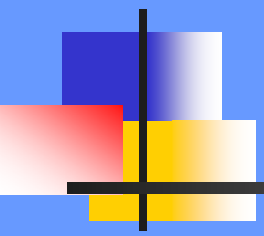


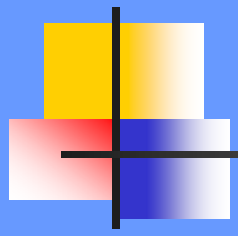
Shifting Innovation to Your Customers via Toolkits for User Innovation

Professor Eric von Hippel

MIT Sloan School of Management

To develop a product or service, information about needs and about solutions must be brought together at a single site.

- **Need** information is usually found at user sites.
- **Solution** information is usually found at manufacturer sites.



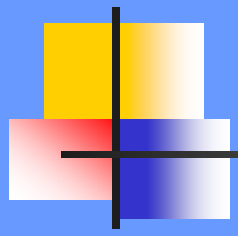
Information is often "sticky"

But need and/or solution information can be very costly to transfer from site to site – is often very "sticky."

Some reasons:

- Information needed by developers may be ***tacit***
  - Can you tell your child how to ride a bike?

- A ***lot*** of information is often needed by developers
  - "You didn't tell me you were going to use the product *that* way!"



Impact of sticky information #1

1. Product or service design should move to the site of sticky information, "other things being equal."

That is:

- If need information is very sticky, and solution information is not, product design should be done at the user site;
- If solution information is very sticky, and need information is not, product design should be done at the manufacturer site (The traditional pattern).

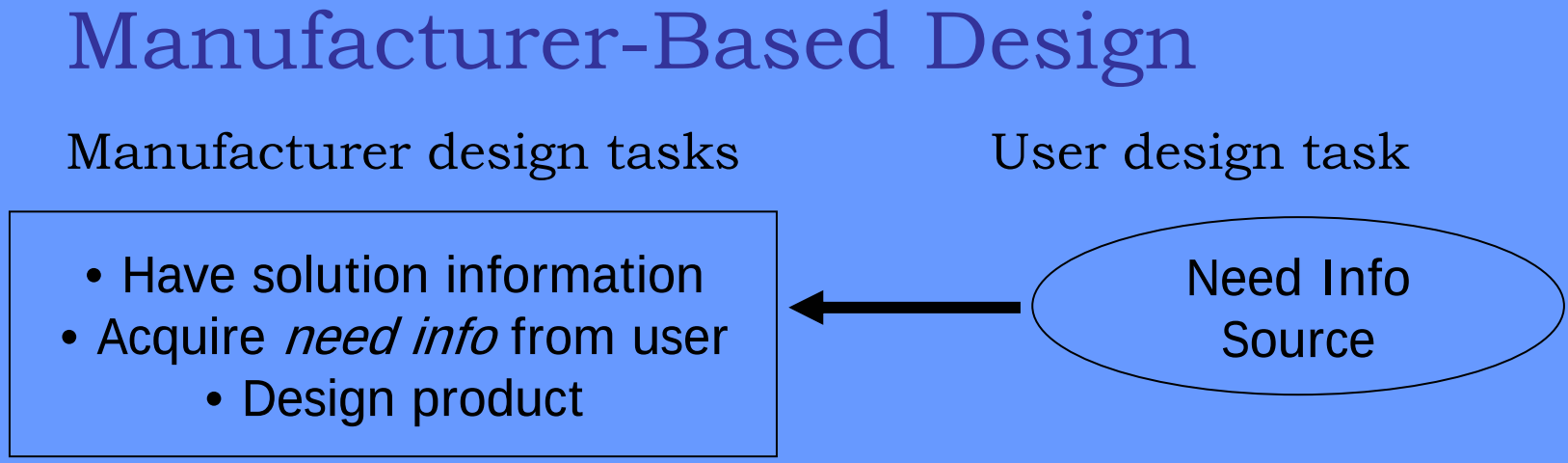
Problem-solving *does* move to sticky information sites

Sample of 24 inventory control system innovations by Seven-Eleven Japan and NEC

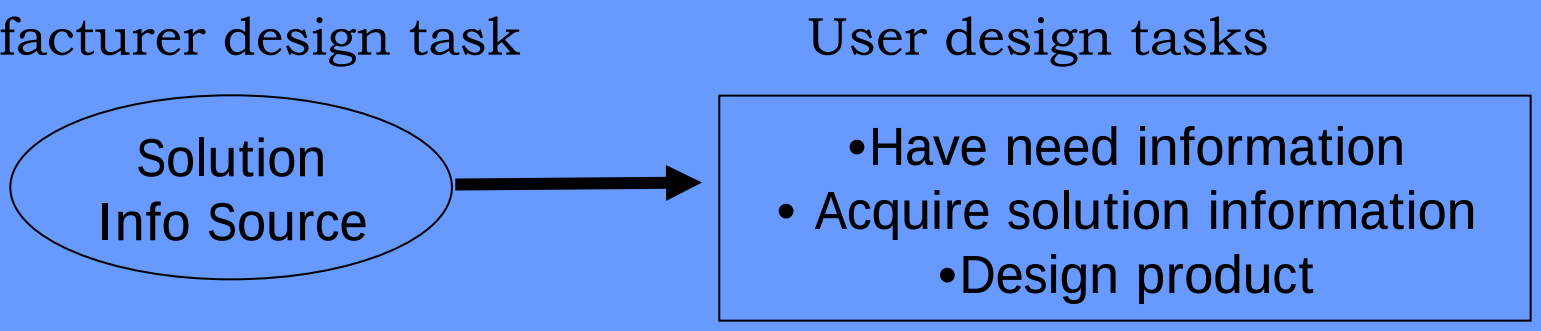
(For this diagram, see:

Ogawa, Susumu. *Does sticky information affect the locus of innovation? Evidence from the Japanese convenience-store industry*. *Research Policy* 26, 7-8, April 1998. Figure 1, p. 78.)

