitable, and how it can be used for particular tasks and conditions of use.
	Learnability: The capability of the software product to enable user to learn its application.
	Operability: The capability of the software product to enable the user to operate and control it.
	Attractiveness: The capability of the software product to be attractive to the user.
Quality in Use	*Effectiveness*: The capability of the software product to enable specified users to achieve specified goals with accuracy and completeness in a specified context of use.

Quality Characteristic	Attributes
	Productivity: The capability of the software product to enable users to expend appropriate amounts of resources in relation to the effectiveness achieved in a specified context of use.
	Safety: The capability of the software product to achieve acceptable levels of risk of harm to people, business, software, property re the environment in a specified context of use.
	Satisfaction: The capability of the software product to satisfy users in a specified context of use.

Table 1: Usability and Quality in Use software design quality characteristics.

ISO 9126 also proposes metrics to evaluate those "screen-deep" quality characteristics of usability and quality in use. That measurement activity can only be formally done when the software product is complete and running. Unfortunately, this quality assurance activity comes late in the development process, where and when it is costly to fix usability and quality in use defects through redesign. Therefore, what knowledge and activities can help to create a software product that will meet those quality characteristics before-hands? How does a designer (or team of designers) think about, plan for and craft out interaction design for usability and quality in use? This question is at the core of the HCI research field.

2.2 Human-Computer Interaction Design

2.2.1 Definition of Design

In answer to the problem of identifying a single definition of the various forms of design, Archer (1984) proposes examples and counter-examples. He claims that sculpture or mathematical calculation is not design while architecture and page layout would be. For him, design must have a *practical purpose* and not be *conducted mechanically*. The fact that sculpture, or any artwork, goal to "please" the human spirit is not a practical purpose can surely be argued. But, above all the discussions about design as art or science, most agree that design is primarily an activity and should be studied and regarded as *design situations*.

Accordingly, Winograd (1996) writes "Although we label it with a noun, design is not a *thing*. The questions that we can ask fruitfully are about the *activity* of *designing*". And McPhee (1996) describes the consensus among design scientists around the characteristics of design situations:

- Design situations start with a need and require intention.
- Design situations involve transformation.
- Generation of new ideas is fundamental to design situations.
- Design situations must satisfy constraints.
- Design situations involve problem-solving or decision making.
- Design situations result in schemes for artifacts' implementation.
- Diversity and evolution are inherent to design situations.

2.2.2 User's Perspective in Design

Historically and with the pioneering works of J.C.R. Licklider (1960) and D. Engelbart (1962), HCI research has mostly focused on the design of input/output devices, or *human-computer dialogue design.*

Consequently, psychologist and computer scientists were first interested in the design activity of the user interface "ergonomics". Design guidelines, evaluation heuristics, input/output devices standards and ergonomic norms have been elaborated based on cognitive theories, psychological experiments and other scientific observations, like critical incidents case studies for example. In this first generation of HCI research, the one about *usability engineering*, Hansen (1971) proclaimed the initial principle of interaction design that requires to *"know the user"* for good interface design.

With the democratization of computing brought by the revolution of the personal computers that happened in the late '70s - early '80s, HCI became a field of interest for many researchers from the human sciences and developed into a multidisciplinary science. Inspired by behaviorists, sociologists and anthropologists, this movement gave way to the second generation of HCI, the one of *participative approaches.* Those approaches, like participatory design and contextual inquiry, promote ethnographic methods of describing work by site visits, task analysis and users' involvement in design. Participative approaches want to account for the impact that technology have on people and the influence that people have on technology. One can summarize saying that in regards to the first principle of interaction design, the HCI community shifted its focus from knowing the user's "black box" profile for interface design to the requirement to know the users' classes and work usages for good users/system interaction design.

Around the '90s, a direction in the HCI multidisciplinary research field was to investigate interaction design from the designer's point of view. The same way that some HCI researchers had been inquiring the role of users' knowledge and expertise for interaction design, they are now investigating the designer's knowledge and expertise for interaction design practices. This third HCI generation, *designer as information architect,* was encouraged by the rise of constructivist theories in cognitive and educational sciences. A research example of such a perspective is the famous experiment made by Tullis (1993). In this study on the "common sense" aspect of interaction design, Tullis asked 28 experienced programmers to rank seven user interfaces for a task from "best" to "worst". Tullis, who had beforehand empirically tested the interfaces, showed that there was no correlation between the programmers' rankings and the users test data. He concluded that, at least for those user interfaces and that particular task, good interaction design was not just "common sense".

Consequently, some members of the HCI community advocated for the importance to identify, capture and model design rationales of, and for, interaction design (Moran & Carroll, 1996). So, interaction design methodologies were put forward where knowledge about users and HCI theories are packaged in a way to promote the acquisition by designers of sound users' constructs for usability and quality in use.

The next three sections talk about users' involvement in the design process, present a model of such a process for interaction designers and summarize proposed interaction design methodologies.

2.2.3 User Involvement in the Design Process

As reviewed by Kujala (2003), user involvement in the software design and development of usable systems is a widely accepted principle. The benefits of this principle are:

"1. Improved quality of the system arising from more accurate user requirements.

2. Avoidance of costly system features that the user did not want or cannot use.

3. Improved level of acceptance of the system.

4. Greater understanding of the system by the user resulting in more effective use.

5. Increased participation in decision-making within the organization."

But when looking at different applications of user involvement in software design and development (would it be field studies, qualitative or quantitative research), it appears quickly that this approach can be very demanding and challenging for software practitioners and managers. User involvement does not come easy and pleasant for people untrained in the "human sciences" disciplines required handling such a collaborative work input. Moreover, the reality of the time and costs constraints of software projects often annihilates the possibilities for user involvement. Therefore, personas, acting as proxies for users' involvement in the design and development process, can be viewed as an interesting way to address those challenges. These two points are further discussed in section 4.

2.2.4 Model of the Designer's Process for Design

When talking about design and software development, Löwgren (1995) presents a model of the design process "intended to facilitate the understanding of what happens in design and how designers use different kind of knowledge". This model was developed

by the design methodologists Lundequist & Ullmark (1993). Those architects and researchers in the design methodology field analyze design in a descriptive and normative theoretical framework based on a creative design perspective. The following three-step design process model will later be used in this thesis to discuss personas as a tool for interaction design.

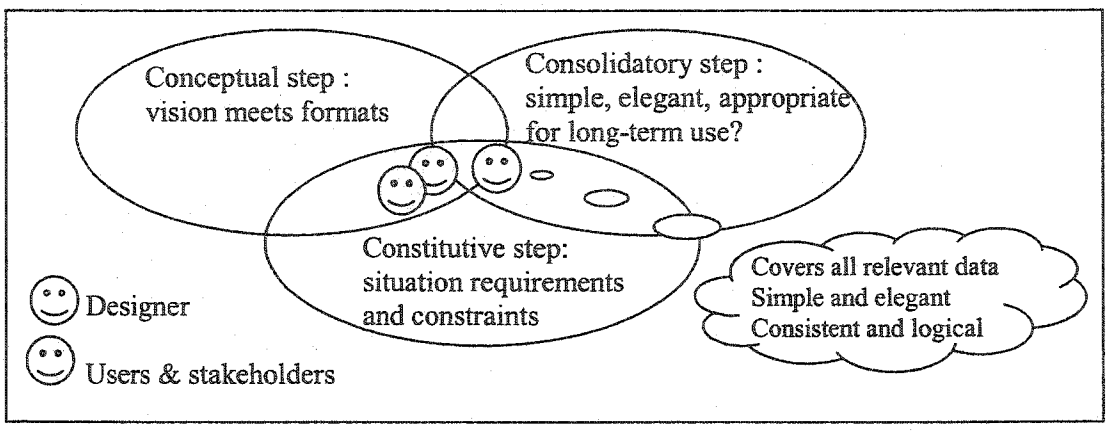


Figure 1: Lundequist & Ullmark design process model

The three qualitative, non-sequential, steps of the model consist of:

- a *conceptual* step, resulting in "formats" (design solution constructs) guided by the designer's visions based on known structures and user's needs and values;
- a *constitutive* step, where a "format" is elected, modified or extended to fit the design situation requirements and constraints;
- a *consolidatory* step, in which the designer professionally judges and refines the selected "format" for design qualities like simplicity, elegance, consistency, coverage, generality and flexibility.

When talking about the role of users (or users' knowledge) from an HCI view of this model, Löwgren (1995) remarks that in the conceptual step, users help the designer to get involved in the future use of the design artifact. He notes that this social route of sharing

values and viewpoints cannot automatically guarantee formats' appropriateness. He insists on the importance of the designer's skill for that step. In the consolidatory step, Löwgren argues that users can play a more active role. Since design concepts are available in that step, users and designer can co-analyze, discuss and choose an appropriate format. He adds that this process can carry some difficulties if concepts communicativeness is deficient, if time for exchanges is limited and participants lack social skills. Finally, Löwgren asserts that the consolidatory step is a genuinely professional activity where the chosen format is detailed with "proven" implementations, state-of-the-art formulas and best practices. He says that studies of users' involvement in that step have shown no benefits since users feel sidestepped and puzzled by the inevitable evolution of "their" solution.

2.3 Interaction Design Methodologies

2.3.1 User-Centered Design

User-centered design (UCD) methodology emerged in the '80s has an answer to the increasing acknowledgment of the poor usability of interactive systems; and that, even if designers were following ergonomic standards and guidelines. Norman & Draper (1986) coined the term "user-centered design" and Gould (1988) articulated the methodology. According to Gould, UCD originates and is executed following these four principles:

1) *Early and continual focus on users*, through direct contact to understand cognitive, behavioral, attitudinal, and anthropometric characteristics of users and their jobs.

2) *Integrated design*, where all aspects of usability evolve in parallel and under one focus.
3) *Early and continual user testing*, to qualitatively and quantitatively measure performance of intended users doing real work with simulations and prototypes.
4) *Iterative design*, where an interactive system in progress is modified based upon the results of user testing cycles.

Gould proposes the following methods to carry out the first principle of focus on the user: talk with users, achieve direct contact, visit customer locations, observe and videotape users working, learn about the organization, try it yourself, have users or experts involved in the design team, do task analysis, administer surveys and questionnaires, test behavioral target goals.

Sometimes called *usability engineering* (Nielsen, 1993), UCD has been generally accepted as an efficient and cost-justified methodology. But that methodology can be complex to put in practice and manage since it demands broad "human" skills, lengthens the analysis-design lifecycle phase and raises development management challenges. Other criticisms of UCD include the inherent problems of subjective versus objective observations and measurements, the methodological requirement for research and experimentation and the possible oversimplification of the user viewed as an automaton outside of a social context (Adler & Winograd, 1992).

Consequently, some HCI researchers advocated for interaction design methodologies based on socially-situated user involvement (Whiteside & al., 1988) like participatory design and contextual design. Others promote methodologies closer to object-oriented software engineering like usage-centered design and scenario-based design. Those

methodologies are discussed next. All those different approaches take into account users' experience and aim at designing with users to respect and enhance that experience.

2.3.2 Participatory Design

Participatory design (PD) is a socio-technical approach to interactive system development that first took place in Europe, especially in the Scandinavian workplace democracy movement. In a seminal book of the PD approach, Ehn (1988) emphasizes the opportunities and constraints of computer artifacts designed for industrial democracy and social responsibility. PD can be viewed as politically grounded in an objection to potential disrespect of work automation systems towards workers and unions. Therefore, PD methodology promotes design as a social and creative activity to anticipate and build alternative futures.

From its primary focus on mandatory workers involvement in the design process, PD evolved to include all sorts of users/designers "participative" methods: ethnography, action research, contextual inquiry, co-development, cooperative evaluation, participatory analysis of usability data, and prototyping, to name a few (Muller, 1993). As one can foresee, PD, like UCD, brings many challenges to the software development process. And even if patent examples of that methodology have been reported in literature, PD did not successfully cross the Atlantic Ocean and opened software development doors. Though, one extremely positive outcome of PD is that it helped develop more fast, light and operational ethnographic methods for software design. "Rapid ethnography" recommends identifying key informants, conducting concurrent interactive observations and decrypting the field data through collaborative analysis

(Millen, 2000). This approach of ethnography is at the heart of the contextual design methodology described below.

2.3.3 Contextual Design

Contextual design (CD) is a design methodology for customer-centered systems. CD has been devised and is being promoted by Holtzblatt & Beyer (1998) as an approach to bring customer (clients and users) data into design through a defined sequence of activities. CD is a result of the authors' on-going experiences of interaction design and a response to a challenge raised by J. Whiteside to "design a process that would lead to new kinds of systems rather than iterating on existing systems" as prototyping and usability testing do (Holtzblatt and Jones, 1995).

CD consists of five sequential activities:

1) *Contextual inquiry*; which is based on ethnographic data, gathered by one-to-one interviews with customers and followed by team interpretation sessions. It results in a customers-designers common understanding and view of the interactive system to be designed.
2) *Work modeling*; which consists in detailing the work of each customer interviewed in a diagrammatic representation. Work models are of five types: flow, sequence, artifact, cultural and physical.
3) *Consolidation*; which brings through affinity diagramming, all work models together showing the underlying pattern and structure of the work that the design must address.

4) *Work redesign*; which uses scenarios and storyboards so a design team can devise a vision that improves customers' work. That work structure reengineering is based on customers' issues rather than available ad hoc technical solutions.

5) *User environment design*; which formalizes the interactive system functions and work objects as "focus areas". Focus areas are in turn formalized in terms of purpose, functions, links to other focus areas and interaction objects. Focus areas can also include constraints (like speed, reliability or availability) and open design issues (like UI ideas, implementation concerns or quality requirements).

The following figure shows an example of focus areas for user environment design in the case of a slide presentation interactive system like Microsoft PowerPoint.

