

Hello — and thanks for taking a look at my work!

My name is Rich Kurz and I am an experienced graphics design professional. My philosophy has grown simpler through the years. Good design is not about me, but about us. I want to do good work that serves the needs of my client and that I am proud to put in my portfolio. This pdf shows some of my capabilities.

Note that all concepts and initial drawings are owned by the Hewlett-Packard Co. The images in this document are my own and are under copyright to Rich Kurz.

Working at HP was a wonderful experience back in the day. There was lots of freedom to flex one's creative muscle and to try things out, and even forgiveness if things just did not work out. But not everything that was asked for made it into the customers' hands - or even past a first round review. Such was the case here.

After a brainstorming session about new ideas to market the emulator product line, it was my turn to produce some visuals illustrating the different feature/benefit statements from the session. The ideas were flowing, but they went no further. Chalk it up to preoccupation with more pressing matters. Too bad. I was really looking forward to turning them into photos and ads.

The first four pages are notes from the brainstorming session that I recorded. Then comes the sketches, sometimes identified by their feature.

Deliverables: 8-1/2" x 11", marker on tracing paper, mounted on bond paper

I am available to discuss your design, illustration, marketing, and advertising needs. Let's talk!



Rich Kurz

ATT, just right.

Wafer with bubbles containing our tools, high-levels of integration in semiconductors.

Line of tools to take you into the 21st century - we're here for a long time.

Focus on software implementation tools - x development systems.

X development a DIFFICULT JOB - need best tools.

Welding two dissimilar pieces of metal with our tools - ditch on side with discarded pieces.

Don't need to know uP backwards and forwards.

Breaking new ground - barrier.

We promised you the complete solution and we delivered.
(is this dangerous because we axed the VAX?)

You don't have to be a hardware designer to use our software tools.

"No" sign with soldering iron in it or screw driver.

Gone from assembly language to C.

Release standards - is quality a problem?
Is a third of your lab working on defects?
Is your maintenance team larger than your software development team?

Lawn with big dandelion. The longer you wait to fix this problem/defect, the worse it's going to get. What if you leave your software process the way it is today?

Axxx T-shirt image - conquer, supremacy, i.e. a non-verbal.

We spend alot of \$ so you don't have to.

What if your customer finds it (defect) first?

FURPS (is this proprietary?)

Notes from first SE products brainstorming session:

General comments on se tools:

- Keep SE tied in with instruments, emulators, analyzers.
- Don't lose track of selling systems.
- Ideas: Two skulling teams - overload one end of one boat.
 - One of few measures what you're doing.
 - How do you know when you're done if you don't know what you've done.
 - If you don't have the right debug tools, system integration will eat your lunch.
 - Caveman hacking off corners on a square shape - making a circular shape.
 - Wheelbarrow with square wheel (other vendor solutions).
 - Another with circular wheel (HP tools).

Specific Product Comments: keywords

AxLS: (compiler)

Standards

Reliability

High-performance (see data sheet)

Integration (i.e. emuls. customers - v.p.t.)

Optimizing

- Ideas: someone chasing a bug/defect - PACMAN/flyswatter with big mesh or big holes in it/sharp tools/best tools for chasing bugs/use the right kind of tools for the job.
- Big bugs one guy with inappropriate weapon and one guy with big weapon (HP tools).
- New self/old self
- Cross features AxLS
- Ease of use/productive/working - se generally
- Tools for productivity - se generally.
- Pay now or pay later . . . & later.
- Chart about finding defects in different stages of product development/\$.

AxDB (sw designer/debugger)

Ease of use

Flexible

Multi-stage usability

AxLS parallels

Hi-level & low-level views

Don't need two languages to write and debug - use same one.

More photogenic than some of the other Axxx products.

Supports two methods for software development. (Conventional time-shared

workstation costs less \$ costs more time. Evolving to one station per user (SUN). VAX/HP best of both worlds.

AxTS (test system Basis Branch Analyzer)

Ability to measure "metric"

Communication - among team - feeling good about where you are.

Levels in family of products.

Integration of product set Axxx.

Tied in to sources - don't have to go back and forth.

Emphasize tools made to work together.

Reducing/eliminating risk.

Ideas: Symphony - when it all plays together, it can all sound beautiful.