Manufacturer-Based Design



User-Based Design



Example of the impact of sticky information on the locus of innovation:

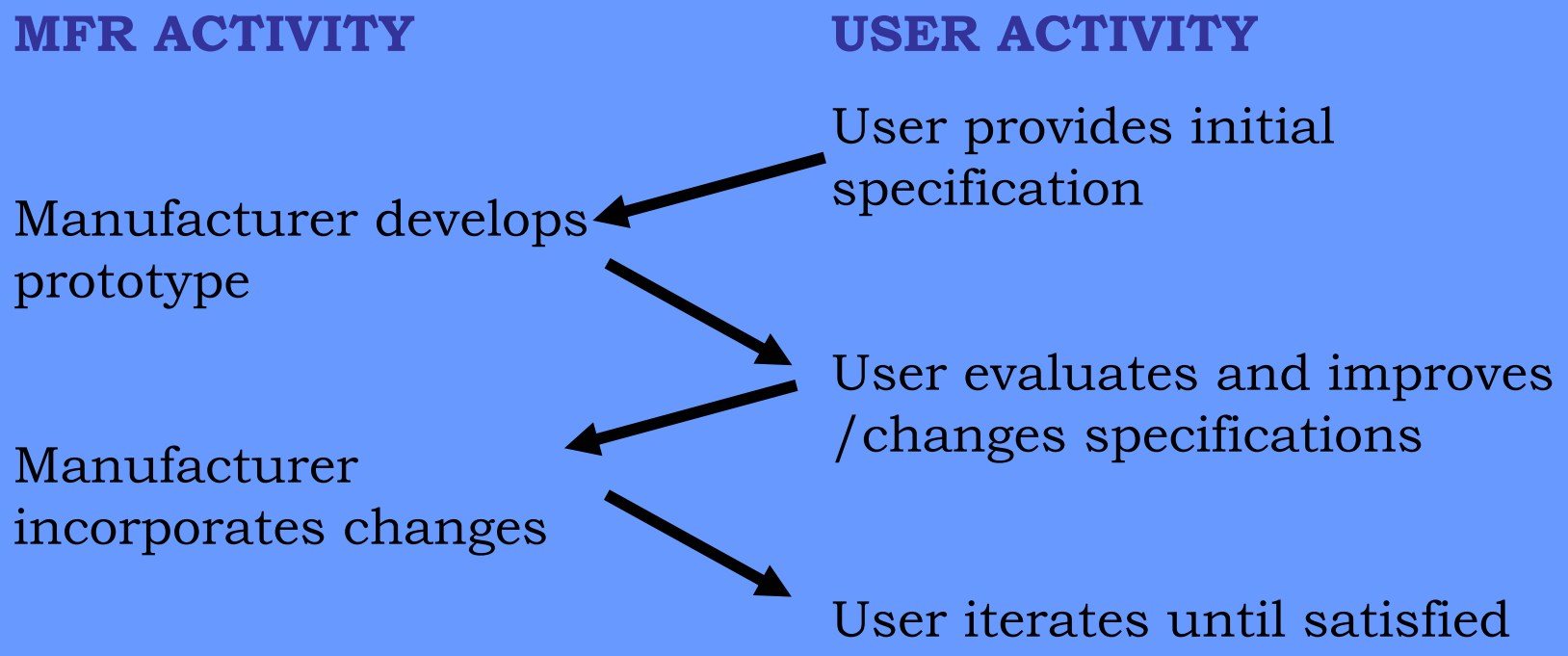
Fifty percent of all prescriptions written in the U.S. are written for "off-label" uses of prescription drugs

- **New prescription drugs are generally developed in the labs of pharmaceutical firms** – sites where much specialized information about drug development has been build up over the years.

- **Off-label applications are generally found by patients and physicians.** They apply the drugs many times under widely varying field conditions – and discover unanticipated positive (or negative) effects thereby. ("Doctor: this blood pressure medication you gave me is causing my hair to regrow!")

Impact of sticky information #2

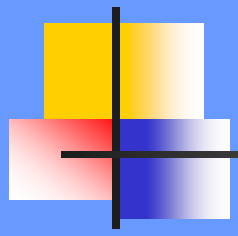
2. If both need and solution information are sticky, problem-solving activity will tend to ***iterate*** between user and manufacturer sites, as information from each site is drawn upon for problem-solving



Evidence for repeated site shifts during problem solving

(For this chart, see:

von Hippel, Eric and Marcie J. Tyre. *How Learning by Doing Is Done: Problem Identification in Novel Process Equipment. Research Policy*. 1994.)



How can you reduce iteration?