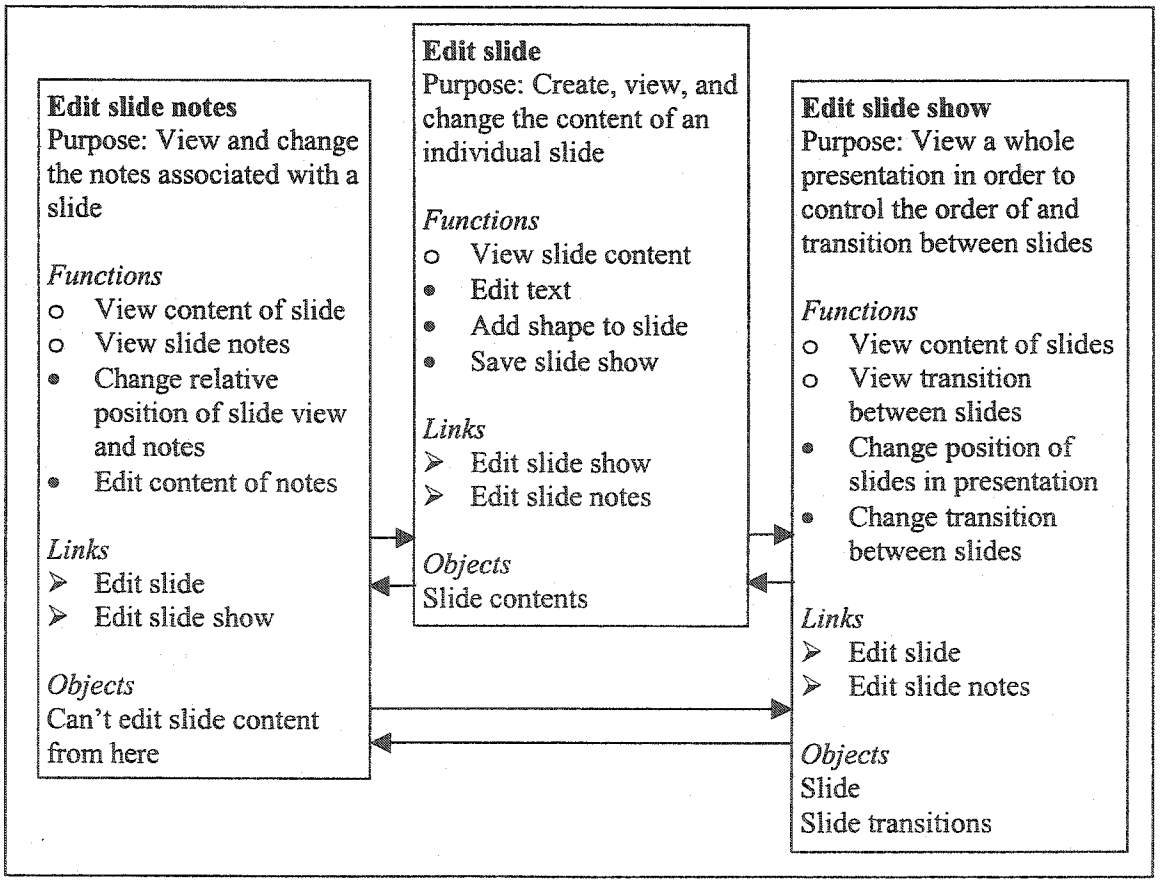


Figure 2: Example of user environment design focus areas

Only after the user environment design is completed that system mock-ups can be created, tested with customers and refined giving way to implementation works.

As shown, CD is a thorough approach to interaction design embracing business, organizational, work and users considerations and knowledge. According to the promoters of this approach, CD answers to the challenge of coupling usability engineering and innovative design. Holtzblatt (2001) assert that CD allows innovative design since this methodology conveys invention planning through recombination and renovation of known material across contexts to achieve customers' intents.

Another attempt to answer to that challenge is usage-centered design described below.

2.3.4 Usage-Centered Design

Also, pinpointing the limitations of UCD, Constantine & Lockwood (1999) propose a design methodology called usage-centered design (UsageCD). According to them, UsageCD differentiates itself from UCD by a focus on usage instead of users because "user studies too easily confuse what users want with what they truly need" (Constantine, 2001). Citing Parush (2001), Constantine also argues that "rapid iterative prototyping is too often a sloppy substitute for thoughtful and systematic design. And usability testing is a relatively inefficient means of finding problems that could have been avoided through proper design". The following figure compares differences between UCD and UsageCD as advocated by Constantine & Lockwood (2001).

User-Centered Design	Usage-Centered Design
▪ Focus on users: user experience and user satisfaction	▪ Focus is on usage: improved tools supporting task accomplishment
▪ Driven by user input	▪ Driven by models
▪ Substantial user involvement	▪ Selective user involvement

User-Centered Design	Usage-Centered Design
- User studies - Participatory design - User feedback - User testing ▪ Design by iterative prototyping ▪ Highly varied, informal, or unspecified processes ▪ Design by trial-and-error, evolution	- Explorative modeling - Model validation - Usability inspections ▪ Design by modeling ▪ Systematic, fully specified process ▪ Design by engineering

Table 2: Comparison of user-centered and usage-centered design

Accordingly, UsageCD addresses the whole software design process and promotes concurrent design activities of aesthetic and graphic design (called creative design), visual and interaction design and "internal system" design of class models and sequence diagrams. The following figure by Constantine (2000) illustrates that process. It is interesting to note that a document of "user & customer profiles" appears in the model though user studies are not supported in this methodology.

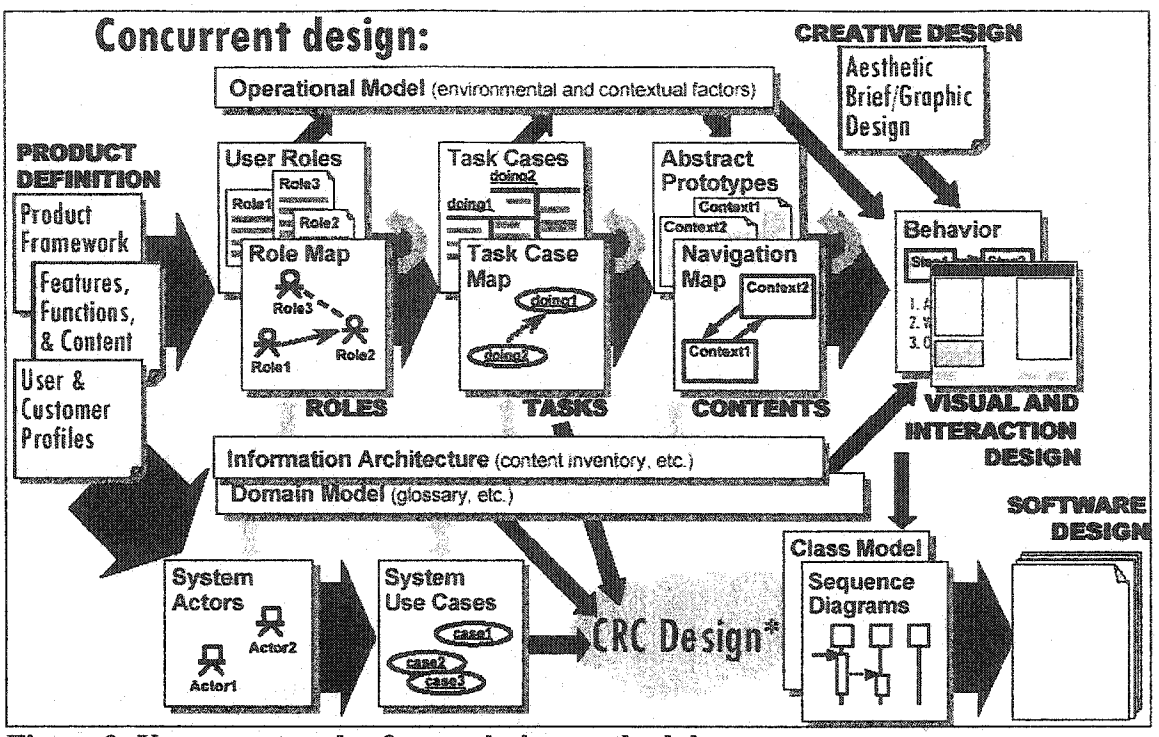


Figure 3: Usage-centered software design methodology

As said previously, UsageCD is driven by modeling with selective user involvement, called "joint essential modeling", and extends Unified Modeling Language (UML) with three abstract models: *user model*, *task model* and *content model*.

The user model is a declination of the "actor" model. It distinguishes between system actors and human actors. The latter is exploded into structured *user roles* and *user role maps*. User roles are abstractions of the users' needs, characteristics, behaviors, interests, expectations and responsibilities that model the relationship a user has with the system. A user role map shows the relationship of affinity, specialization or composition between user roles. Interestingly, Constantine has written on the topic of personas in one of his company newsletter (2002). He says that "a persona is a plausible personification of the archetype or ideal represented by the Role".

The task model consists in a set of *task cases* (also called essential use cases) and *task case maps*. Similar in form to use-cases, task cases are abstract and technology-free description of the user-system interaction steps for a task. Task case maps put in a diagram the relationships between task cases. On his company website (http://www.foruse.com), Constantine & Lockwood give an example that shows the difference between a use case and a task case.

Use case for: Increasing temperature	
USER ACTION	SYSTEM RESPONSE
1. Press "program" button	
	2. Display first program option
3. Step to "temperature"	
	4. Display current temperature setting
5. Press "increase" until desired setting is displayed	
6. Press "program" button	
	7. Resume normal operation

Task case for: Increasing temperature	
USER INTENTION	SYSTEM RESPONSIBILITY
1. Increase temperature to desired setting	
	2. Confirm new setting and resume normal operation

Table 3: Examples of use-case and task case

The content model represents the user interface materials and their organization to support the task. User interface materials are called *abstract prototypes* and the flow between those is the *context navigation map*.

Though scenarios are not stated in the literature consulted on UsageCD, they are a technique implied in this methodology as they are used in UML. An explicit method using scenarios is described below.

2.3.5 Scenario-Based Design

Scenarios are used in many software design methodologies from requirements gathering and analysis to testing phases. A formal methodological approach, scenario-based design (SBD), is proposed by Carroll (2000) who did extensive work on the topic of design methodologies and processes from the 90's to today. Carroll identifies six properties that make design difficult:

- Incomplete description of the problem to be addressed
- Lack of guidance on possible design moves
- The design goal or solution state cannot be known in advance
- Trade-offs among many interdependent elements
- Reliance on a diversity of knowledge and skills
- Wide-ranging and ongoing impacts on human activity

For him, since software transforms human activity and is constrained by the contexts of that activity, stories about people behaviors and actions (called *scenarios of use*), provide a direct approach to analyze and design software by making "use" explicit. That way, SBD help analysts and designers uncover their assumptions about people and their tasks that usually end up implicitly in systems and applications. At last, designing for scenarios' characteristics (*setting*, *agents* or *actors*, *goals* or *objectives*, *actions* and *events*) makes "use" the purpose of the software and addresses usability and quality in use as well as the designer activity's inherent challenges. The following figures summarizing SDB are taken from Carroll (2000). They illustrate a scenario of use, how SBD attends to challenges of interaction design and the role of scenarios of use in interaction design for usability and quality in use.

Harry is interested in bridge failures; as a child, he saw a small bridge collapse when its footings were undermined after a heavy rainfall. He opens the case study of the Tacoma Narrows Bridge and request to see the film of its collapse. He is stunned to see the bridge first sway, then ripple, and ultimately lurch apart. He quickly replays the film, and then selects the associated course module on harmonic motion. He browses the material (without doing the exercises), saves the film clip in his workbook with a speech annotation, and enters a natural language query to find pointers to other physical manifestations of harmonic motion. He moves on to a case study involving flutes and piccolos.

Figure 4: Example of scenario of use

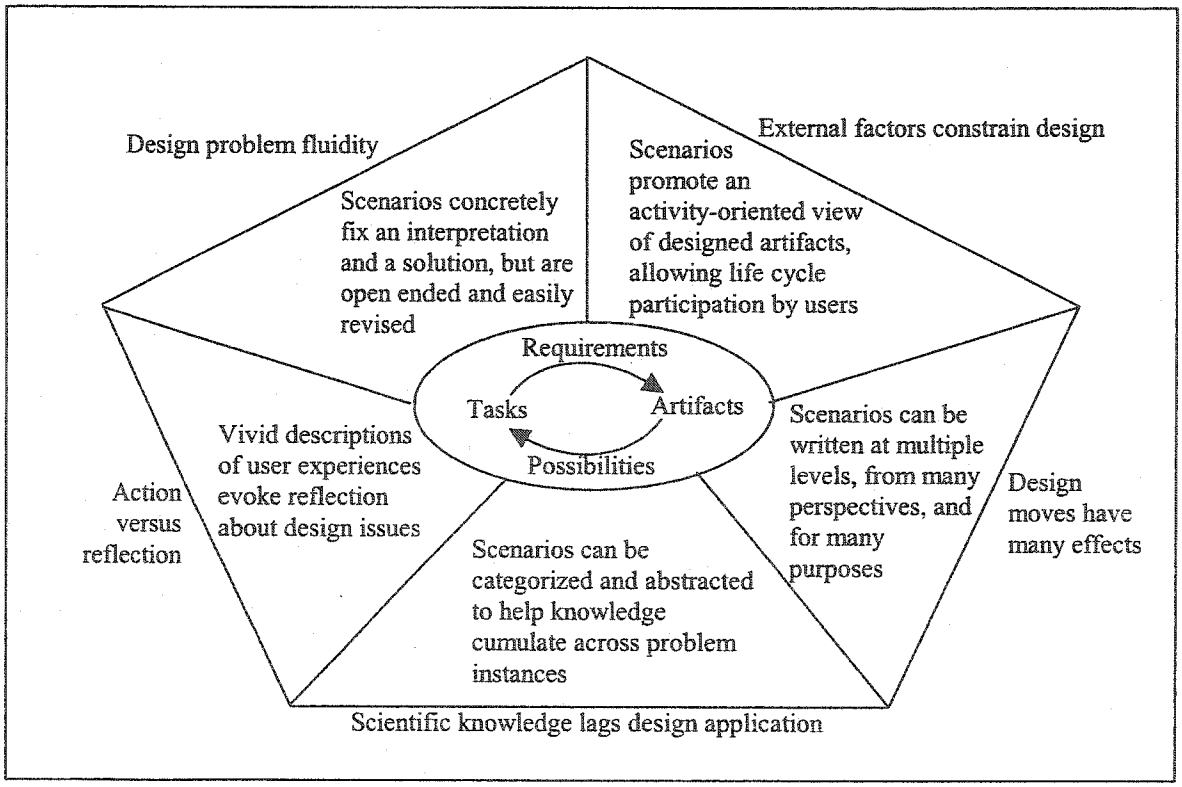


Figure 5: Scenario-based design and interaction design challenges

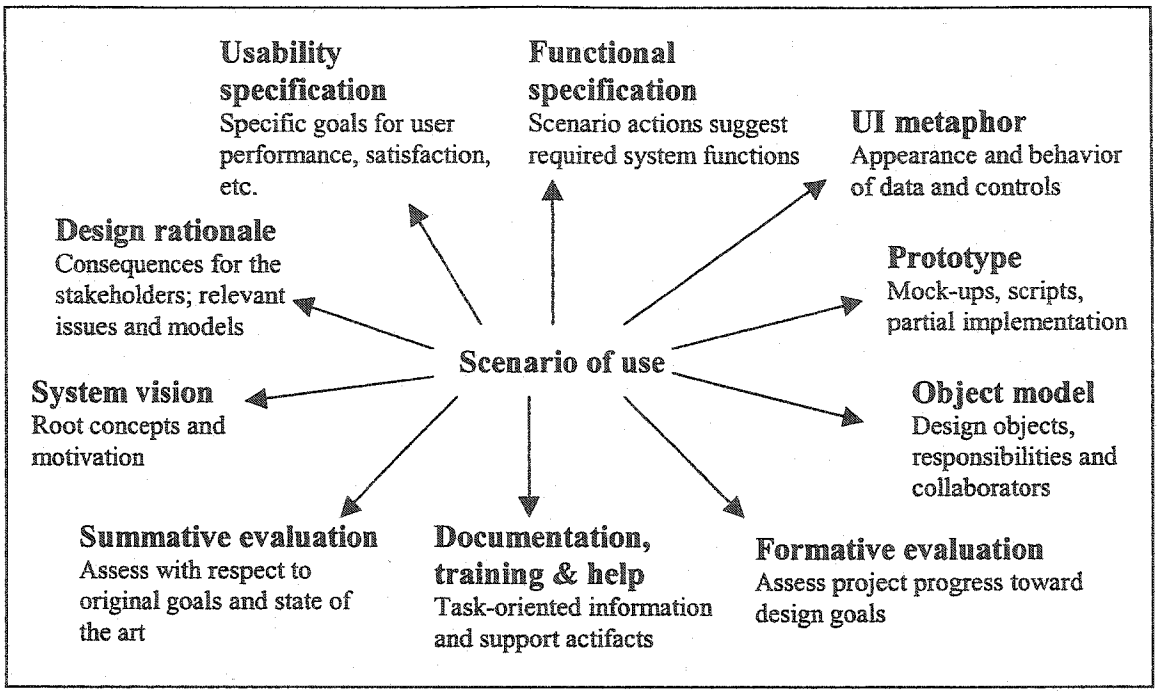


Figure 6: Scenario of use role in interaction design for usability and quality in use

Operationally, Carroll suggests that scenarios can come from many sources: ethnographic field study, participatory design, reuse of prior analyses and scenario typologies, theory-based scenarios, technology-based scenarios and transformations (brainstorming). Scenarios are elicited and detailed until they make significant design issues apparent and allow *claim analysis*. Claim analysis encapsulates HCI knowledge and theories and is the process from which the design decisions and trade-offs emerge. Claim analysis activity results in the overall design rationale. Claim analysis is facilitated by:

- scanning for causes and effects,
- participatory analysis,
- systematic questioning,
- questioning stages of action reuse of prior analysis,

- theory-based claims, and
- transformations (scenarios or claims refactoring from new data).