Teamwork:

Control development cycle, project.

Communications and organization.

Design bugs worst in project.

Hi-leverage.

warning - not as integrated as rest of tools

Manage/reduce/eliminate/complexity

Methods/tools what to know

Methodology sale.

Picture - oriented - picture worth a thousand words.

Managing project.

Ideas: Right hand knows what left hand is doing.

Tightrope walker and pole.

Team coordination.

Concepts: (se general)

Expanding beyond CAE solution 1979-85-88

Emulation and analysis and soda, spa & . . . & today.

Like Virginia Slims ad "long way baby"

Past and present i.e. progress.

Software development tools.

When it's your head on the block wouldn't you want to know it's been done right?

More complete spectrum of products than competition. Spectrum could be a visual.

HP arm cracking workstation in a vice.

ATT, just right.

Wafer with bubbles containing our tools, high-levels of integration in semiconductors.

Line of tools to take you into the 21st century - we're here for a long time.

Focus on software implementation tools - x development systems.

X development a DIFFICULT JOB - need best tools.

Welding two dissimilar pieces of metal with our tools - ditch on side with discarded pieces.

Don't need to know uP backwards and forwards.

Breaking new ground - barrier.

We promised you the complete solution and we delivered.
(is this dangerous because we axed the VAX?)

You don't have to be a hardware designer to use our software tools.

"No" sign with soldering iron in it or screw driver.

Gone from assembly language to C.

Release standards - is quality a problem?
Is a third of your lab working on defects?
Is your maintenance team larger than your software development team?

Lawn with big dandelion. The longer you wait to fix this problem/defect, the worse it's going to get. What if you leave your software process the way it is today?

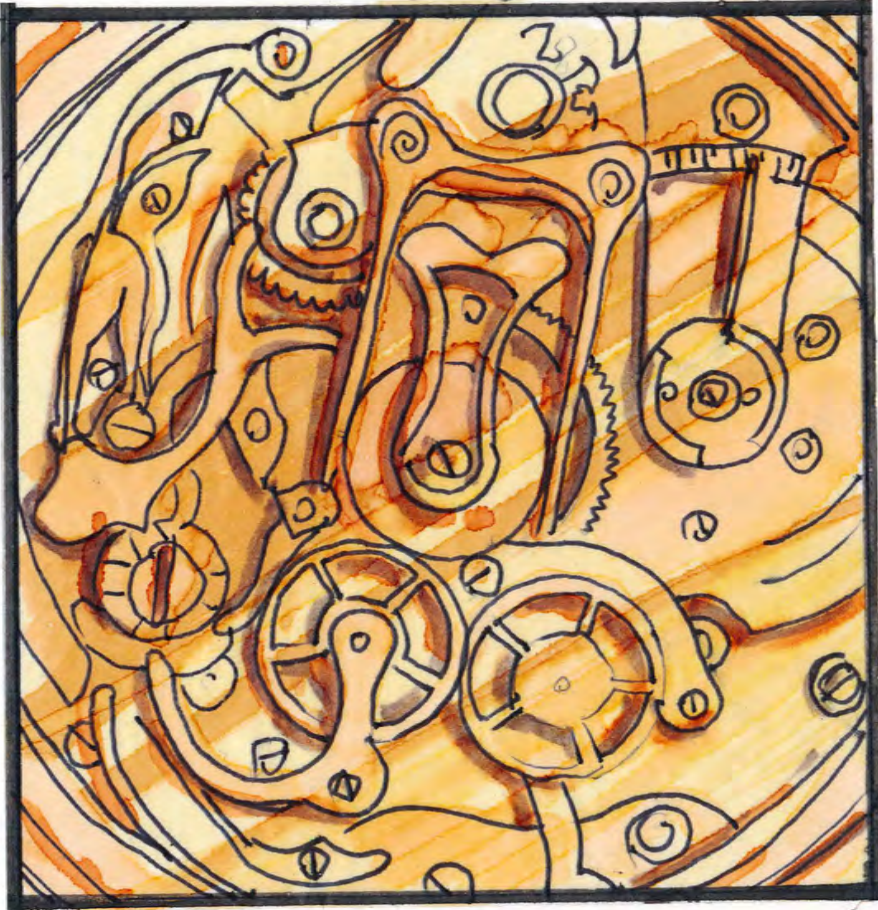
Axxx T-shirt image - conquer, supremacy, i.e. a non-verbal.