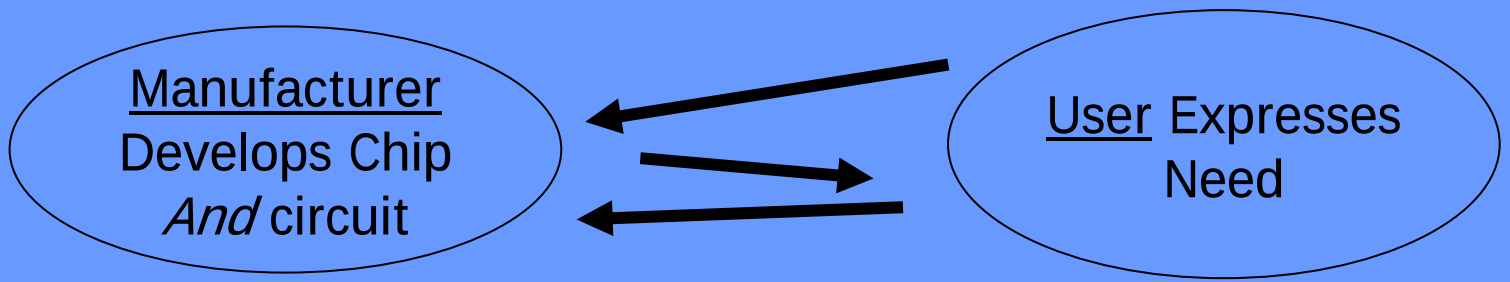
Repeated shifts of problem-solving sites during product development can be very costly – what can you do to reduce the need for it?

3. Reframe the initial product or service design problem which draws on two sticky information sites into sub-problems – each of which draws on sticky information location at only one site

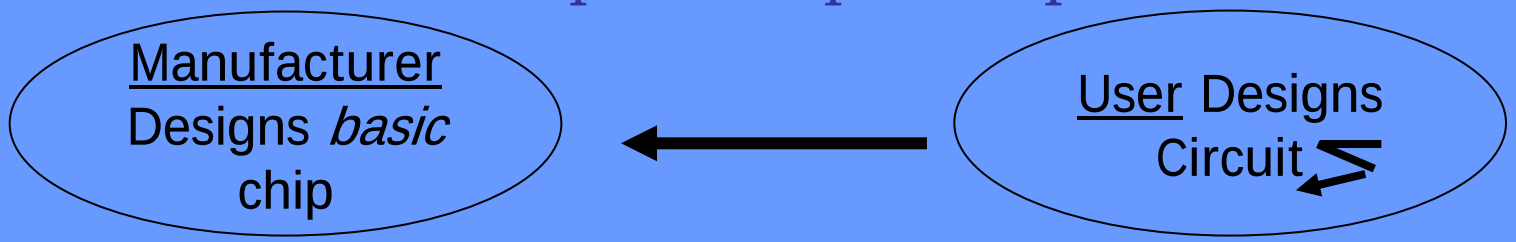
Example: Custom Integrated Circuit Design

"Full Custom" IC design vs ASIC / FPLD Design

"Full custom" chip development procedure

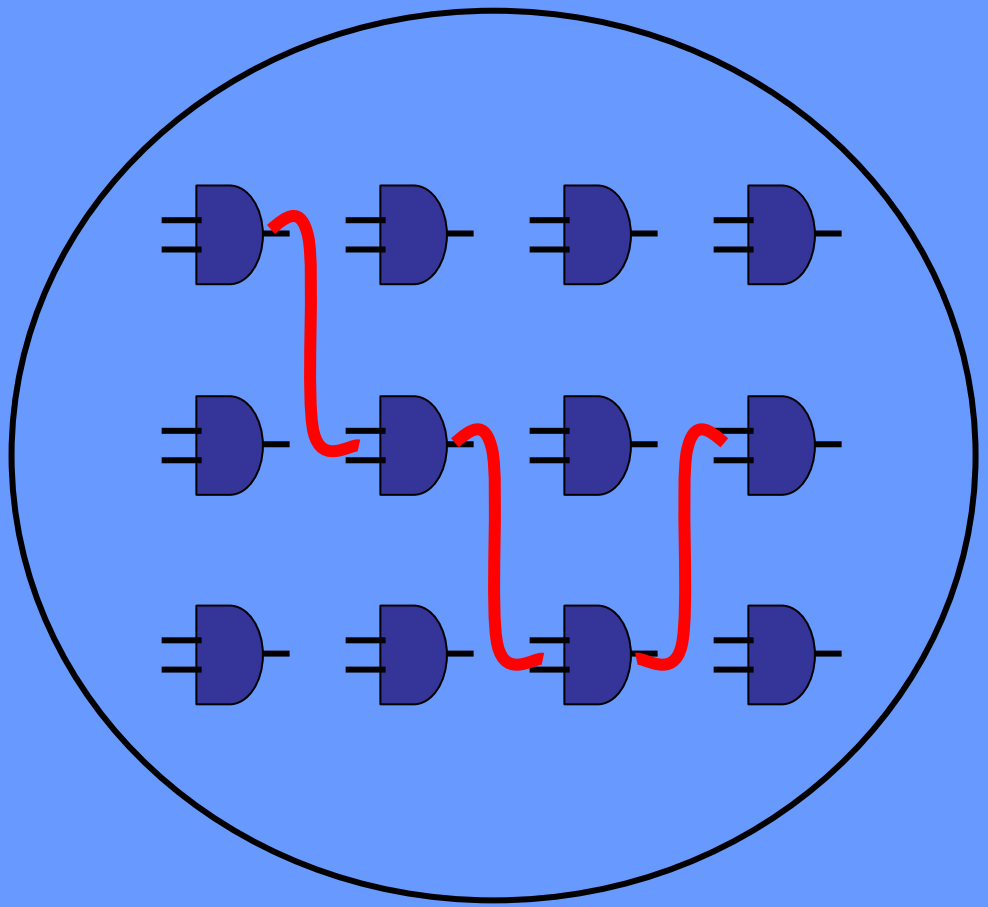


ASIC custom chip development procedure



Example

"Full-custom" IC Design vs "Gate Array IC Designs"