As one can see, SBD is a powerful yet flexible design methodology that can be integrated in any software design methodologies and development lifecycle. As an example, scenarios are part of contextual design, usage-centered design and goal-directed design that will be described in the next section. Although, Carroll (2000) admits that the practice of SBD is still to be appraised with evidences of how the SBD techniques work.

2.3.6 Goal-Directed Design®

Goal-directed design® (GDD) is a design methodology promoted by Alan Cooper, the father of Visual Basic. Almost no literature is available on the methodology since it is rather new and knowledge of it is acquired by attending GDD practicum at Cooper Interaction Design Inc. But in his book, The Inmates are Running the Asylum (1999), Cooper gives an overview of GDD and its main idea of designing for *goals* instead of tasks, since "goals are the reason why we perform tasks". He adds that "The goal is a steady thing. The tasks are transient." He means by that a goal is an end condition, while the tasks to achieve the goal are process-based and dependant of technological tools available at a particular time. Therefore, designing for tasks does not guarantee the satisfaction of the user's goal. Cooper also claims that "The essence of good interaction is devising interactions that let users achieve their practical goals without violating their personal goals.". Consequently, it is important to identify user's "true" goals and differentiate them from "false goals". The following table gives some examples of goals.

"True" goals	"False" goals
Personal goals • Not feel stupid • Not make mistakes • Get an adequate amount of work done • Have fun Practical goals • Avoid meetings • Handle client's demands • Record client's order • Create numerical model of the business Corporate goals • Increase profit and market share • Defeat competition • Offer more products or services • Go public	• Save memory • Save keystrokes • Run in a browser • Be easy to learn • Safeguard data integrity • Speed up data entry • Increase program execution efficiency • Use cool technology or features • Increase graphic beauty • Maintain consistency across platforms

Table 4: Cooper's "goal stack"

Calde & al. (2002) gives a "nutshell" view of GDD methodology. GDD consists of four phases: research, model, envision and refine. In the *research* phase, ethnographic methods (mostly stakeholders' one-on-one interviews), product marketing and literature reviews are conducted. The *model* phases results in domain and user models in the form of personas. This activity is detailed in later sections. The *envision* and *refine* phases use scenario-based design, interaction design principles and patterns to design an encode solutions in a behaviors specification document.

A final remark on Cooper's GDD is that it does not include any user testing because "in this fast-moving, high-tech world, after it is built, it ships. Post-facto testing cannot have

much influence on the product."(2002). Cooper recognize that testing before programming could be valuable but he notes that those pre-facto tests are skewed by the experimental setting and testers' biases. "You can get a lot about users' reactions from a paper puppet-show, but without doing design first, what gets tested can still be quite inappropriate."(2002). Finally, Cooper advocates for good interaction design before programming; and agrees that usability testing can be useful to demonstrate this necessity to recalcitrant software professionals and managers.

In GDD, identifying and modeling knowledge about users and their goal is central. This is done through the devising of *personas* acting as user proxies for interaction design. The next section reviews more "traditional" procedures of users' and customers profiling to introduce the persona tool and technique that is detailed in section 3.

3 Users' Knowledge in Software Design

In his classic book, The Psychology of Everyday Things (1988), Norman talks about how "knowledge in the head" and "knowledge in the world" influences human behavior and performance. He talks about how that knowledge should be taken into account by designers and should influence design. In that line of thinking, the HCI community traditionally advocated to elicit and model that information on users to reveal design "constraints". This section reviews the most common theories and models developed in HCI to identify, describe and apply that awareness of the knowledge of and about users. First, knowledge of and about the individual user, the *user's profile*, is exposed. Second, knowledge about the context of use is described. After, cognitive models of users are presented. And finally, knowledge about groups of users, the *users' classes*, is discussed from HCI and marketing perspective.

3.1 User Profile

3.1.1 Physical characteristics

For interaction design, it is obviously primordial to take into account the physical characteristics of the user: handedness, color-blindness, disabilities, age and sometimes gender. Design implications naturally derive from those characteristics, for example, a government website on tax deductions for senior citizens should not display its information in small fonts that are not legible for visually-impaired people.

3.1.2 Motivation and attitude

User motivational and attitudinal characteristics necessarily impact his or her adoption, performance and appreciation of an interactive system. Opinions about computer, about technology and automation at large, concerns about confidentiality and privacy, fears or excitement for new ways of doing things, willingness and pleasure to learn, are all examples of user's psychological characteristics that have been studied for interaction design. In task-performance system, mandatory versus discretionary use is also an element influencing user's motivation and attitude towards interactive system. When use of a system is mandatory, the user will strive for job performance benefits and readily available help and documentation to balance stress, anxiety or boredom caused by the introduction of a new system. On the other hand, when use is discretionary, a user will need first-time ease of use, ease of learning and power to be designed into the system or he will simply not use it (Mayhew, 1992).

3.2 Context of Use

3.2.1 Environmental characteristics

The physical and cultural environment of the user affects interactive system's use. Physical environment characteristics like urban or rural region, time zone, telecommuting and mobility, dust/wind/noise levels, lighting, workspace dimensions and layout, furniture and equipments have all to be considered from an ergonomic point of view in interaction design. The user's physical environment may dictate collaboration, privacy or security concerns and constraints.

Also to take into account is the user's cultural environment. User's cultural environment characteristics like language(s) spoken and written, level of education and literacy and aesthetic, color or pictogram codes, even religion and society beliefs, all have an impact on how a user will decode an interactive system for use.

Also important to take into account when designing enterprise-wide systems are corporate cultural variables. Explicit corporate policies or tacit customs (turnover rate and promotion rules, for example) are too often overlooked in interaction design. This doing can give an explanation for failures in business interactive systems' adoption.

3.2.2 Jobs and tasks

For the design of work-supporting systems, one approach to "know the users" is job or task analysis. This approach is based on observation and modeling of job description and task procedures of workers categories, often similar to job tiles: accountant, secretary, call-center agent, etc.

Job analysis activities are guiding the optimization, adaptation, organization and training elaboration in order to reengineer the job. Those analysis activities have multiple objects of research, methods and modeling techniques like site visits, gap analysis, GOMS (Goals, Operators, Methods, and Selection rules), hierarchical task analysis, etc. Those activities can be grouped and classified based on their goals.

- Work analysis: study of the work conditions and context.
- Prescribed task analysis: study of the task to be done. Focus is put on the "what" with no regards to the "how" the task has to be done or how it is already done.
- Effective task analysis: study of "how" the task should be done.

- Real task analysis: the "what" and "how", here and now.

Attempts to classify users based on generic job and task description have not taken off because of the diversity of jobs, tasks and work contexts. No two accountants do exactly the same job and tasks. Therefore, task analysis remains a powerful, though time and resource-demanding, tool for interaction design. Job and task analysis is an ad hoc tool that needs to be re-enacted at each design. Furthermore, Diaper (1987) notes that "Current practice in a task is frequently tied to the existing technology employed in the task and it is therefore difficult to produce creative, novel solutions to system design based on such methods".

3.3 Cognitive Models

3.3.1 Human information processing

The human brain, like a computer, processes information. Human information processing has been, and still is, studied in order to maximize the performance of the human-computer duo. Models of human information processing have been used to think about tasks allocation between humans and computers and guide interface design. The following figure presents a model of human-information processing (Gagné, 1988).

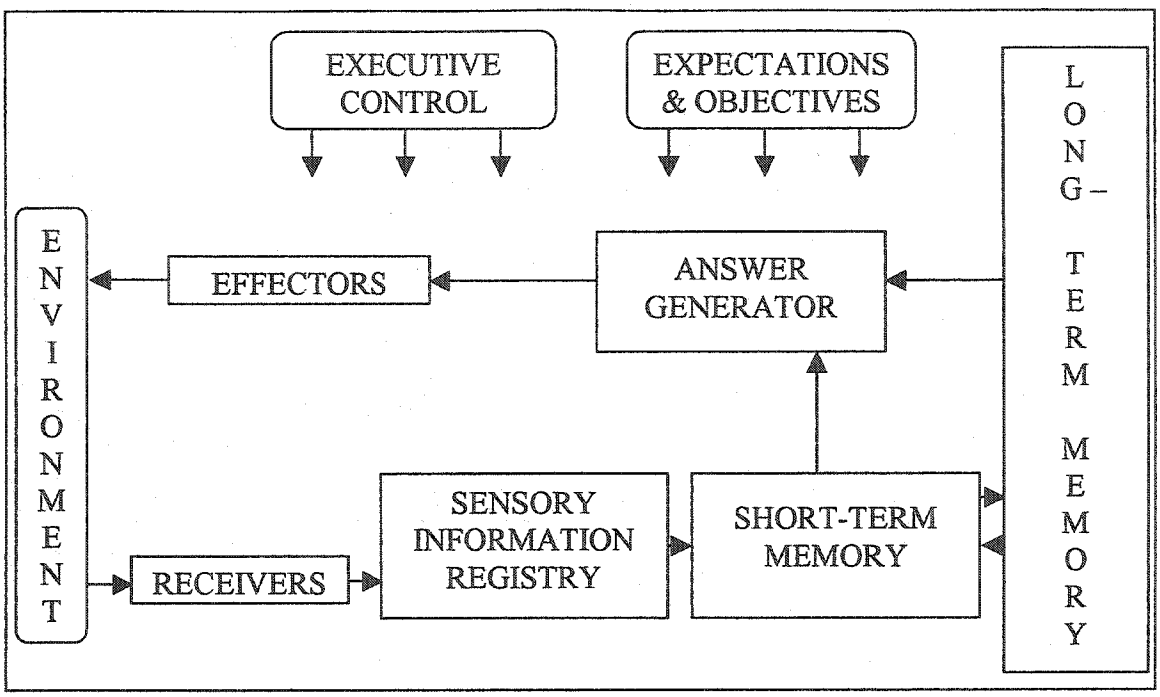


Figure 7: Human-information processing model

The receivers, the five senses, get stimuli from the environment and transmit those electro-chemical impulsions to the sensory information registry (SIR). The SIR is the entrance hall to the human information processing system. It seems that there would be as many entrance halls as types of information. The storage capacity of SIR is huge but waiting time is very short (1/4 second) and competition for transmission is high. Therefore, some sensory information can be lost in time or masked by new incoming information. Information selection and identification is done in the short-term memory (STM), also called working memory. The STM is an electric memory where conscious cognitive operations are done. STM storage and processing capacity is limited. Miller (1956) said around 7±2 items for a maximum of approximately 15 seconds. To process cognitive operations, STM fetch for information stored in the long-term memory (LTM). For example, when processing "2 x 4 + 2 = ?" in STM, symbol decoding, knowledge about mathematical operations and tables of multiplication are retrieved from LTM to

STM. Then processing and answer encoding occur. If it needs to be returned to the environment immediately, the answer "10" is passed to the answer generator (AG). Else, it is transferred to LTM to be stored and available for future use. LTM is a chemical memory where resides, in a passive form, all the information known by an individual. To be stored, new information incoming from the STM has to be integrated in the organized and structured content of the LTM. Cognitive sciences researchers have postulated many models of knowledge structures and organizations in LTM (see for example: Minsky, 1975; Bower & al. 1979). Discussion on those models of *frames*, *schemas* or *scripts* is beyond the scope of this thesis. But most of those models distinguish between *declarative* knowledge (that structures the "what") from *procedural* knowledge (that organizes the "how"). That information in the LTM is supposed to be stored permanently. Forgetting and memory loss would be an effect of the inefficiency of the information retrieval process caused by physiological problems, lack of remembering cues, information interference and conflicts. Finally, the AG is a processing unit managing the answers coming from STM and LTM. AG stimulates the effectors to deliver motor, verbal, physiological answers back to the environment. Knowledge in the LTM can also be directly transferred to the AG that produces an answer in the form of an automatism. Those answers, and all human information processing operations, are ruled by cognitive strategies that are instigated by the executive control unit. Choice of the cognitive strategies is also motivated by human expectations and objectives and environmental context. The cognitive strategies and implications for interaction design are summarized in the following figure. Nielsen (1993) adds to those recommendations this usability heuristic: "minimize user memory load by aiming at recognition rather than recall".

Strategy	Description	Interaction design implication
Attention	Principal strategy for receivers, attention allows humans to perceive the abundant stimuli coming from the environment. Attention is conditioned by the expectations and objectives and has a maximum threshold. Attention can be guided and called upon.	➢ Attention is attracted by salience (blinking, size, color, etc.). ➢ Attention is diminished by audio or visual confusion. ➢ Attention is reinforced by feedback. ➢ Attention prefers center to of outlines, groups to isolated elements. ➢ Automatisms require less attention than complex or conscious processes.
Selective perception	Principal strategy for SIR in order to enhance speed and throughput, selective attention compares or associates prominent characteristics of sensory information with knowledge in STM fetched from LTM. This process would explain "sensory illusions" like identifying human faces from ink stains, hearing telephone rings from a specific sound frequency. This more or less conscious decoding of sensory stimuli allows an individual to build in STM everything that he knows on the stimuli. This could be a good thing to know when designing visual displays, for example.	➢ Perception is subject to the spatial organization of sensory components. Under the principle of similarity and co-location, well-organized sensory information will be decoded and encoded in a "better" way than disorganized information. ➢ By the principle of emergence, selective perception is promoted by emergence of salient information items (noise level and frequency, "surprise effect", bold, underline, italic, color, etc.). ➢ Efficiency of selective perception is increased by following the compatibility principle of information items related to experiences of the "real world". Compatibility can be based on stereotypes (red = hot, blue = cold), on affordances (like a screw, turn right = open, turn left = close) and on familiar shape recognition (C/-\T , T/-\E).
Chunking	A strategy useful for STM, chunking organizes information items into meaningful "chunks" allowing more information to be stored and processed in the limited capacity STM. Exam-ples of chunking are phone numbers, (3) (1) (0) (2) (3) (5) (5) chunked in (310)-(23) (55)	➢ Designers should organize user interface information into chunks of 3 to 4 elements, preferably meaningful.