We spend alot of \$ so you don't have to.

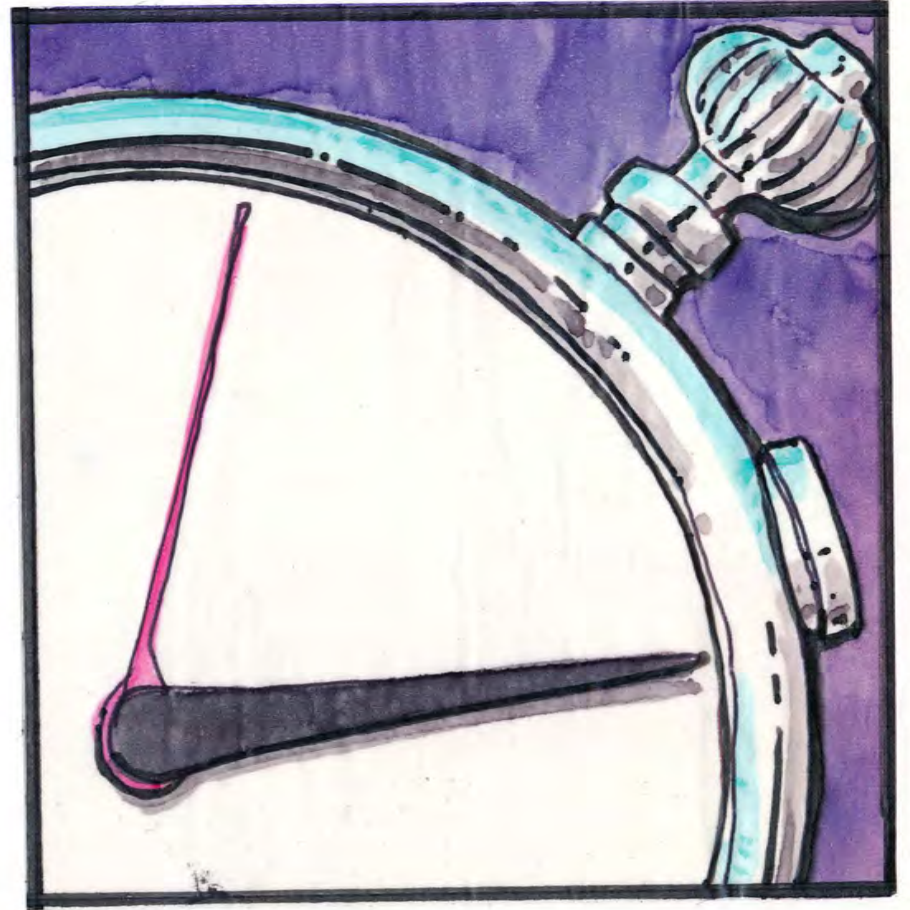
What if your customer finds it (defect) first?

FURPS (is this proprietary?)