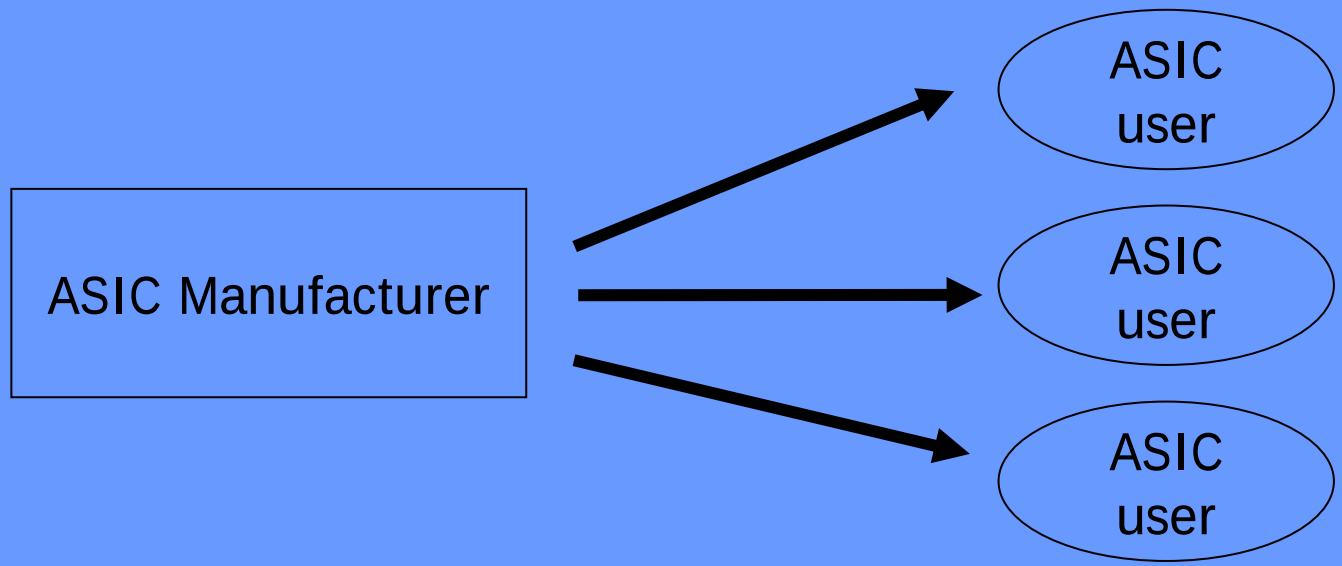


Why all this leads to toolkits

Economics of sticky information tends to shift the locus of problem-solving to users. For custom design projects, manufacturer information is standard from project to project but user need differs

Example:

Each ASIC design may require the same information from the ASIC manufacturer, but unique information from the ASIC user.

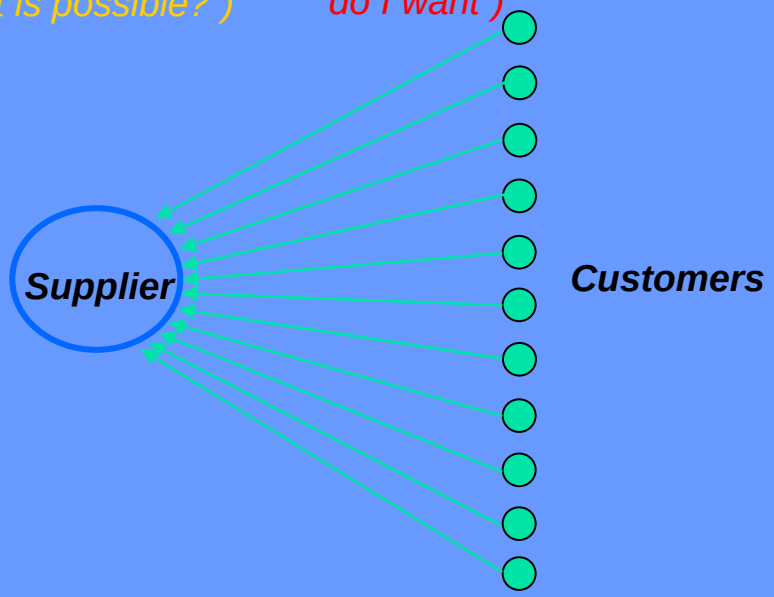


With toolkits customers – not manufacturers - need to "understand customer need"