Strategy	Description	Interaction design implication
Repetition	Repetition of an information item allows that item to stay longer than the initial 15 seconds in STM. Also, repetition prepares information to be transformed for storing in LTM. An actor memorizing his lines is a good example of that.	➢ Designers should use information redundancy and repetition when appropriate.
Codification	Codification transforms information items in the STM into meaningful groups, stories or images to take less space and be more easily stored in LTM. These groups of information will also have better chances for retention if they are anchored in existing knowledge. Classic examples of codification are mnemonics. A mnemonic is a sentence or rhyme used to enhance memorization like, counting the letters of each word of the sentence "God, may I have a large container of coffee?" gives you the value of pi to 7 places.	➢ Designers should try to give "meaning" to information of the user interface. ➢ Recommendation to use metaphors to support user interaction is grounded in that cognitive strategy.
Search and retrieval	That strategy takes place in the LTM to extract knowledge from it. It uses internal or external cues (comparisons, context, etc.) to recall the desired information. Concentration also helps search and retrieval.	➢ Designers can try to give external cues to promote information search and should be careful not to disrupt concentration with salient information.

Table 5: Cognitive strategies and interaction design implications

3.3.2 Mental workload and human error

As illustrated above, human information processing is powerful but limited in terms of available resources at some point and time. To understand possibilities of human cognitive error and maximize human information processing performance in tasks like problem-solving or decision-making, research on users' mental workload has been conducted and integrated in HCI theories and knowledge.

Mental workload is the relation between the availabilities of human information processing resources versus cognitive and psychic demands. Cognitive demands consist of information decoding/encoding and mental operations. Psychic demands include psychological loads like stress, fear, anxiety or boredom caused by monotony and repetition. Even if this concept is vague and hard to measure and anticipate, some interaction design guidelines, like "provide feedback and clearly marked exits, prevent error", have been suggested in order to maximize the mental work efficiency as well as minimize mental workload. The following figure shows a dimensional representation of cognitive resources availability in terms of time-sharing (Wickens, 1984).

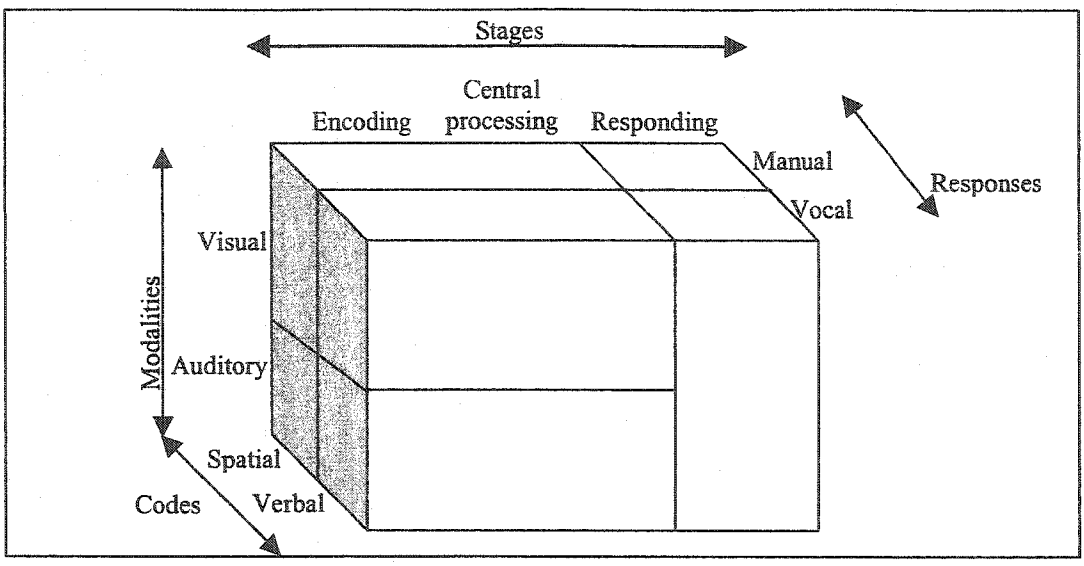


Figure 8: Mental workload time-sharing model

With this model, Wickens suggests that "any two tasks demanding separate 'cells' should yield perfect time-sharing. (...) There may well be a layer of 'general capacity' that is added, like frosting on a cake, to the top and front of the separate resources. These resources, competed for by all tasks, would then prevent perfect time-sharing of all but heavily data-limited tasks." Therefore, it illustrates real-life time-sharing and non-error prone situations like driving a car (where: modality = visual, code = spatial, response = manual) and talking on a mobile phone (modality = auditory, code = verbal, response = vocal) at the same time.

Another important cognitive characteristic that has been studied by HCI researchers is the influence of a user's knowledge of computers and computer applications has on information processing and mental workload performances. Models of this user's "knowledge in the head and in the world" of computers and their applications are called *mental models* and are discussed below.

3.3.3 Mental models

The concept of mental model can be defined as "a representation formed by a user of a system and/or task based on previous experience as well as current observation, which provides most (if not all) of their subsequent system understanding and consequently dictates the level of task performance" (Wilson & Rutherford, 1989). Norman (1986) proposed a distinction between mental models with this classification.

- *Conceptual model*, the designer's mental model of the target system.
- Target *system image*, characterized through displays, documentation, structure and operation.
- *User's mental model* of the system.

Wilson & Rutherford (1989) argue that the system image is not a mental model and should be replaced by another important mental model: the scientist's or designer's mental model of the user's mental model. For clarity purposes, this last one will be called here: the *designer's user model.*

At one point, and due to the popularity of the mental model concept, lots of interaction design guidelines were recommending designing the system image based on *metaphor* of the user's mental model (the metaphor of the desktop, for example). This approach has been criticized because users' mental models are, by essence, incomplete, inaccurate, non-observable, dependant on many factors and they varies greatly between users. Nonetheless, these design guidelines, like "speak the user's language" or "design in consistency with other systems and tasks" (Nielsen, 1993) are generally accepted.

In conclusion of this section, let's summarize saying, like Norman (1988) that "user's knowledge in the head" is extremely efficient but at the high cost of human information

processing and mental workload limitations. While "user's knowledge in the world", which is perceived, interpreted and recognized rather than learned and recalled, may need more processing time but is readily available and easy to use. Therefore, it would be beneficial to focus on uncovering and applying users' "knowledge in the world" in interaction design. Yet, that goal is challenging since users' profiles are as diverse and the number of individual in the interactive system targeted community of users. That's why strategies of classification have been proposed based on concrete and observable knowledge from outside the users "black boxes".

3.4 User Classes

3.4.1 User groups

Based on the assumption that different types of users need different types of interfaces, clustering of users in groups or categories have been recommended by HCI specialists (Nielsen, 1993). Those clusters can be based on attributes like experience, occupation, attitudes, cognitive styles, intelligence and demographic variables such as age and gender. But those grouping methods, being theoretically based and grounded on artificial boundaries from oversimplifying continuous population variables in unitary concepts, do not provide valid design guidelines (Potosnak, 1986).

3.4.2 Expertise

Classification based on expertise is a cost-effective shortcut to lengthy job and task analysis. Dreyfuss (1986) classification of the knowledge that possesses a user of a job, task or existing systems (like computers) as novice, advanced beginner, competent, proficient or expert can surely help designers to decide to include keyboard shortcuts or

not but are useless for "screen-deep" interaction design because classification can be viewed as over-simplification.

3.4.3 Buyer behaviors

Marketers are good at classifying consumers into segments or classes: reluctants, enthusiastics, pragmatics, trend-setters, opportunists, leaders, turnarounds, savers, etc. But whatever the level of sophistication of those classifications, buyer behavior is not that informative on usage behaviors, therefore of not help for interaction designers.

3.4.4 Consumers' technology adoption

Moore (1991) describes a consumers' technology adoption lifecycle that, being of more interest for designers of technology products, does not provide concrete interaction design information either. Moore distinguishes between successive groups of technology adopters and describes strategies so that products "cross the chasm" to reach mass-markets. Interestingly for interaction design, one of these strategies is forcing the marketing and product development teams to focus on specific scenarios of how the product will be bought and used. Implicitly, he advocates for scenario-based interaction design. Moore also talks about the need of more research in marketing and product development "to ensure a successful experience right out of the box". That strategy is obviously congruent with interaction for usability and quality in use. The following figure illustrates Moore's consumer's technology adoption lifecycle and consumer classes that can be useful for the persona tool.

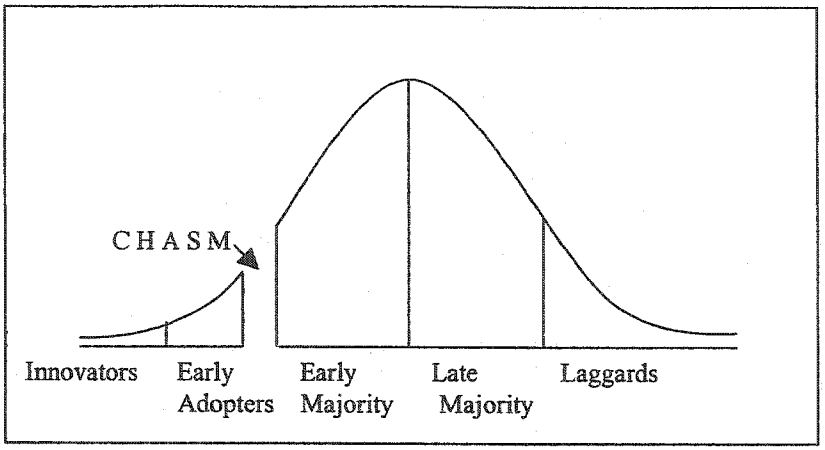


Figure 9: Moore's technology adoption lifecycle

- Innovators are enthusiasts who like and buy technology for its own sake.
- Early Adopters have the vision to adopt an emerging technology to an opportunity that is important to them.
- The chasm is the gap in technology adoption that is not crossed by many software products.
- Early majority pragmatists are the people who do not like to take the risks of pioneering, but are ready to see the advantages of tested technologies. They are the beginning of a mass market.
- Late Majority are pragmatists, who represent about one-third of available customers, they dislike discontinuous innovations and believe in tradition rather than progress. They buy high-technology products reluctantly and do not expect to like them.
- Laggards are traditionalists also who do not engage with high technology products - except to block them. They perform the valuable service of pointing out regularly the discrepancies between the day-to-day reality of the product and the claims made for it.

4 Persona, the User Archetype

The term *persona* is a Latin word meaning "actor's mask" as per Jung's psychoanalytic concept.

> "The persona is a complicated system of relations between individual consciousness and society, fittingly enough a kind of mask, designed on the one hand to make a definite impression upon others, and, on the other, to conceal the true nature of the individual."
> In: C.G. Jung, The Relations between the Ego and the Unconscious (1928).

The following three sections give a detailed definition and illustration of the persona technique and tool, expose its development lifecycle and discuss its use in the interaction design activity and practices.

4.1 Origin: Archetypes in Human Sciences and Marketing

Archetype is a concept used in human sciences to represent, in Plato's words, "element forms of being". Ancient gods and goddesses would be expressions of archetypes. Archetypes have been identified in religion, art, literature, cinema and oral tradition. Examples of archetypes are the *prophet*, the *hero*, the *mother* or *caregiver*, the *magician*, the *outlaw*, the *orphan*, the *emperor* or the *sage*.

In marketing research, archetypal analysis is an alternative to the traditional market segmentation for overcoming consumer heterogeneity. While segmentation makes the implicit assumption that there are several "average" consumers found in the statistical center of each segment, archetypal analysis assumes instead that there are several "pure" consumers (archetypes) who are on the "edges" of the data. All other consumers are

considered to be mixtures of these pure types. Marketing to those archetypes would be efficient because calling upon consumers' "deep structure for human motivation and meaning" (Mark 2002). In a similar way, *users' personas* can be viewed as an archetypal analysis technique to "know the users" for interaction design. The persona tool is the topic of the next section.

4.2 Persona in Design

4.2.1 Definition and Example

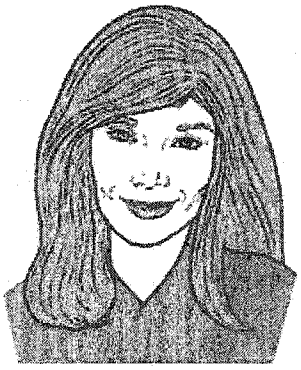
An ancestor of personas can be found in *character maps* presented by Verplank (1993) in an INTERCHI tutorial. Hartfield (1996), co-founder of IDEO industrial design firm, recounts how they use character maps, "detailed personality and activities description for a small set of envisioned typical users", to communicate potential users' observations to designers in order to guide their work. Personas are very close to those character maps.

First introduced by Cooper (1999), the persona is a model of a user, also called a *user archetype*. The persona is a fictional and detailed "real-life" character that captures and represents the behaviors, goals and motivation of a group of actual or potential users of a software product (Calde & al., 2002). The persona focuses on modeling users' personal and practical goals instead of depicting some psychological profile or class. See figure 10 in section1.3.6 for an illustration of those goals. The persona is obtained by analysis of patterns in domain, usage and workflow data gathered through rapid ethnography studies and contextual inquiry research methods.

A persona can be one of five influential types:

- *primary*, which needs are so unique that it calls for a distinct interface form and behavior;
- *secondary*, which needs are going to be fulfilled by the primary interface with minor modification/addition;
- *supplemental*, which needs are completely fulfilled by the primary interface;
- *served*, which is not an actual user but who is indirectly affected by the product and its use;
- *negative*, or *anti-persona*, which is whom designers and developers should not design for.

The following figure, taken from Whitney Quesenbery on her company website (http:www.cognetics.com) illustrates a example persona.

<table>
<tr><td rowspan="4">  </td><td>Favorite quote:
"A designer knows he has achieved perfection not when there is nothing left to add, but when there is nothing left to take away." Saint-Exupery</td></tr>
<tr><td>Favorite design tool:
The whiteboard - or anything that lets me iterate the design quickly.</td></tr>
<tr><td>Member of:
SIGCHI, UPA and a local usability discussion group</td></tr>
<tr><td></td></tr>
<tr><td>Name:</td><td>Linda</td></tr>
<tr><td>Title:</td><td>Interaction Designer</td></tr>
<tr><td>Company:</td><td>Cognetics Corporation</td></tr>
<tr><td>Age:</td><td>mid-30's</td></tr>
<tr><td>Education:</td><td>M.S. in HCI</td></tr>
<tr><td>Specialties</td><td>Web, Intranet, Database Searching</td></tr>
</table>

Responsibilities:	Interview Users Define Requirements Produce Visual Designs Produce Specifications Coordinate Usability Testing Produce UI Style Guide

Summary:

After initially graduating with a Computer Science degree, Linda spent several years as a web administrator, database designer, and programmer two software companies. She then returned to school to complete a Human-Computer Interaction degree and joined Cognetics upon completion of her degree. Linda is an experienced web-surfer and is familiar with programming principles, though she no longer actively codes.

