

The origin of personas

 cooper.com/journal/2008/05/the_origin_of_personas

Alan Cooper

May 15,
2008

The Inmates Are Running the Asylum, published in 1998, introduced the use of personas as a practical interaction design tool. Based on the single-chapter discussion in that book, personas rapidly gained popularity in the software industry due to their unusual power and effectiveness. Had personas been developed in the laboratory, the full story of how they came to be would have been published long ago, but since their use developed over many years in both my practice as a software inventor and architectural consultant and the consulting work of Cooper designers, that is not the case. Since *Inmates* was published, many people have asked for the history of Cooper personas, and here it is.

In their book, *Fire in the Valley*, authors Paul Freiberger and Mike Swaine, credit me with writing the “first serious business software for microcomputers” as far back as 1975. Like so much software of the time, it was terribly hard to use, and its real power was in demonstrating that making software easy to use was harder than everyone thought. Despite my commitment to making software more user-friendly, it wasn’t until 1983 and about 15 major business and personal applications later that I began to develop a more effective approach.

I was writing a critical-path project management program that I called “Plan*It.” Early in the project, I interviewed about seven or eight colleagues and acquaintances who were likely candidates to use a project management program. In particular, I spoke at length with a woman named Kathy who worked at Carlick Advertising. Kathy’s job was called “traffic,” and it was her responsibility to assure that projects were staffed and staffers fully utilized. It seemed a classic project management task. Kathy was the basis for my first, primitive, persona.

In 1983, compared to what we use today, computers were very small, slow, and weak. It was normal for a large program the size of Plan*It to take an hour or more just to compile in its entirety. I usually performed a full compilation at least once a day around lunchtime. At the time I lived in Monterey, California, near the classically beautiful Old Del Monte golf course. After eating, while my computer chugged away compiling the source code, I would walk the golf course. From my home near the ninth hole, I could traverse almost the entire course without attracting much attention from the clubhouse. During those walks I designed my program.

As I walked, I would engage myself in a dialogue, play-acting a project manager, loosely based on Kathy, requesting functions and behavior from my program. I often found myself deep in those dialogues, speaking aloud, and gesturing with my arms. Some of the golfers were taken aback by my unexpected presence and unusual behavior, but that didn’t bother me because I found that this play-acting technique was remarkably

effective for cutting through complex design questions of functionality and interaction, allowing me to clearly see what was necessary and unnecessary and, more importantly, to differentiate between what was used frequently and what was needed only infrequently.

Plan*It was eventually sold to Computer Associates and released to the public as SuperProject, which became a critical and commercial success. For several years it was the dominant project management program for personal computers, and its design is the model for today's market leading product from Microsoft: Project.

Another product I designed using a similar technique was a visual programming language code-named "Ruby." I eventually sold Ruby to Bill Gates, who married it to his QuickBASIC language to create Visual Basic, the paradigm-defining product that presented an old language in a revolutionary new way. I designed it for a single user loosely modeled after an IT Manager working at Bank of America's world headquarters in Concord, California.

In 1990, instead of creating another program, I offered my software design experience to my colleagues as a consultant for the first time. I quickly learned that consulting is quite different from entrepreneurial invention. Previously, I just built what I thought was right. Now I found that I had to persuade my clients before they would follow my lead or see the benefits of my ideas. This imperative for communication eventually impelled me to formalize the notion of personas.

In 1995 I was working with the three founders of Sagent Technologies, pioneers in the field of what is now called "Business Intelligence" software. During discussions with them regarding interaction design for their product, I found myself continually engaged in a circular dialogue. I would ask them for a specific example of how someone would use their program. The answer would invariably follow this characteristic pattern: "Well, someone could create a crosstab of sales information...but it could be a chart, or if it were marketing data they could present it as a table. They could do anything!" It was almost impossible for those brilliant, logical programmers to conceive of a single use of their product when it was obviously capable of so many uses. In frustration I demanded to be introduced to their customers.