"Find a need and fill it" model

Solution Information *("What is possible?")*

Need Information *("What do I want")*

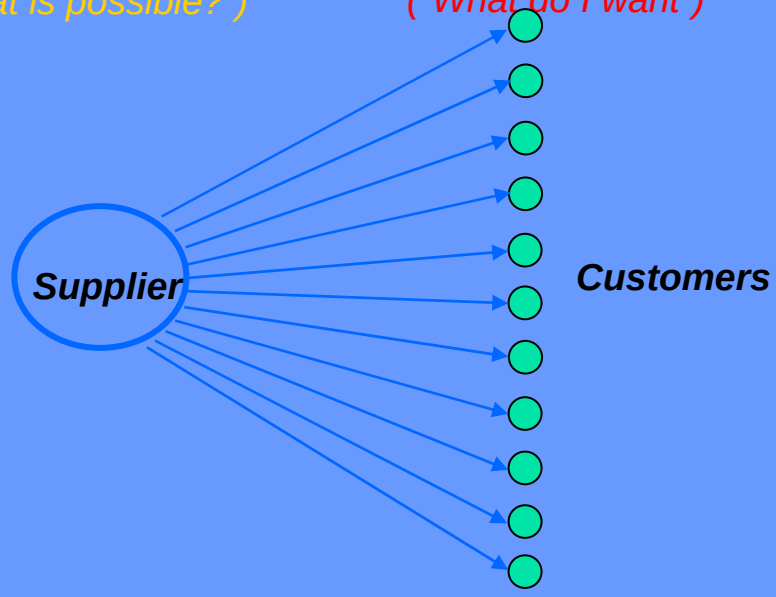


"Market Research"

Toolkits model reverses information flow

Solution Information *("What is possible?")*

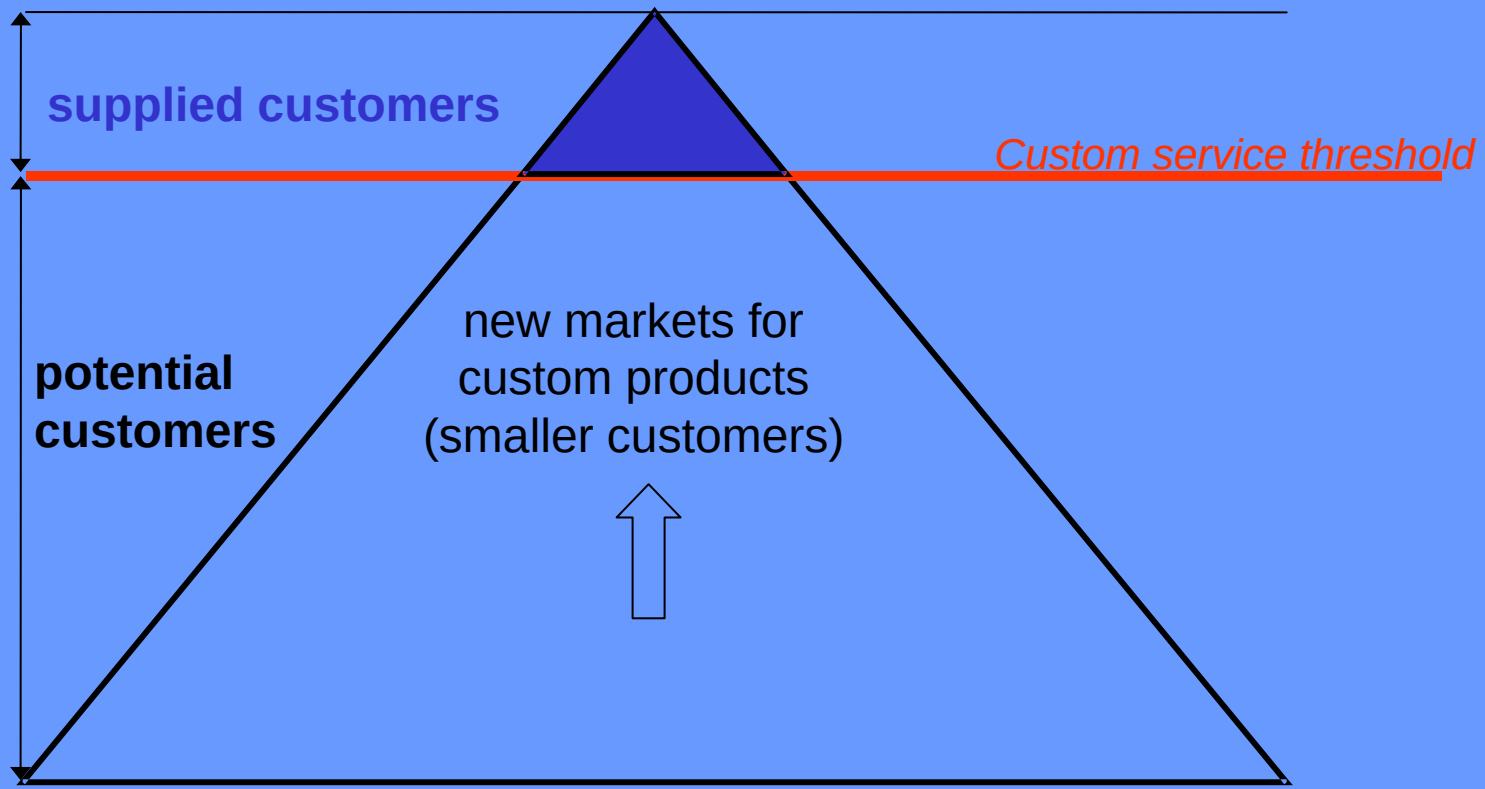
Need Information *("What do I want")*



"Innovation Toolkits"