Linda works with clients to clearly establish a product's vision. With the vision in place, she works with users as appropriate to analyze their needs and requirements. She then uses that data to produce a draft of a user-interface and manages an iterative design process, combining expert review with usability testing as needed. She starts the design process in Visio, but she prefers to construct low-fidelity HTML prototypes as soon as possible for both review and testing. Once the design is stable, Linda typically delivers annotated specs for the full interface and the user interface style guide used to construct the prototype. She varies these deliverables for the specific project.

Goals:

- Produce an "a-ha!" interface that seems both innovative and self-evident at the same time. Once presented, it is the obvious way to approach the workflow.
- Serve as a user advocate, helping clients to align business needs and user needs.
- Perform iterative design and testing within the parameters of schedule and budget.
- Work with clients until they are confident in the user interface produced for a product.

Constraints:

- Linda is one of the few women who are red-green colorblind.
- Access to users for user analysis is not always feasible, so Linda must sometimes gather user data in more creative ways (tech support logs, surveys, remote interviews, etc.).

Figure 10: Example of user persona

More examples of personas are given in Appendix I.

4.2.2 Rationale and Claims

Persona as a tool for design is based on two justifications brought by Cooper (1999): "design for just one person" and "the elastic user".

Design for just one person

According to Cooper, to please a broad audience of users, the logic of designing a broad array of functionalities in a product is wrong. This strategy inevitably end up creating awkward and complex-to-use products that, trying to please everyone, finally please no one in a significant manner.

Cooper (1999) gives two examples of the successful strategy of products designed for "just one person": the roll-aboard suitcases and the 3M post-it notes. The roll-aboard suitcase with built-in wheels and retractable handle was first designed for airplane crew members that needed an ergonomic luggage that was easy to carry and to store in overhead bins. This successful design was then adopted and used with satisfaction by a larger audience: the business travelers. Later on, the product has been adapted to please

other target market segments and users like, long-stay travelers' oversized luggage with built-in wheels and kids' small roll-aboard toy suitcase.

As for the example of the sticky notes, Cooper (1999) relates how Art Fry, a 3M adhesive engineer, invented the Post-it® notes. Singing in a church choir, Mr. Fry needed a better solution than falling paper bookmarks to put in his hymn book. He wanted something else that sticky tape that could damage the church's book. He recalled about an adhesive that failed previous 3M stickiness tests. Putting that adhesive on yellow paper squares, he designed the Post-it® notes for his own use. This successful design has been extended and adapted to other target markets and uses afterwards.

The elastic user

Cooper (1999) also claims that even though designing for the user is the goal of any interaction designer, the term "user" is often problematic and in need of precision. To illustrate this, he coined the term of the "elastic user". Depending on the situation and on the design perspective, designers tend to "stretch and adapt" the user's needs to accommodate the design problem. Marketers, envisioning beginner and naïve users, define requirements for "user-friendly" interfaces while developers and engineers would conceive sophisticated wizards with technical steps for computer-literate expert users. Between the two, the real users are overlooked. When present in a product development team, interaction designers are caught in the middle and are talked about the users in a non-operational fashion. Even if all are trying to take the user's perspective, they fail. Cooper (1999) thinks that the users' models are so imprecise that software designers have no choice than to solve design problems in a "natural" self-referential and self-interested way. *The user* becomes *elastic* while real users are not.

4.2.3 Attributes and Characteristics

The fundamental attribute of a persona is that the persona is an *imaginary* hypothetical person. Cooper (1999) puts forward that imaginary people are better for design than real people because the latter's idiosyncrasies and behavioral anomalies can be detrimental to pervasive interaction design.

A persona is imaginary but *specific* and *believable*; it has a name, picture, age, gender family, job, lifestyle, hobbies and interests; it plays roles in his community; does sports or not; takes public transit or drives a car of a specific brand and make. A persona is a stereotype, even sometimes at the detriment of diversity and political-correctness. The goal here is to make the persona believable and humanistic enough so designers and developers get a strong sense of the user through the persona, prohibiting them to fill-in the "from the user's perspective" blanks with self-referential extrapolations.

A persona is imaginary but *precise* and *representative*; it has motives, aspirations, fears, concerns, skills and activities that are representative of a users' population without "edge-case" particularities. A computer graphics designer persona would not be someone who only uses the keyboard; a pilot persona would not be a bush pilot that flies alone most of the time or a student that only does short day flights. But that doesn't mean that a persona is an "average" user. No persona has 2.3 kids and follows "between 0 to 2" television shows per week. A persona needs to be solid and concrete, based into reality, so that it does not allow elasticity and cannot be stretched and twisted in the hands of the designers and developers looking for the "user's perspective".

A final attribute of personas is that a persona is not a buyer but a *user personification*. Cooper (1999) emphasizes that, though marketers are familiar with the concepts of

market segments underlying personas, the latter are "end-users", not consumers. As an example, he exposes the mistake made by lots of software development firms when they design IT systems for, and sometimes with, their clients: the purchasing IT managers. In those cases, it is not rare to see those IT managers go back to their vendors with unfulfilled user requirements after receiving many end-users complaints following the deployment of new interactive systems.

4.2.4 Benefits and Risks

According to Cooper (1999), persona's imaginary, though realistic, representation of users' goals and skills is beneficial to designers, programmers and managers at ending "feature debates". Personas eliminate the construct of "the user" and guides functional specifications. Cooper (1999) reports this example.

Programmer: "Would Rosemary want to print this out?"

Designer: "No. Although Jacob will want some printed reports on a quarterly basis."

Programmer: "Well, if they are so rarely needed, we should save ourselves time and effort by not writing a fancy, proprietary report-writing feature, but instead license a commercially available tool."

Manager: "And that shaves two weeks off the shipping schedule".

Calde & al. (2002) adds: "personas remind everyone they are building a product to solve people's goals, not to showcase technology."

Relating Microsoft's experiences with personas, Grudin & Pruitt (2001) gives those benefits to the use of personas.

- Personas bring users' research to life promoting a user/work focus and awareness not only for interaction design but broader use (product strategy, vision documents, feature specs, bug bashes) in an end-to end approach.
- Personas promote cross-team synergy and communication creating a shared vocabulary and conduit to talk about users and work settings.
- Personas enhance attention, memory and organization of detailed user data through the effectiveness of narrative.
- Personas encourage creative and explicit design decision-making processes by allowing the mind to project new settings or situations and to extrapolate criteria.
- Personas drive the construction of scenarios from a real people/real life perspective.

On the other hand, there are some pitfalls to avoid when using personas as an interaction design tool. The major risk, reported by both Grudin & Pruitt (2001) and Mikkelson (2000), is the challenge to get the right persona or set of personas without stereotyping. Persona creation implies choices and biases that could over generalize or exclude users. As said previously, Cooper would reply to this argument saying that it is better to design for *any* one person instead of ineffectively designing for the "masses". Grudin & Pruitt (2001) also talks about the risks of reusing and overusing personas, meaning stretching them so much that they loose precise representativity. He finally adds that some may try to substitute personas to all other user-centered design principles of ongoing user research and usability evaluation.

4.3 Persona Lifecycle

Adlin, Jamesen & Pruitt (2002), are proposing a framework to capture the process of creating and using personas for interaction design. This persona lifecycle consists of 5 phase: family planning, birth, maturation, adulthood and retirement & lifetime achievement. Those phases are described below.

4.3.1 Family Planning

Family planning is the user research phase. In this phase, the complexity, size and costs of the persona project is evaluated and the scope of the personas effort is defined. Also, target market segments are identified, existing buying and behavioral customer/user data is reviewed. From that preliminary analysis, it should be possible to identify hi-level user roles. Those roles are usually job or task-oriented (nurse, IT manager, small business owner, etc.) and depend on the product to be designed. With further data, the user roles will evolve or may even be discarded, but, at this early stage, they are useful in orienting and planning the user research strategy. This plan includes answers to questions like the following ones.

- Which rapid ethnography research method will be used, site visits, indirect observation through log books, semi-structured interviews, co-discovery of tasks, or other data gathering methods?
- Is anti-persona data should be gathered also?
- How many subjects will be visited, interviewed?
- Is there diversity (gender, age, disabilities, etc.), cultural (ethnic, languages, etc.), political, security, confidentiality, accessibility issues?
- Who (real users or proxies) and how are those subjects going to be recruited?

- When and where the research will take place? Who is going to conduct the observations and interviews? How data analysis is going to be done? By whom?
- And mostly, what data must be looked for: what should be the focus, who should be key information to be looked for? Which questions and how?

4.3.2 Birth

Birth is the persona creation phase where user research data is turned into "imaginary people". More than one technique can be used to do that conversion from user data to persona. The choice of the method could depend on expertise and experience of the persona team members or the time and resources available. Below is exposed a "conversion" technique adapted from goal-directed design® and contextual design methodologies.

Step 1: Variables identification

From the family planning research data and using affinity diagramming, extract the recurring socio-demographic and behavioral information (with a focus on goals and usage) in terms of variables by looking for "the differences that make a difference"(Goodwin, 2001).

Step 2: Patterns identification

With the variables identified in step one, produce scales for major and meaningful variables. Ranges with logical opposite variables can also be useful. Position the research subjects on those scales or ranges. Patterns should emerge significantly from where research subjects cluster. The next figure illustrates that method.

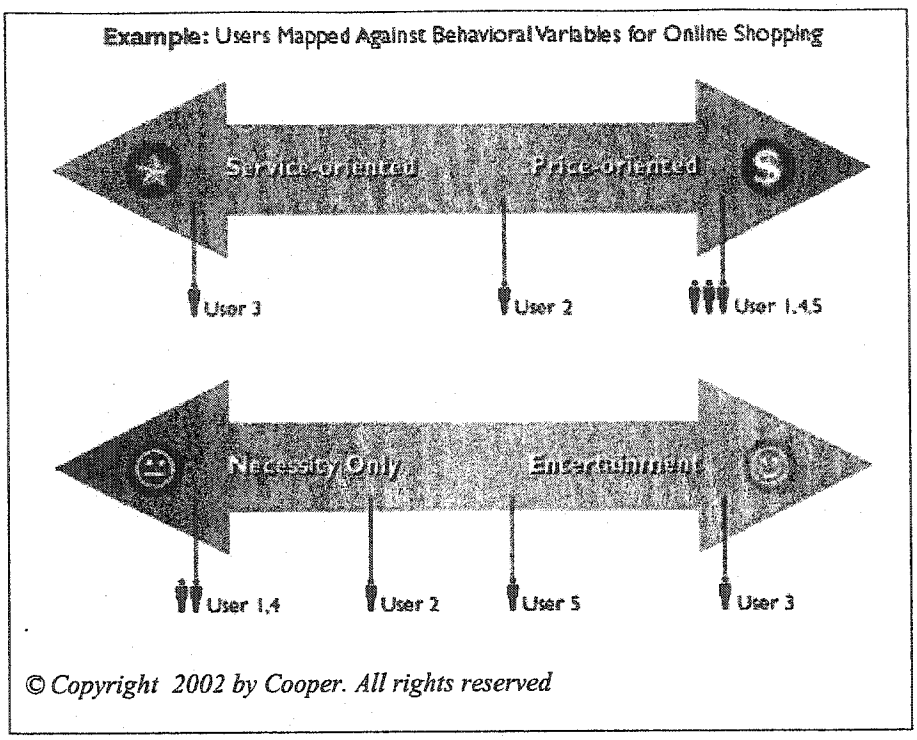


Figure 11: Example of behavioural variables pattern analysis

Step 3: Personas creation

Use the emergent patterns to build the persona characters. A main persona and secondary one (and an anti-persona, if that data was gathered in the family planning phase) should materialize from data analysis. Create the personas in a narrative form: name them, illustrate them with a picture, describe their background and family, depict their goals and activities, portray their aspirations and concerns and give the personas quotes and taglines. Be specific and precise. Strive for believability. Shield the personas against "elasticity".

The next figure presents the table of contents of Microsoft's persona central "foundation" document used to facilitate the creation of personas (Grudin & Pruitt, 2001).

Overview – Persona name (job title, work domain)
Get to know the persona, his business and family.

A Day in the Life
Follow the persona through a typical day.
Work Activities
Look at the persona's job description and role at work.

Household and Leisure Activities
Get information about what the persona does when he's not at work.

Goals, Fears, and Aspirations
Understand the concerns the persona has about his life, career, and business.

Computer Skills, Knowledge, and Abilities
Learn about the persona's computer experience.

Market Size and Influence
Understand the impact people like the persona have on our business.

Demographic Attributes
Read key demographic information about the persona and his family.

Technology Attributes
Get a sense of what the persona does with technology.

Technology Attitudes
Review the persona's perspective on technology, past and future.

Communicating
Learn how the persona keeps in touch with people.

International Considerations
Find out what the persona is like outside the U.S.

Quotes
Hear what the persona has to say.

Figure 12: Microsoft's persona foundation document

4.3.3 Maturation

Maturation is the validation and diffusion phase, where the newly-born personas are communicated to the "world". Personas should first be communicated to the product marketing people who are more accustomed to that type of artifacts. Also, the marketing can validate the precision and believability of the personas. Adjustments to the personas to fit corporate strategic goals may be needed. After, and depending of the context of the project, there are many ways to present the personas to the design, development and test teams. Some examples and ideas from participants of 2002 UPA personas workshop were: personas' showroom and greeting meeting, personas' intranet posters placed in hi-level traffic area, t-shirts, mugs, etc. Some project even gave the personas email addresses so users of that interaction design tool could ask questions to them. When the persona tool is introduced for a first-time into an organization, an emphasis on giving the design teams information on the persona concept and its use should be envisaged. Training sessions can be implemented to promote acceptance of that tool.

4.3.4 Adulthood

Adulthood is the phase where personas are "grown-ups" and have a job to do. Maybe they need to be "fixed" to keep up with the evolution of design/development project's shifting requirements and priorities. These potential changes are necessary and need to be communicated. Otherwise the usefulness and representativity of the personas may disappear and they could "die on the vine".

4.3.5 Retirement and Lifetime Achievement

Retirement and lifetime achievement is the phase when personas have done their job and the design/development/testing project is over. The contribution of the personas to the software lifecycle should be captured in post-mortem documentation. Also, lessons learned and possibility of personas reuse for future product version or across products line should be reviewed and evaluated. As described in the next section, personas can be used in all the product development lifecycle and should be treated as any other software development artifact.

5 Putting Personas into Practice

Persona is a relatively new, even controversial, interaction design tool. Therefore, information and research on the integration of personas into the software practice is almost inexistent. For that reason, this section describes some potential usages of personas in the overall software lifecycle phases of analysis, design and evaluation. After, a focus on the use of personas in the design activity is taken and discussed. Finally, quality in use of personas as an interaction design tool is looked at.