At the time, the company was new, and the product wasn't yet released, but the founders had well-established business relationships from three previous companies. Alice Blair, Sagent's head of product marketing, took me on a tour of a half-dozen or so of their intended clients. One of them was a financial analyst for a large university. One was a marketing analyst for a software firm. Some were computer savvy and some were not. I asked them open-ended questions to learn about their jobs and what they were trying to achieve by studying these large quantities of data. Even though the variation among the users was dramatic, a clear pattern emerged after just a few interviews. The users fell into three distinct groups, clearly differentiated by their goals, tasks, and skill levels. Had I been creating the software myself, I would have role-played those users as I

had with Ruby and SuperProject, but in this case I had to describe those user models to the Sagent team. So I created Chuck, Cynthia, and Rob. These three were the first true, Goal-Directed, personas.

Chuck was an analyst who used ready-built templates and reports. Cynthia was an analyst, too, and she used similar ready-built templates. But Cynthia also wrote her own templates, which she gave to Chuck to use. Rob was the IT manager who supported both Rob and Cynthia. He could optimize Cynthia's templates, but he would never originate or use them.

At the next group meeting, I presented my designs from the points of view of Chuck, Cynthia, and Rob instead of from my own. The results were dramatic. While there was still resistance to this unfamiliar method, the programmers could clearly see the sense in my designs because they could identify with these hypothetical archetypes. From then on, I always presented design work using one of the personas, and eventually even the Sagent engineers began to talk about "what Cynthia would do" or "whether Chuck could understand" some dialog box. When the product eventually came to market, its interface was clumsy in places, but it possessed a fundamentally clear, three-faced interaction structure: Chuck's interface allowed him to select and use ready-built templates and reports; Cynthia's interface allowed her to design and publish templates; and Rob's interface allowed him to optimize the performance of Cynthia's templates without altering their content or behavior. The product was so successful that it defined a new product segment. The company was a success, too, going public four years later.

The effectiveness of Chuck, Cynthia, and Rob as design tools for me and communication tools for the entire construction team was obvious and significant. I began to use personas in all of my design projects, as did the designers at Cooper. Over the next few years, we developed and perfected the technique. It became our secret interaction design weapon. Several people counseled me to keep the notion of personas a secret, arguing that it would give me a competitive advantage. But my reasoning went in the other direction. Personas were so simple, clear, and powerful that it seemed only a matter of time before other practitioners discovered the technique for themselves. When this happened, I would lose my competitive advantage anyway, but if I disclosed what I knew about personas, at least I could receive some credit for contributions to an industry that I loved. This is what prompted me to write about personas in *Inmates*.

The book was intended to alert managers to the problems inherent in designing software for use by non-engineers. It was never meant as a "how-to" book for interaction designers. However, at the time the practice of interaction design was relatively unknown, and my methods in particular were so unfamiliar, I felt it necessary to add some description of the Goal-Directed methodology. My goal was simply to demonstrate that there was indeed some substance to my method: that it was different, that it was effective, that it was real.

I'm tempted to say that personas are counter-intuitive, but it would be more accurate to say that they are counter-logical. I suspect that this is why they originated in practice rather than in the laboratory or in academia. If, responding to the directive *design for the user*, you follow logic, you tend to canvass the user community, collect their requests for functions, and then provide them a product containing all of those functions. I call this "the sum of all desired features." There is abundant empirical proof that this solution is only marginally effective at best. The problem is that while logic is a powerful and effective programming tool, it is a pathetically weak and inappropriate interaction design tool.

Many of my predecessors have employed ethnographic user research and created persona-like constructs to aid their designing. Product marketing professionals have also been using persona-like entities for many years to define demographic segments. But personas are unique and uniquely effective.

Personas, like all powerful tools, can be grasped in an instant but can take months or years to master. Interaction designers at Cooper spend weeks of study and months of practice before we consider them to be capable of creating and using personas at a professional level. Many practicing designers have used the brief 25-page description of personas in *Inmates* as a "Persona How-to" manual, but a complete "How-to" on personas has yet to be written. I hope someday that one of the very accomplished architects at Cooper will write that book because they have developed the technique to a degree of sophistication well beyond my seminal efforts. I look forward to contributing to it.

Want to learn about creating personas? Register for Cooper Professional Education's [Design Research Techniques](#) course.