You can't afford to understand the needs of smaller customers

Companies cannot afford to design custom solutions for smaller customers

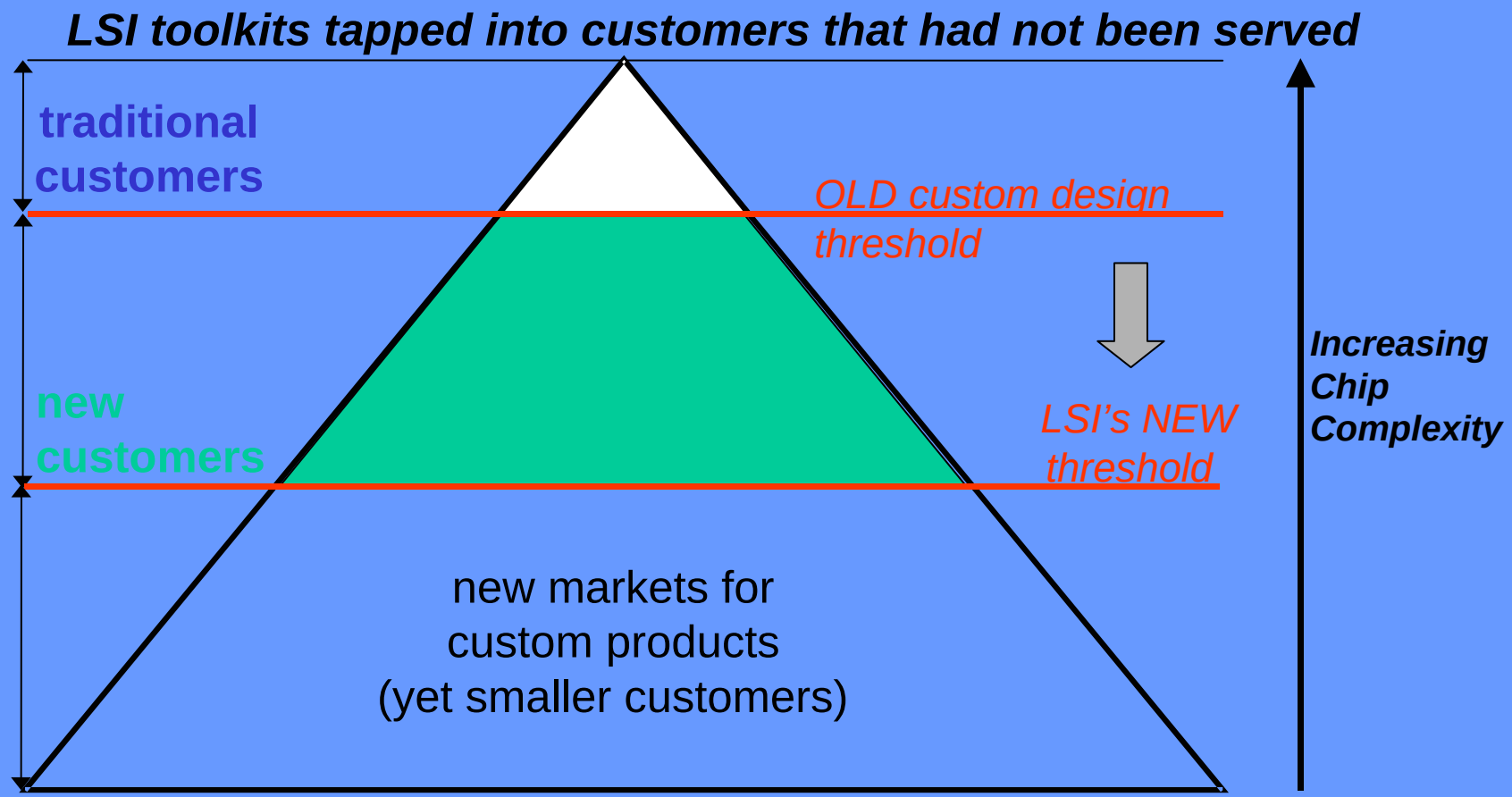


The Solution: LSI's Development Toolkit

- Customers design chips that are produced by LSI
- User-friendly and integrated toolkit (using simulation and CAD technology)
- Traditional suppliers were reluctant to make tools available to markets (intellectual property)
- Fujitsu even refused to share its tools with US division