5.1 Personas within the software lifecycle

5.1.1 Analysis & Requirements

Personas could be a valuable tool for a lot of activities occurring in the analysis phase of any interactive system development or maintenance project. Capturing the knowledge about users in terms of goals, personas help discover and elicit user requirements from an end-user perspective, not from clients or purchasers' point of view. Personas contribute in the system requirements activities in facilitating the definition of functional requirements. As well, personas are an important starting point when further cognitive or task analysis is needed or work reengineering has to be carried out. Evidently, personas also assist the setting of non-functional requirements and principally usability and quality in use objectives. Consequently, personas assist scope and plan quality assurance strategies. Also, personas should help identify the needs for user documentation or training. Finally and importantly, personas can effectively be called upon to prepare and conduct

marketing research strategies like competitive benchmarking or product acceptance analysis.

5.1.2 Early Design & Prototyping

Intentionally, personas are of great help and influence in the design phase. Their narrative form and "real life" representational mode promote and facilitate scenarios creation and use-cases building. Information enclosed in the personas can certainly be useful for early design where low to mid-fidelity prototypes and user interface metaphors are defined. In those stages of hi-level to low-level design, personas help the conceptualization of user interfaces, tasks and features allocation, as well as guide the right choices of design implementation.

Personas, with weighted importance, can also lead software designers and managers prioritize scenarios or features in a structured way. The following figure gives an example of that use (Grudin & Pruitt, 2001).

	Persona 1	Persona 2	Persona 3	
Weight:	50	35	15	Weighted Sum
Feature 1	0	1	2	65
Feature 2	2	1	1	150
Feature 3	-1	1	0	-15
Feature 4	1	1	1	100
Etc.	-	-	-	-

Figure 13: Features-personas weighted priority matrix

But importantly, we believe that personas can provide software designers with guidance and support for innovation, reflection-in-action and commitment to usability and quality in use. This view will be discussed in more details in section 4.2.

5.1.3 Design Quality Assessment

Personas are assisting many quality assessment activities of software designs. They can support design review meetings and usability assessment activities by giving a voice to end-users. They guide cognitive walkthroughs and discount usability evaluations by allowing evaluators to put themselves in the shoes of the personas when assessing early designs and deciding the need for further design iterations. In addition, personas are certainly precious information for usability expert when doing heuristic evaluations of design artifacts.

5.1.4 Development and Testing

Personas can support activities of software development and testing. They can aid the production of test cases and actual testers' activities for quality control and user acceptance phases. As for formal post-implementation usability and quality in use testing goes, personas are of great value for recruiting subjects, crafting scenarios and test protocols.

5.1.5 Usability Maturity of an Organization

The use of personas throughout the software development lifecycle promotes not only user-centered design but can also raise usability concerns in teams other than the design team. By giving a "real-life face" to market segments, personas help product managers in thinking about clients in terms of end-users. They may also aid developers in identifying

search algorithms, or others, that will serve "persona Jill" better. Also, personas should assist quality control workers in the production of realistic test cases and testing activities for quality assurance and user acceptance phases. Finally, when viewed in an end-to-end approach of the software lifecycle, the greatest contribution of personas is being a mean to support and promote cross-functional teams' communication about usability and quality in use.

5.2 Designing with and for Personas

Implementing methods for designing with quality targets of usability and quality in use is a recurrent challenge. HCI advocates have studied individual and organizational obstacles to this goal; see Gould & Lewis (1985) and Poltrock & Grudin (1994) for good studies. But those issues are not the focus of this thesis. Here, and to link back to our HCI perspective exposed in section 2.2, personas are studied as a tool promoting usability and quality in use by bringing operational knowledge about users in the conceptual, constitutive and consolidatory steps of the design activity.

5.2.1 Conceptual Step

Remembering that this step is the where *creativity* take place, we think that personas could be an extremely powerful tool for designers to create *inventive* and *innovative* solutions of quality. Personas are grounded in true user data without being complex, limitative and "boring" like task analyses. Personas are playful and communicative without being "flaky Hollywood actors". Therefore, we think that this tool should encourage designers to generate fruitful and sound ideas based on "real-life" users' needs

and values, in the same way archetypes of real-life inspire striking artists and harmonious relationships with real-life motivate great architects.

5.2.2 Constitutive Step

The constitutive step is where design decisions and trade-offs are occurring. We think that, by demoting self-referential mode, personas allow for and enhance *reflection-in-action*. Schön (1982) coined the term "reflection-in-action" to describe the internal "backtalk" a practitioner does in the midst of his practice activity when a question or problem arises. He calls those questions or problems "surprises" and qualifies the backtalk process of "discussions with materials". He gives the analogy of jazz improvisation where a musician is reflecting-in-action when he is surprised by and answers to other band members' music. In the same way, when modeling and planning designs, a software practitioner can play the "persona game". He can show his stuff to the persona, listen to the persona's level of surprise and they can discuss on the materials to make design decisions and negotiate trade-offs. By enhancing designer's reflection-in-action with an operational user's perspective, we believe that personas should promote usability and quality in use in an efficient manner that fits better than user involvement in the introspective activity of design.

We also note the similarities and potential beneficial exchanges between personas' use and the claim analysis process of scenario-based design described in section 2.2.4.5. We acknowledge that the persona is a less powerful tool than claim analysis which is based on HCI theories and knowledge and produces the design rationale. But the limitations of the persona tool should not prevent its use all together. Personas must be viewed as a tool amongst others in the designer's toolkit. And, in conjunction with interaction design

patterns per example, personas may be sufficient to address a broad array of design situations. As a craft, design requires the right tool for the right material, at the right time.

5.2.3 Consolidatory Step

In the consolidatory step, the designer judges the elegance of the design and evaluates its appropriateness for use. As suggested by Grudin & Pruitt (2001), we think that personas, being a vehicle promoting personal and emotional engagement towards users, could help designers make that decision in a more socially responsible way.

A persona, with its real-life representations and details, necessarily includes cultural indicators. Therefore, when designers use personas, they internally build "politicized" mental models of the users. In that constructivist way, we believe that it is more likely for designers to show social responsibility in their design activity and the resulting artifacts.

Consequently, the use of personas as a tool for interaction design would encourage flawless and authentic concern about users by replacing self-reference with empathy when elegance and appropriateness judgments are done. Personas could even promote ethical actions.

5.3 The Quality in Use of personas

A good question to ask is "does the persona tool possess the qualities it strives for"? Do personas encompass usability and quality in use? Are personas usable, effective, productive and satisfactory to designers? First, we investigated if personas' users, e.g. software designers, find the tool satisfying and usable. Then, we examined the quality in use of personas with the help of an ethnographic study looking at how that tool supported design tasks in a software development project context.

5.3.1 Usability and User Satisfaction of Personas

In a software engineering course focusing on user-centered design given at Concordia University in the spring of 2003, 54 undergraduate students were given a class project including the creation and use of personas for design. After the completion of the design projects, students were administered a survey about their perceptions of the usability and their satisfaction of the persona tool. The questionnaire used for this informal evaluation consists of an adaptation of the System Usability Scale (SUS) devised by Brooke (1996). The questionnaire, which uses a ten-item scale giving a global view of subjective assessments of usability and user satisfaction, is included in appendix II. The output of the questionnaire is a score from 0 to 100 giving individual user's rating of the perceived tool usability and their satisfaction. The following figure gives the distribution of the scores collected with that questionnaire.

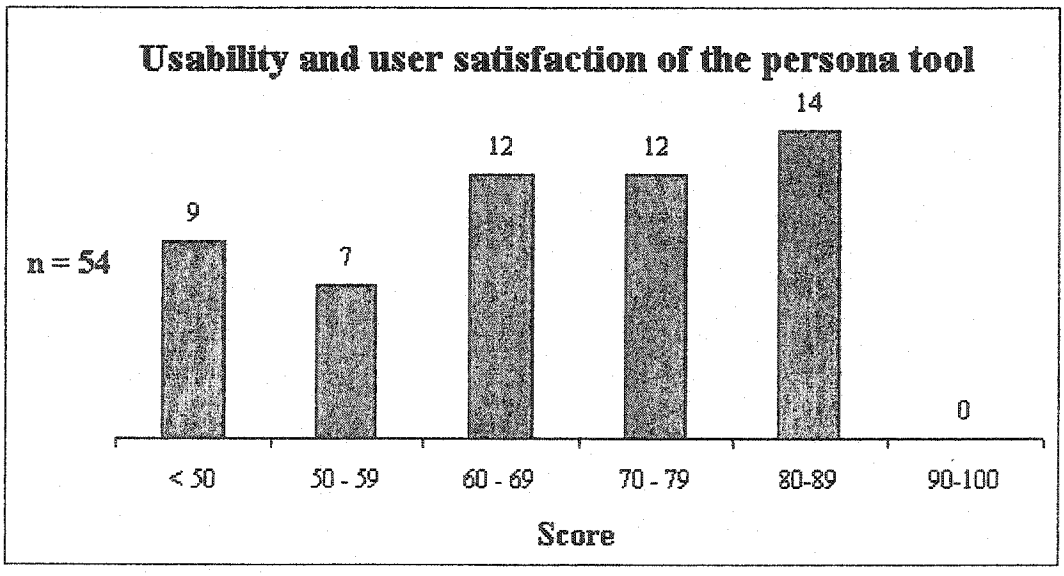


Figure 14: Scores' distribution for the usability and user satisfaction of personas

Overall analysis of that survey tells that 26% of the respondents strongly agree that the personas are usable and are satisfying with the tool (scores equal or above 80 points). Around 44% of the respondents agree that personas are usable and satisfying (score between 60 and 79 points). While about 30% of them disagree or are neutral about the usability of personas and are not satisfied with them (score of 59 points and less). Since the questionnaire uses Likert ordinal scales, a preferred central trend measure for the scores distribution would be the median. The median, which value is 67.5, indicates that most respondents agree that the tool is usable and are satisfied with personas.

This preliminary and qualitative assessment of the perception of usability and user satisfaction of the persona tool for the design of interactive systems is encouraging. But evidently, more through analysis of the understandability, learnability, operability and attractiveness of that design tool is needed. For example, an analysis using those criteria could be performed on personas' creation/validation processes and documentation/ communication support tools in order to evaluate its overall usability.

5.3.2 Quality in Use of Personas

In a recent field study conducted at Swedish IT company "trying out" personas in a web portal project, Blomquist & Arvola (2002) identified major issues with the use of that tool. They report that the interaction design team appreciated personas for design and successfully used them in design critique meetings. But they also account that, although all the project team members found that personas were a good tool to "think things through and focus the team members' effort", persona suffered of a lack of acceptance from developers and project managers who did not see their relevance. Therefore, the design team was unsatisfied with the tool for project meetings. For us, this study suggests

that the persona tool would not possess quality in use attributes of effectiveness, productivity and satisfaction. Personas would be deficient in terms of quality in use when examined in a software development context of use. This tells us that further and more formal investigation of the quality in use of personas for the design of interactive systems in software development work context is needed. For example, comparative experiment of the usability and the quality in use of two interactive systems, one designed with personas and the other one without but in the same work context, could be conducted to assess the effectiveness and productivity of the tool.

6 Conclusion

In this thesis, we first reviewed, via an analysis of the literature, the concept of personas as a way for involving the end-users in software development with a focus on the early stages: the design process. We highlighted the benefits and challenges of user involvement, summarized the interaction design methodologies taking the users into account and presented the proposed approaches for understanding/modeling those users. Then, we looked at personas as an information and communication tool for designers to "more easily" involve users in the design process. That way, we addressed the questions of usability, effectiveness and productivity of personas in the design process. We also investigated the satisfaction of software practitioners using personas in their work activities.

We found out that personas clearly bridge the gap between marketing research and user research for designers' discovery and analysis of user requirements. Personas help individual and teams of designers "discuss with their materials" and judge the appropriateness of their design from the user's perspective. And, as suggested by our review of the potential use of personas in the design activity, and also by our informal usability and user satisfaction survey, we believe that personas are useful and usable for interaction designers.

Nevertheless, we acknowledged, like Blomquist & Arvola (2002), that personas have a credibility problem when entering the world of software engineers. Also, from a practical point of view, like many other interaction design techniques, personas' use will encounter resistance brought by their obligation for ethnographic methods for user data gathering. And therefore, personas are not productive in the software development work activities.

Consequently, more experiments and measurements are needed to evaluate if the persona tool is effective, productive, safe and satisfactory for software interaction design practitioners in the specific context of use for software development. And, if the persona tool is declared to be of quality in use, more articles, case studies and training will be needed to promote the use of that interaction design tool.

We also note, like Sutcliffe (2000), that "a fundamental mission of HCI is bringing psychology and other sciences such as sociology to bear upon design. If HCI fails to employ knowledge about the very people it is designing for then it is left with technology and creative inspiration." Hence, we can foresee objections to the use of personas since they are deficient in encapsulating HCI theories and knowledge and are of little help in systematically mapping requirements into usable interface designs. Two avenues are possible to address such views. One, personas' use should be investigated in conjunction with other interaction design tool, like usage-centered design, scenario-based design and claim analysis. Two, personas could be extended to include HCI theories and knowledge, like patterns for examples. In any case, personas are popular right now in the interaction design community and it will be interesting to see if that novelty will "find its spot" in software design and development practices as well as in the HCI research agenda.

"The real question before us lies here:
do these instruments further life
and enhance its values, or not?"
Lewis Mumford in:
Technics and Civilization, 1934.

7 Bibliography

Adler, P., Winograd, T. (1992). The usability challenge. In Adler, P., Winograd, T. (eds.) *Usability: Turning Technology into Tools*, pp.3-14. New York: Oxford University Press.

Adlin, T., Jamesen, H., Pruitt, J. (2002). Usability Professionals' Association 2002 Conference. Notes from the workshop "Creating and using personas".

Archer, L.B. (1984). Systematic method for designers. In Cross, N. *Developments in Design Methodology*, pp57-82. New York: John Wiley & Sons.

Blomquist, A., Arvola, M. (2002) Personas in Action: Ethnography in an Interaction Design Team, NordiCHI 2002 Proceedings, pp. 197-200. ACM Press.

Brooke, J. (1996). SUS: A Quick and Dirty Usability Scale. In P. W. Jordan, B. Thomas, B. A. Weerdmeester and I. L. McClelland (eds). Usability Evaluation in Industry, 189-194. London: Taylor & Francis.

Calde S., Goodwin K., Reiman, R. SHS Orcas: The first integrated information system for long-term healthcare facility management, Experience Design Case Study Archive, American Institute of Graphics Arts. (2002)

Carroll, J.M. (2000) *Making use: Scenario-based design of human-computer interactions.* Cambridge, MA: The MIT Press.

Constantine, L.L., Lockwood L.A.D. (1999). *Software for Use: A Practical Guide to the Models and Methods of Usage-Centered Design*. Reading, MA: Addison-Wesley.

Constantine, L.L., Lockwood L.A.D. (2001). Usage-Centered Engineering for Web Applications. <http://www.foruse.com/articles/webapplications.htm>

Constantine L.L. (2000) The Usability Challenge: Can UML and the Unified Process meet it?, TOOLS Pacific, Sydney. <http://www.foruse.com/presentations/tools.htm>

Constantine L.L. (2002) forUse: The electronic newsletter for Usage-Centered Design, #26, October. <http://www.foruse.com/newsletter/foruse26.htm#3>

Cooper, A. (1999), *The Inmates are Running the Asylum.* Indianapolis, IN: Macmillian Computer Publishing.