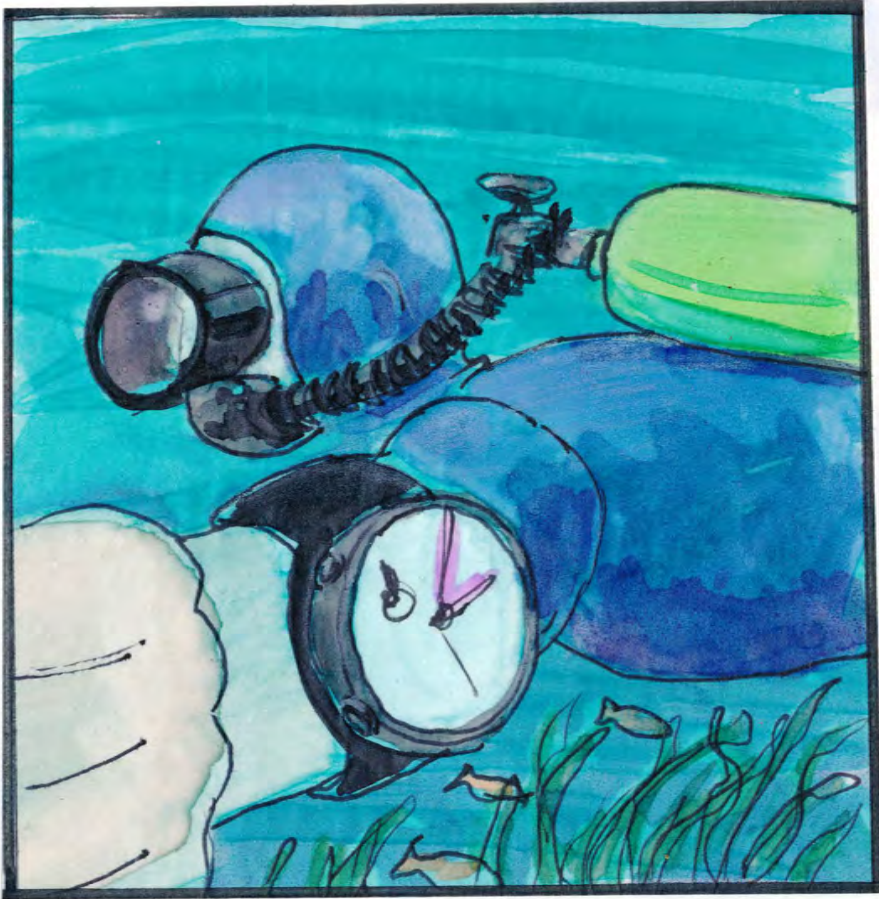
functionality



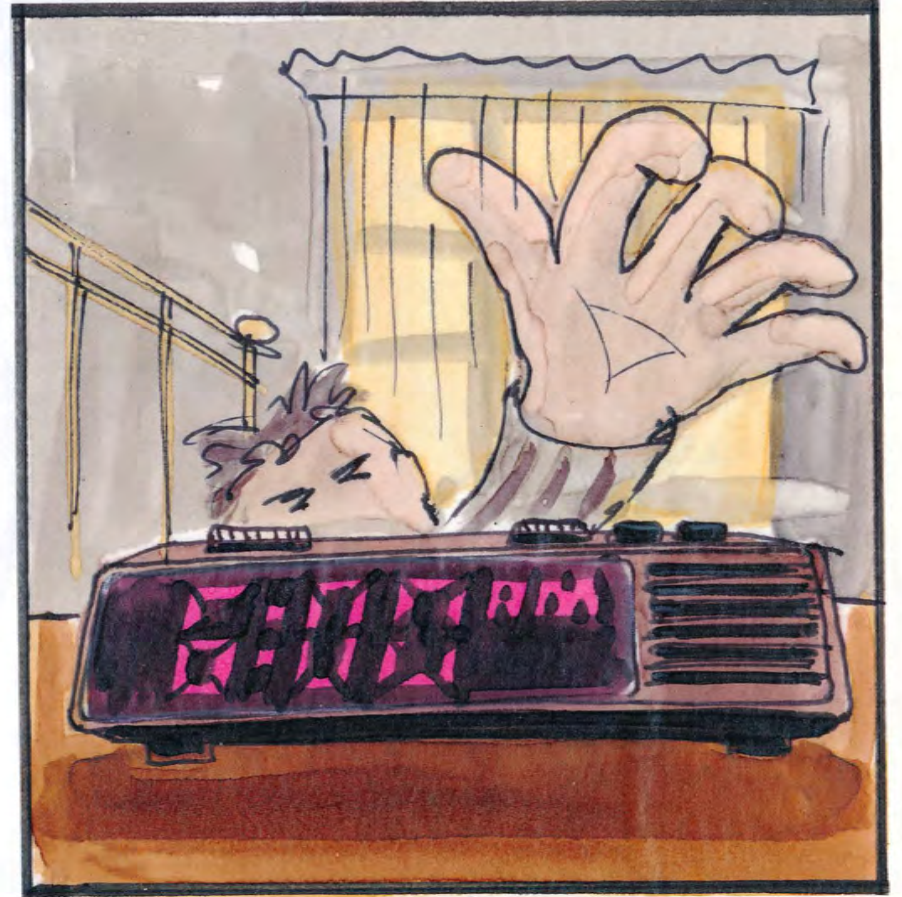
usability



reliability



productivity



supportability