Diaper, D. (1987) Designing Systems for People: Beyond User Centred Design. in *Software Engineering. Proceedings of the Share European Association (SEAS) Anniversary Meeting*, pp. 283-302.

Dreyfuss, H.L. (1986). Five Steps from Novice to Expert. In Dreyfuss, H.L., Dreyfuss, S.E., *Mind over Machine*, Free Press.

ISO (1994) *International Standard ISO/IEC8402. Quality management and quality assurance – Vocabulary*, International Organization for Standardization, International Electrotechnical Commission, Geneva.

ISO (1991). *International Standard ISO/IEC 9126. Information technology -- Software product evaluation -- Quality characteristics and guidelines for their use*, International Organization for Standardization, International Electrotechnical Commission, Geneva.

Licklider, J.C.R. (1960). Man-Computer Symbiosis. *IRE Transactions on Human Factors and Electronics*. Volume HFE-1, pp. 4-11.

Engelbart D. (1962). A Conceptual Framework for Augmenting Human Intellect. *SRI Project No. 3578* <http://sloan.stanford.edu/mousesite/EngelbartPapers/B5_F18_ConceptFrameworkInd.html>

Ehn, P. (1988). *Work-Oriented Design of Computer Artifacts.* Stockholm: Arbetslivscentrum.

Gagné, R.M., Driscoll M.P. (1988). *Essentials of learning for instruction.* Englewood, NJ: Prentice-Hall.

Gilmore, D. (2002). Understanding and Overcoming Resistance to Ethnographic Design Research. *Interactions, May-June*, pp.29-35. ACM Press.

Goodwin, K. (2001). Getting from Research to Personas: Harnessing the Power of Data. <http://www.cooper.com/content/insights/newsletters/2002_11/getting_from_research_to_personas.asp >

Gould, J.D. (1988) How to Design Usable Systems. In: Helander, M. (ed.) *Handbook of Human-Computer Interaction*, pp. 757-789. Amsterdam: Elsevier.

Gould, J.D., Lewis, C. (1985). Designing for Usability: Key Principles and What Designers Think, *Communications of the ACM*, March 1985, Volume 28, Number 3. ACM Press.

Grudin, J., Pruitt, J. (2001). Personas, Participatory Design and Product Development: An Infrastructure for Engagement. *Proc. Participatory Design Conference'01.*

Hansen, W.J. (1971). User engineering for interactive systems. *AFIPS Conference Proceedings 39*, AFIPS Press, 1971, pp. 523-532.

Hartfield, B., Winograd, T. (1996). IDEO. In Winograd, T., *Bringing Design to Software*. New York, NY: Addison-Wesley.

Holtzblatt, K., Beyer, H. (1998). *Contextual Design: defining customer-centered systems.* San Diego, CA: Academic Press.

Holtzblatt, K. (2001). Driving Real Innovation with Customer Data. Tutorial notes. User Interface 2001 West conference.

Holtzblatt, K., Jones, S. (1995). Conducting and Analyzing a Contextual Interview. In Baecker, R.M., Grudin, J., Buxton, W.A.S., Greenberg, S. (eds.) *Readings in Human-Computer Interaction: Toward the Year 2000.* San Francisco, CA: Morgan Kaufman.

Kujala, S. (2003). User involvement: a review of the benefits and challenges. *Behaviour & Information Technology*, Volume 22, Number 1, pp. 1-16. Taylor & Francis Ltd.

Löwgren, J. (1995). Applying Design Methodology to Software Development. *DIS Proceedings.* Ann Arbor, MI: ACM Press.

Lundequist, J., Ullmark, P. (1993). New concepts for the design of industrial buildings. In Tornqvist, A., Ullmark, P. (eds.) *Appropriate architecture: Workplace design in post-industrial society.* IACTH, Chalmers University of Technology, Sweden.

Mark, M., Pearson, C.S. (2002). *The Hero and the Outlaw: Building Extraordinary Brands through the Power of Archetypes*. New York, NY: McGraw-Hill.

Mayhew, D. (1992). *Principles and Guidelines in Software User Interface Design.* Englewood Cliffs, NJ: Prentice-Hall.

McPhee, K. (1996) Design Theory and Software Design. *Technical Report TR 96-26.*, Dept. Computer Science, University of Alberta, Edmonton, AL, Canada.

Millen, D.R. (2000). Rapid Ethnography: Time Deepening Strategies for HCI Field Research. *DIS'00 Proceedings*, Brooklyn, NY: ACM Press.

Miller, G. A. (1956). The magical number seven, plus or minus two: Some limits of our capacity for processing information. *Psychological Review,* 63, 81-97.

Minsky M. (1975). A framework for representing knowledge. In Winston P. (ed.), *The Psychology of Computer Vision*, New York, NY: McGraw-Hill, pp. 211-277.

Moore, G.A. (1991). *Crossing the Chasm: marketing and selling high-tech products to mainstream customers.* New York, NY: Harper-Collins.

Moran, T. P., Carroll, J.M. (eds.) (1996). *Design rationale: Concepts, methods and techniques*. Hillsdale, NJ: Erlbaum.

Muller, M.J., Wildman, D.M., White, E.A. (1993). Taxonomy of PD Practices: A Brief Practitioner's Guide. *Communications of the ACM,* Vol.36, No.4. ACM Press.

Nielsen, J. (1993). *Usability Engineering.* San Francisco, CA: Morgan Kaufmann.

Norman, D.A., Draper, S.W. (Eds.) (1986). *User centered system design: New perspectives on human-computer interaction.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Norman, D.A. (1988). *The Design of Everyday Things.* New York, NY: Bantam-Doubleday.

Parush, A. (2001). Usability design and testing. *ACM Interactions.* Volume 8, number 5. September/October.

Poltrock, S.E., Grudin, J. (1994) Organizational Obstacles to Interface Design and Development: Two Participant Observer Studies. *TOCHI* 1(1): 52-80. ACM Press.

Potosnak, K.M. (1986) Classifying Users: A Hard Look at Some Controversial Issues. CHI 1986 Proceedings, pp. 84-88. ACM Press.

Schön, D.A. (1982). *The Reflective Practitioner: How professionals think in action.* New York, NY: Basic Books.

Sutcliffe, A. (2000). On Effective Use and Reuse of HCI Knowledge. *Transactions on Computer-Human Interaction.* Volume 7, Number 2, June, pp 197-221. ACM Press.

The Standish Group (1995), *Chaos.* Report T23E-T10E.
<http://www.scs.carleton.ca/~beau/PM/Standish-Report.html>

Tullis T.S. (1993) Is user interface design just common sense? *Proceedings of the Fifth International Conference on Human-Computer Interaction*, Volume 2, p. 9-14, Elsevier Science.

Verplank, B. (1993) Observation and invention: Use of scenarios in interaction design" INTERCHI tutorial, Amsterdam.

Whiteside J. & al. (1988). Usability engineering : Our experience and evolution. In . In: Helander, M. (ed.) *Handbook of Human-Computer Interaction*, pp. 757-789. Amsterdam: Elsevier.

Wickens, C.D.(1984). *Engineering psychology and human performance.* New York, NY: Harper Collins.

Wilson, J.R., Rutherford, A. (1989). Mental Models: Theory and Application in Human Factors. *Human Factors,* Volume 31, Number 6, pp. 617-634.

Winograd, T., (1996). *Bringing Design to Software.* New York, NY: Addison-Wesley.

Appendix A: More Personas examples

iRippage.com user personas

Test Respondents

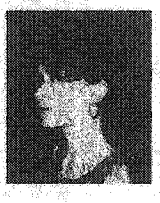
Seth:
male, 13,
8th-grade gamer

- First-time skater
- Internet and gaming addict with a PC and 56K modem; carries on friendships with kids around the world via instant messages and message boards
- Small town; no skate shops or parks nearby; wants to learn tricks, trade tips with other kids his age; worships Eric Koston
- Complains about the lack of regular updates on his favorite tips-and-tricks sites

Garrett:
male, 26,
bicycle messenger

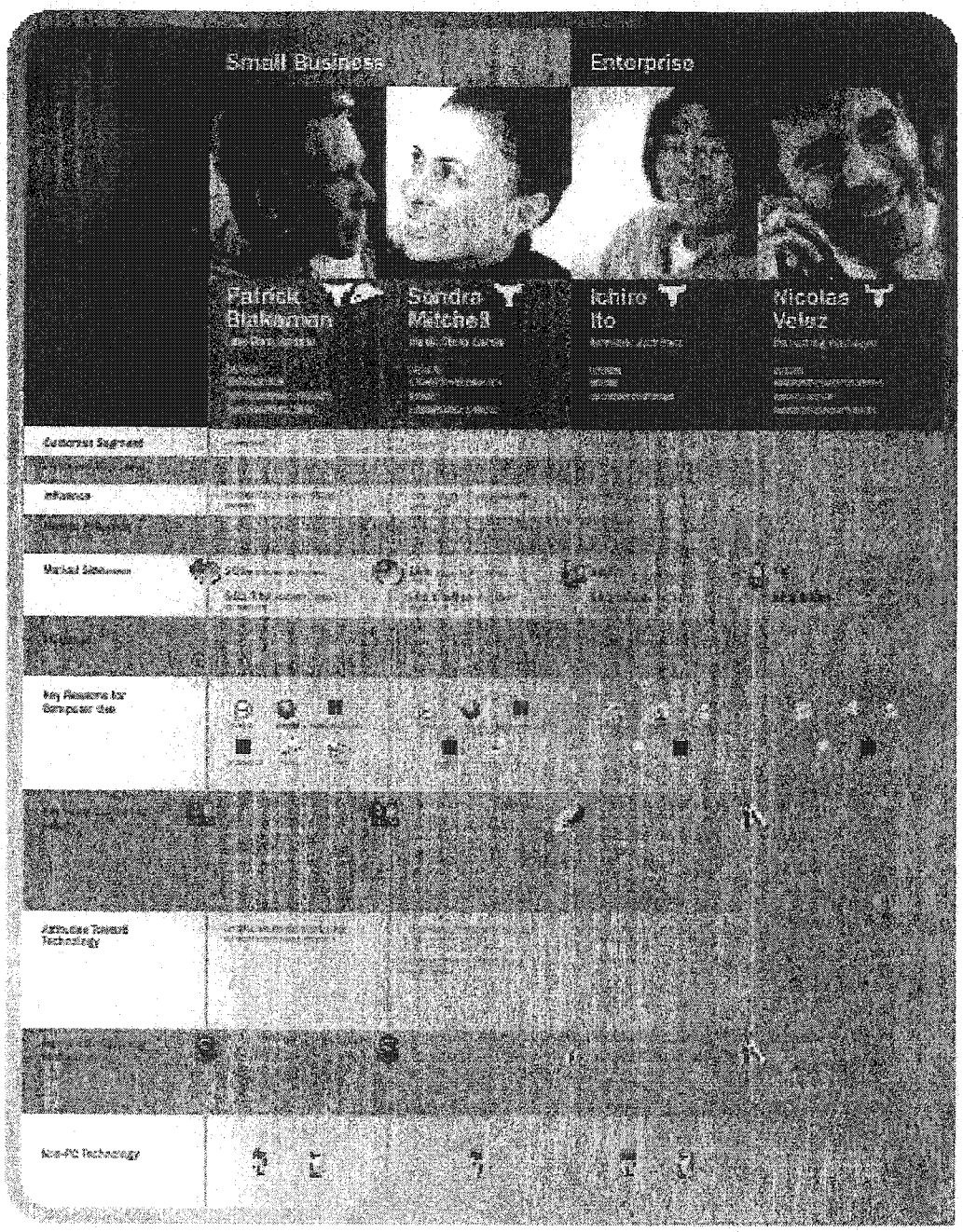
- Skateboarder since middle school
- New to the Web; surfs once or twice a week on either his roommate's ancient PC with a 28K modem, or occasionally on WebTV at his girlfriend's house
- Lives in a medium-sized city; building a skate park with some friends to encourage younger kids to pick up the sport; would like to browse plans and trade ideas with other experienced skaters; respects Mike Vallely
- Complains about slow-loading pages, feeling "buried" in Web sites; often unsure about how he got where he is or how to get back to where he was

Jessie:
female, 17,
high-school senior



- Four years of skateboarding experience
- Hard-core Mac enthusiast; got a G4 and a DSL line for Christmas; runs her own girl skater zine; checks her email and favorite message boards daily
- Urban setting; access to several skate parks; an aspiring photographer who shoots her skater friends in action; emulates Elissa Steamer
- Takes issue with the rarity of skate sites featuring female skaters or offering accessories and apparel for girl skaters

From: http://builder.cnet.com/webbuilding/pages/Graphics/Conceptualize/ss02.html



From: Grudin, J., Pruitt, J. (2001). Personas, Participatory Design and Product Development: An Infrastructure for Engagement. *Proc. Participatory Design Conference'01.*

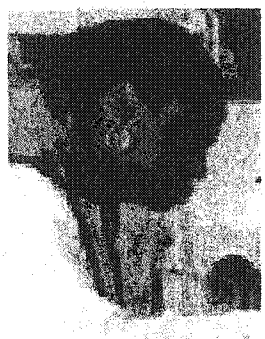


Figure 1: The team created detailed descriptions of each persona, including a stock photo and a biography. The text to the right represents a summary of Rhonda's persona description (each description was about a full page in the actual deliverable).

Persona Example: Rhonda Wilson, RN
Nurse Unit Coordinator

Rhonda is a 36-year-old RN who has worked at several skilled nursing facilities. She started out in acute care but moved to long-term care so she could have more autonomy. Rhonda was promoted to Unit Coordinator four years ago because she is very competent and generally well organized.

Rhonda is entirely overwhelmed and is drowning in paper, even more so than the average nurse. She often misses eating dinner with her boyfriend because she has to work late, filling out forms and reports.

Rhonda's goals are to:

- **Spend time on patient care and staff supervision, not paperwork.**
- **Be proactive.** Rhonda needs to understand trends in order to solve problems before they happen, instead of just reacting to crises.
- **Know that things are being done right.** Rhonda supervises the unit because she's good at what she does. If nurses aren't following procedure or documenting things, she wants to know right away.

From: Calde S., Goodwin K., Reiman, R. SHS Orcas: The first integrated information system for long-term healthcare facility management, Experience Design Case Study Archive, American Institute of Graphics Arts. (2002)

Daisy Bead Company Persona - Sara Locke (Primary)

Computer Savvy
***

Web Savvy
***

Online Shopping Savvy
**

Background

Sara is a 25 year old, single woman living in the University District of Seattle Washington. She of average height and build, fairly athletic and has brown hair and green eyes. She is unmarried but has a boyfriend of 2 1/2 hears and they are starting to think about tying the knot. She lives with her roommate, Katie, and has 2 cats, Bob and Rufus. She working in the marketing department of a high volume airplane parts manufacturer in Everett, 20 some miles north of Seattle. Her hobbies include dancing, mountain biking and snowboarding. She loves going to the movies. She also loves to shop.

Web usage

Her Internet usage is limited mainly to work related functions and E-mail. She has a hotmail account for her personal correspondence and uses it almost every day at lunch and sometimes on weekends or eves. He has shopped online but only a few times, she often finds the process confusing. Because of how she needs to use the Web at work she has a pretty good grasp of how most things work, she considers herself fairly savvy as she uses a mailing list program and spends a bit of time on marketing type sites. She doesn't enjoy spending a whole lot of time on the computer, so she prefers to get on and off quickly, and likes those sites that help her do that. Sara likes it simple and straightforward.

Online shopping

As far as shopping sites goes, she likes Amazon, and Nordstroms, but doesn't have a whole lot of experience with others. She uses these mainly for gifts and would prefer to hit the mall or downtown before resorting to shopping online. Often times she may look for some information about a particular product online and then go pick it up (or not) at the store. Most of her shopping experiences have been limited to research and information gathering. When she has made a purchase she finds that sometime the process is a bit overwhelming.

From: http://www.7nights.com/asterisk/archives/example_personas.php

Daisy Bead Company Persona - Jane King (Secondary)

Computer Savvy
***

Web Savvy
***

Online Shopping Savvy
****

Background

Jane is a 33 year old entrepreneur, she runs a small boutique in downtown Seattle selling jewelry, clothing and shoes. She is married, has no children (yet) and lives with her husband, Brian, a mortgage banker in a Bellevue condo. She is tall, thin and has dark hair and light eyes. She spends most of her time at work or working on the condo, which they bought and moved into earlier this year. When she has free time she loves to read, travel and catch some live jazz.

Web usage

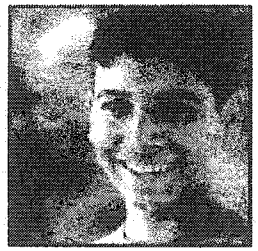
Jane uses the web quite a bit. Her store has a website so she does her best to make sure that it's up to snuff. She has an old college friend who has his own freelance Web design business do the maintenance for her. The site has a shopping cart application, that while it's no Amazon, works fairly nicely and thus far the site has meant quite a bit to her business. She also spends lots of time checking out other similar sites to hers and keeping up on news and that sort of thing. She gets quite a bit of email, too much spam however, and wishes that the Web in general were a safer, less intrusive place. But she can't knock what it's done for her business. She doesn't get too many sales, but lots of referral calls and interest. She sees her site as a marketing and advertising vehicle more than a store front at this point.

Online shopping

As she has a vested interest in online shopping she spends as much time (and money) as she can shopping on the internet and has noticed marked improvement in the process over the years. She still thinks there is lots of room for improvement, but likes where it's going. Her biggest frustration is returning or exchanging of items. Personally she likes the actual experience of going down to the store and touching the goods, but can't deny the convenience of shopping online. She views it as the future of her own business, but doubts it'll ever replace her shop. In other industries, but in the clothing and jewelry business there is a need to have contact with the items.

From: http://www.7nights.com/asterisk/archives/example_personas.php

personas



Marcus Verdeja
15 years old
Drinks soda, no preference.

Marcus's story: Marcus lives in Austin, Texas, with his parents and three siblings. He's 15, a freshman in high school. He works part-time at McDonald's after school, and most of the money goes to clothes and school supplies. He's saving up for a Playstation 2.

Marcus's relationship with Sprite: Marcus loves soda, and never will drink water or milk if he can get his hands on a soda-pop. He drinks a number of different brands with no preference for any one, though he likes lemon-lime and orange drinks more than colas.

Marcus's interest in the promotion: Marcus has been saving up for a Playstation 2, and he's hoping he could buy games for it with Rocketcash (or maybe even win one!) Marcus tends to enter a lot of contests anyway -- he has limited spending money and is always hoping to get something for nothing.

David Lincoln
17 years old
Drinks Sprite, not loyal to it.

David's story: David lives in the suburbs of Sacramento, California with his mom. He just turned 17 and he's a senior. He's got good grades, and he's considering college. His mom won't let him work part-time like his friends do, so he's always a little strapped for cash.

David's relationship with Sprite: He used to drink it all the time, but he's getting older and is feeling like maybe he wants to change. He tried coffee and he doesn't like it -- he doesn't even like Coke! But he's started drinking Mountain Dew occasionally. He likes the commercials-- very edgy and exciting. Sprite still tastes the best to him, though.

David's interest in the promotion: He's always short on cash, but wants to keep up with his friends, get the latest CDs, cool clothes, and so on. He sees the promotion in his local store and thinks, "cool, I like Sprite anyway."

Sharee Thompson
12 yrs old
Sprite drinker.

Sharee's story: Sharee lives in Chicago with her Dad. Her Dad is a magazine editor and works late. He gives Sharee ten dollars a week in allowance, and though he pays for her clothes, he hates shopping so Sharee doesn't get to go as often as she would like.

Sharee's relationship with Sprite: Sharee drinks Sprite when given the choice of soft drinks, usually three to five times a week. Her dad buys the big two-liter bottles for home, but she'll buy a can or 16-ouncer sometimes when at school or out with friends.

Sharee's interest in the promotion: Sharee is often frustrated with her Dad, who doesn't have much time (or inclination) to take her shopping. She spends time online surfing clothes sites and looking for fashion advice.

From: http://www.carboniq.com

Chris Rothschild

Occupation: Lawyer **Age:** 40 **Technology:** Early Majority

Quote:*New toys! Ahh!*

Chris is a 40-year old tax lawyer with a crush on photography. He's been taking pictures semi-seriously since he was 25, but since he got into the tax law business, his equipment spending has surged somewhat, while the amount of time he can devote to photography has plunged.

He likes JoesPhoto.com for their personality. He feels like he knows Steve from the articles and discussions, and Chris is grateful to him for all the things that he's learned while being a part of the community. Joe's Photo also have pretty good, reliable product info, links to their own and others' reviews, product comparisons, etc., and he appreciates that a lot: Though he does make money now, he doesn't want to squander it on second-rate equipment, and he doesn't have time to do too much research himself. He knows that their prices are generally reasonable, so he doesn't get screwed, and he loves the free overnight shipping. He has a small set of things that he buys regularly, such as film and storage sleeves. He doesn't know a whole lot about computers, but he can generally read a manual and make things work, as he recently did with his Cablevision cable modem at home. And when he does get stuck, he knows who to call. He really likes the community features of JoesPhoto.com, so he asked in a discussion forum there what software they were using to run the site. Stanley pointed him in the right direction, and Chris managed to set up a ACS5 installation on a fresh VA Linux box for his office, that him and his 10 colleagues now use for collaboration. Apart from ACS5, they're using Windows NT, Office 2000 and Internet Explorer.

Chris is married and has two daughters at 6 and 4. He commutes between his house in Westchester and the Midtown law firm every day in his dark blue, 3 year old BMW 535.

Karen Gross

Occupation: Web Master **Age:** 28 **Technology:** Early Majority

Quote: *Stanley, I need this today! Please...?*

Karen owns the content on the site. And when she says *owns*, she means it. She's a perfectionist, always immaculately dressed, and she takes pride in making sure no typos, no incorrect information, and no bugs or other glitches ever find their way into her pages. When they do, she rallies the troops to get them fixed immediately.

Karen has managed web content for about 3 years, but joined Joe's Photo about four months ago. She is a workaholic and a micro-manager that likes to keep a close eye on the status of all portions of the site and in production. Karen has previously worked in Corporate Communications and pays close attention to the image that the Web site presents. She doesn't do any programming, but knows HTML and has dabbled a little in graphic design for the fun of it. She is very specific in what she wants, and she works closely with the template designer. Karen enjoys knowing the site looks and works perfectly. She is always looking for ways to improve it and wants immediate notification when things need changing.

She always takes time to workout every evening, her favorite are the spinning classes on Thursday nights. She lives in an apartment in Soho with her cat and her boyfriend, whom she's been seeing for a year now.

Steve Delaney

Occupation: Photographer and Author **Age:** 41 **Technology:** Laggard

Quote: *Did I tell you about when I took that picture of the two bears mating?*

Steve is a 41 year-old nature photographer who gained notoriety by writing articles and editorials for photography magazines. Joe's Photo hired him to write for the site and to get the community going. Most of the photos, articles and equipment reviews on the site are his, and he takes pride in answering questions from his readers.

While he gets completely carried away with photography itself, Steve is something of a conservative when it comes to technology. He's happy with his trusted friend, the Nikon F2, and is skeptic of high tech gee wiz auto-everything cameras. He is just now supplementing with an F4, and is even experimenting with a D1, as he can no longer deny the advantages.

As far as computers go, he mostly hates them. He's gotten the gnack of rebooting his Powerbook when something goes wrong. As part of his job, he had people come and set up DSL at his house. He finds web browsing to be much more useful now, although he still sometimes gets frustrated when he can't find any genuinely new and stimulating content He has learned MS Word 98 enough to open it, type, and save. He leaves formatting and editing to others.

He travels all over the globe to get the best shots possible, and he writes articles about his experiences. He'll send his articles and photos using whatever internet connection he can get, and he hates it, but if he didn't do it, he wouldn't get his expenses taken care of.

He's married, has two kids from his first marriage, and drives an old, beat-up, brown Jeep CJ7. Sports? Hell no!

Annette Chang

Occupation: Graphic Designer/Content Developer **Age:** 25 **Technology:** Early Adopter

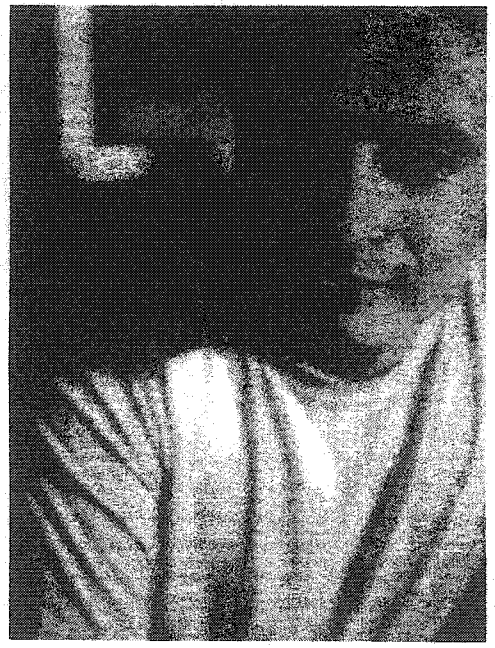
Quote: *Can't we make that design a little tighter?*

Annette has been a graphic designer for a couple years years, since she graduated from the School of Visual Arts. She digs a minimalist style, when she designs, when she dresses, and when she talks. She's been with JoesPhoto.com for about a year now, and does all their graphic design.

She's a wizard with Photoshop 6.0 and Golive 5.0 on her Mac G3 with a 22" Apple Cinema Studio display. That said, that's pretty much all that she really feels comfortable with, when it comes to software, and she *hates* it when computers make her feel stupid.

Since there isn't really that much graphic design work to do *all* the time, and with the job market being so sluggish and all, Pamela, the Head Honcho, has convinced her to take on responsibility for content development, code word for copy writing. It's not really her favorite kind of work, but it pays the rent, and there's a certain satisfaction in doing some braindead work when her creative self has taken a vacation.

When she gets fed up with copywriting, she helps out with interior design on her girlfriend's hip lower east side bar. She rides the subway to work from her cool Williamsburg loft, which she shares with a couple friends from school.



Stanley Jones

Occupation: Web Developer **Age:** 24 **Technology:** Innovator

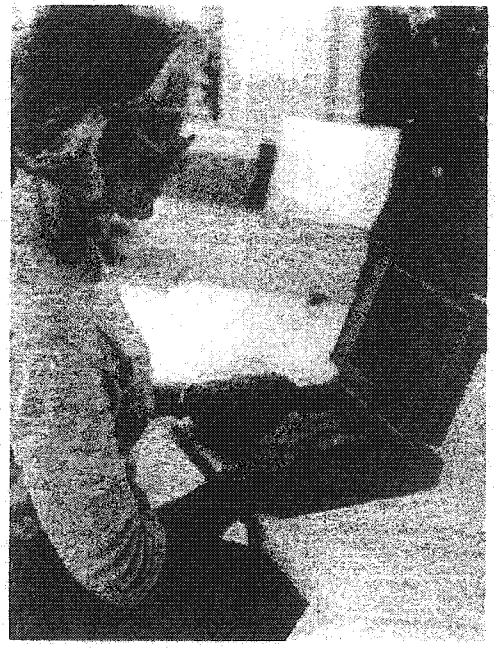
Quote: *Yeah, this web-thing is kinda fun.*

Stanley's a laid-back, 24-year old web developer who graduated two years ago from UMich with a degree in CS. Stanley's best time on the job is when he solves a tough problem with his friends and stays up all night implementing it. After an all-nighter Stanley and his friends love to head down to Ray's Famous Pizza #13 and grab a couple of slices each.

He's a bright kid that doesn't need much by way of motivation, but his ambitions aren't up in the sky. He likes the low-key kind of job he has, staying close to the users of the site, finding ways to make their interactions more meaningful and entertaining. He likes taking photos with his digital camera, and enjoys learning from Steve and the community.

He uses Linux, Emacs and CVS for everything that he can, and is annoyed when Heidi or Steve make him use Word or Excel. His software architecture skills aren't a whole lot to brag about, but he doesn't really care. He can make a site run, pretty much all by himself, and that's nothing to be ashamed of. He's not exactly an Oracle DBA or a sysadmin wiz, but he can fix problems when he needs to, and he knows who and what to ask when he gets stuck.

He recently got a Nomad Jukebox MP3 player with 6 GB disk space, and though he's upset about the shut-down of Napster, he's happily chugging away on Gnutella. He lives in an apartment near Prospect Park in Brooklyn with his friend from college. He doesn't need a car: He has a MetroCard.



Betty White

Occupation: Editor **Age:** 48 **Technology:** Laggard

Quote: *20 years ago, people could* really *write!*

Betty has a lot of experience with language. On a good day, when she's on vacation, she'll finish several novels in a single day. And she'll remember almost every single word, too. Betty just has a thing with words.

She lives with her husband and two kids in a house in Greatneck. Her job isn't what she'd really like to be doing. She'd prefer a job at a literature publisher somewhere, but the competition is so fierce. And there is a certain pleasure in learning about photography.

She doesn't particularly like computers, but she doesn't hate them, either. She's gotten used to use Word, and Karen has made her learn how to use exactly three different HTML tags, and that's about it. When there's something she can't figure out how to do, she either forgets about it or asks someone who knows.

From: http://ccm.redhat.com/user-centered/personas.html

Appendix B: Usability and User Satisfaction Questionnaire

Personas as a tool for the design of interactive systems

Adapted from: Usability Scale © Digital Equipment Corporation, 1986.

	Strongly disagree				Strongly agree
1. I think that I would like to use personas frequently	1	2	3	4	5
2. I found personas unnecessarily complex	1	2	3	4	5
3. I thought personas were easy to use	1	2	3	4	5
4. I think that I would need the support of a HCI specialist to be able to use them	1	2	3	4	5
5. I found the various features/information of personas were well integrated	1	2	3	4	5
6. I thought there was too much inconsistency in personas	1	2	3	4	5
7. I would imagine that most people would learn to use personas very quickly	1	2	3	4	5
8. I found personas very cumbersome to use	1	2	3	4	5
9. I felt very confident using personas	1	2	3	4	5
10. I needed to learn a lot of things before I could get going with personas	1	2	3	4	5