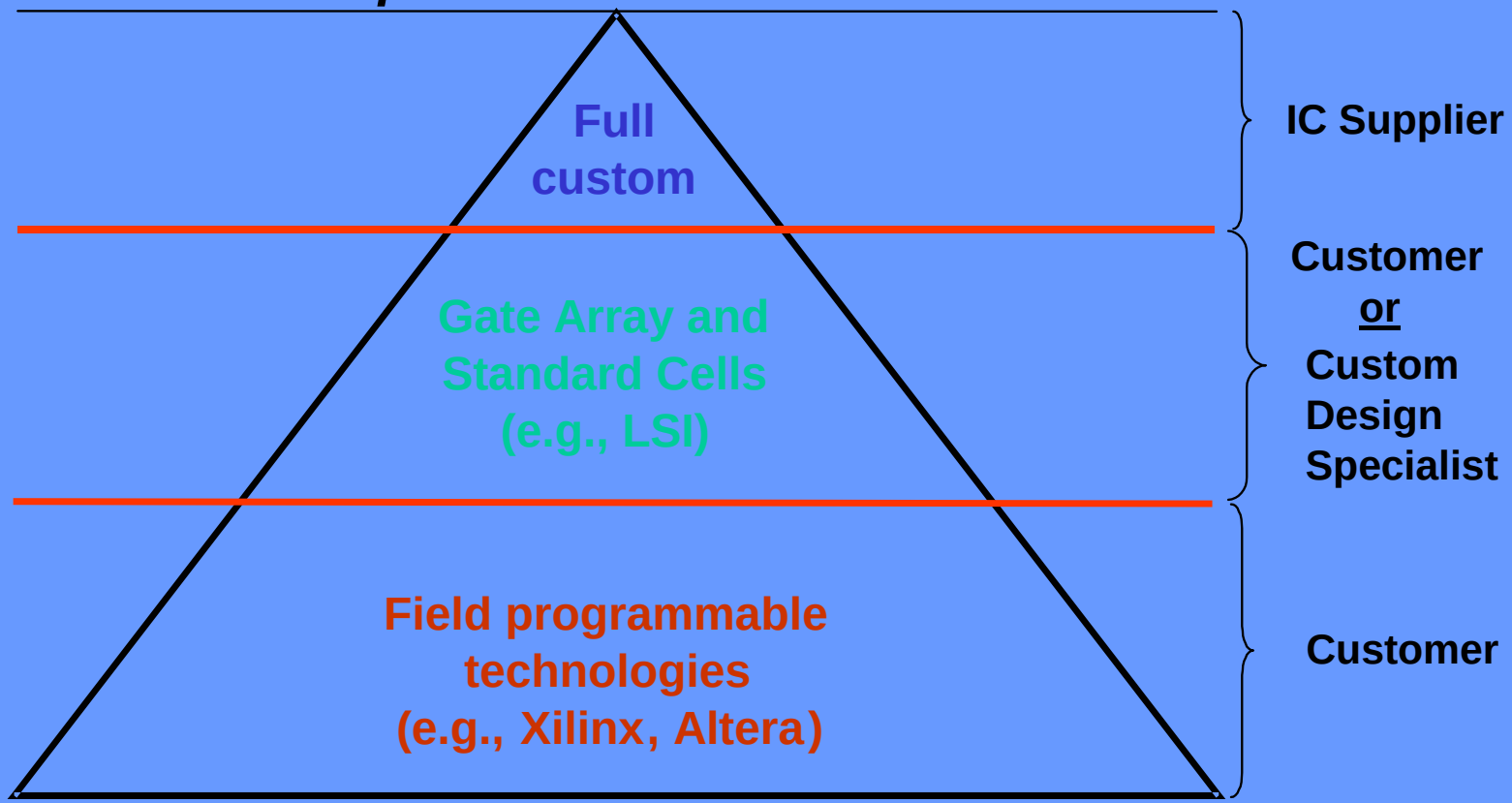
(Image of an advertisement by LSI Logic Corporation with the headline, "Design Our Gate Arrays On Your Workstation".)

Innovation toolkits made many more customers accessible to LSI



The Pattern is Repeated: The Rise of Field Programmable Technologies

Next Steps: Where is the New Growth?



Chip Designs Typically By:

- IC Supplier
- Customer **or** Custom Design Specialist
- Customer

Customers Increasingly Using Toolkits and Designing their Own Custom ICs

(For this World Semiconductor Trade Statistics chart, see:

Thomke, Stefan, and Eric von Hippel. *Customers as Innovators: A New Way to Create Value*. Harvard Business Review, April 2002. Reprint No. R0204F.)

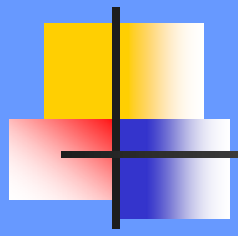
2 major tasks for toolkit development

A. Separate out development tasks that are custom "need-information –intensive" and assign those to users.

Impact on Product architecture can be major

- Custom cake vs custom pizza;
- "Full-custom" IC vs custom ASIC

B. Develop the tools users need to carry out the need-intensive tasks assigned to them.

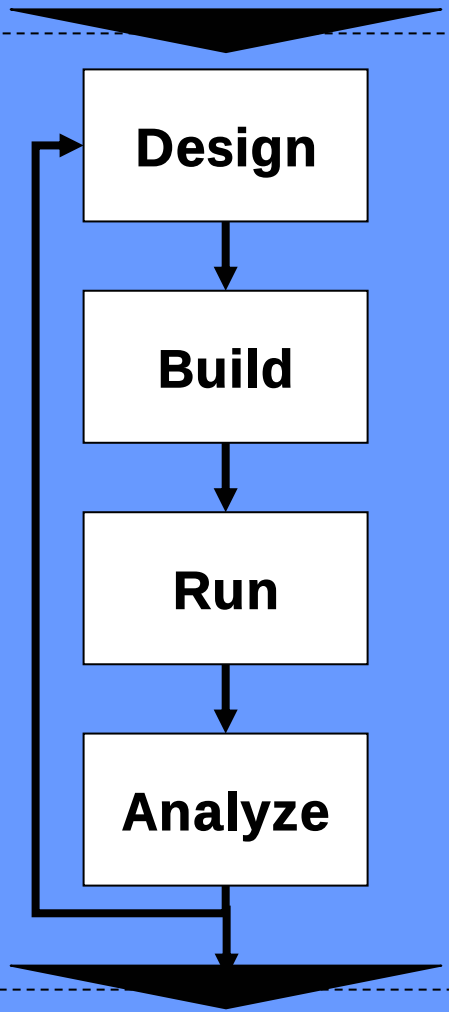


(B) Toolkits for users contain:

Tools to carry out trial-and-error design:

1. That are “user-friendly”
2. That offer the right “solution space”
3. That offer libraries of pre-designed modules
4. That can translate from user-language to producer language without error

Toolkits should help users to do the trial-and-error work of problem-solving in design

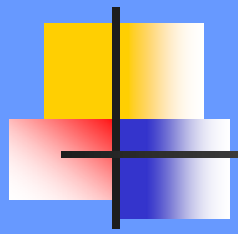


- Design a possible solution
- Develop models prototypes
- Test model/prototype In real or simulated use environment
- Analyze findings previous step

Tools to enable user to carry out design by trial-and-error

Four steps in trial-and-error-process:

	ASICs example
Design	Design custom circuit
Build	Create functioning prototype
Test	Take prototype for a “test drive”
Analyze	Compare expected and actual results. If needed, do trial-and-error cycle again. (“Iterate”)



(1) Offer “user-friendly” tools

“User-friendly” means that the user does not have to learn a new design language.

Examples:

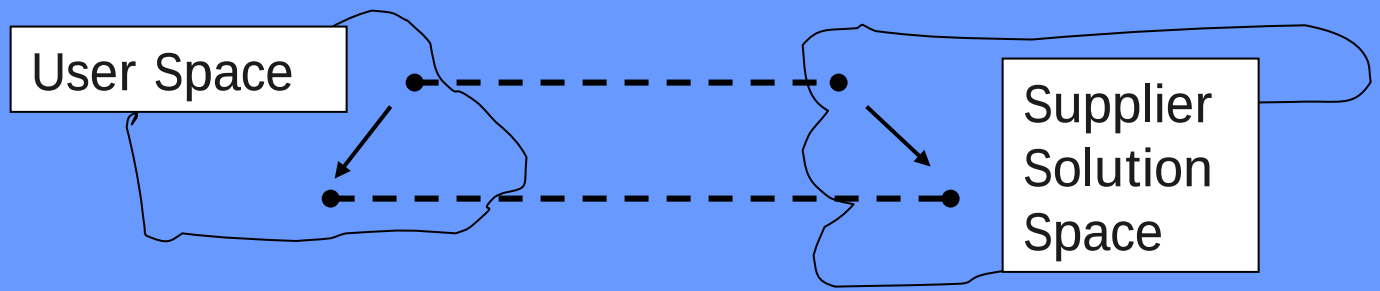
- Allow integrated circuit designers to use their customary design language: Boolean algebra
- Allow hair styling customers to use (virtual) mirror, comb and brush.

Creating user-friendly design systems

Identify the independent design dimensions that are important to the user.

Give each design dimension a familiar, functional name (e.g., “thickener” instead of xanthan gum”

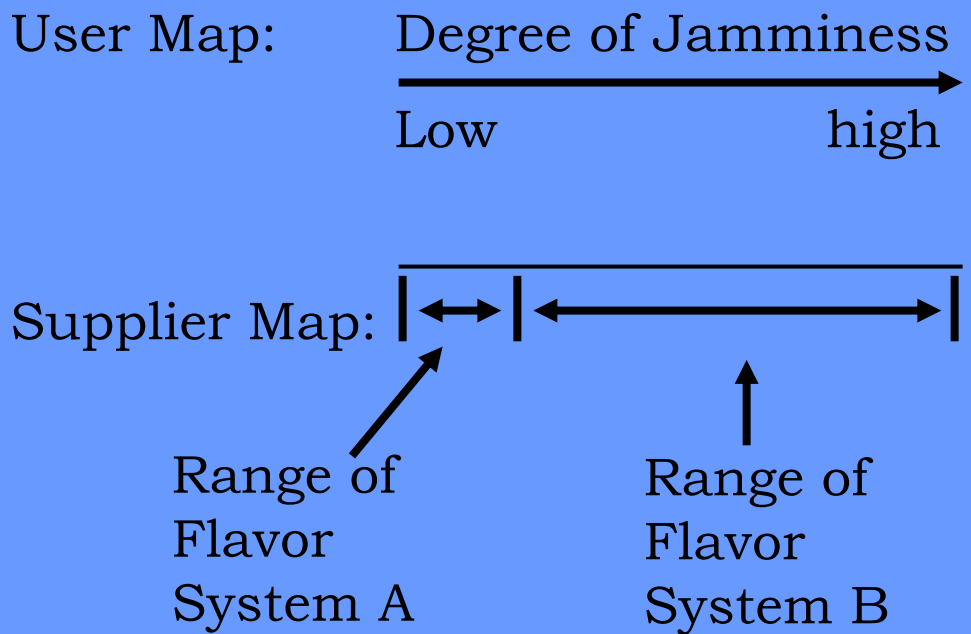
Create a translator – hidden from the user – that translates each move by a user-designer in user solution space to a move in manufacturer solution space. (Flag the user when a user move can’t be done in manufacturer solution space.)

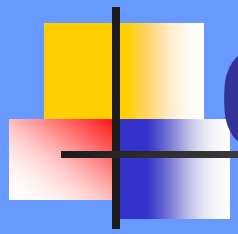


Translations can be "bumpy" – but must be error-free

Smooth movement across user solution space may involve bumpy translations on supplier map

Example: "Jammy" flavor note





(2) Offer the right “solution space”

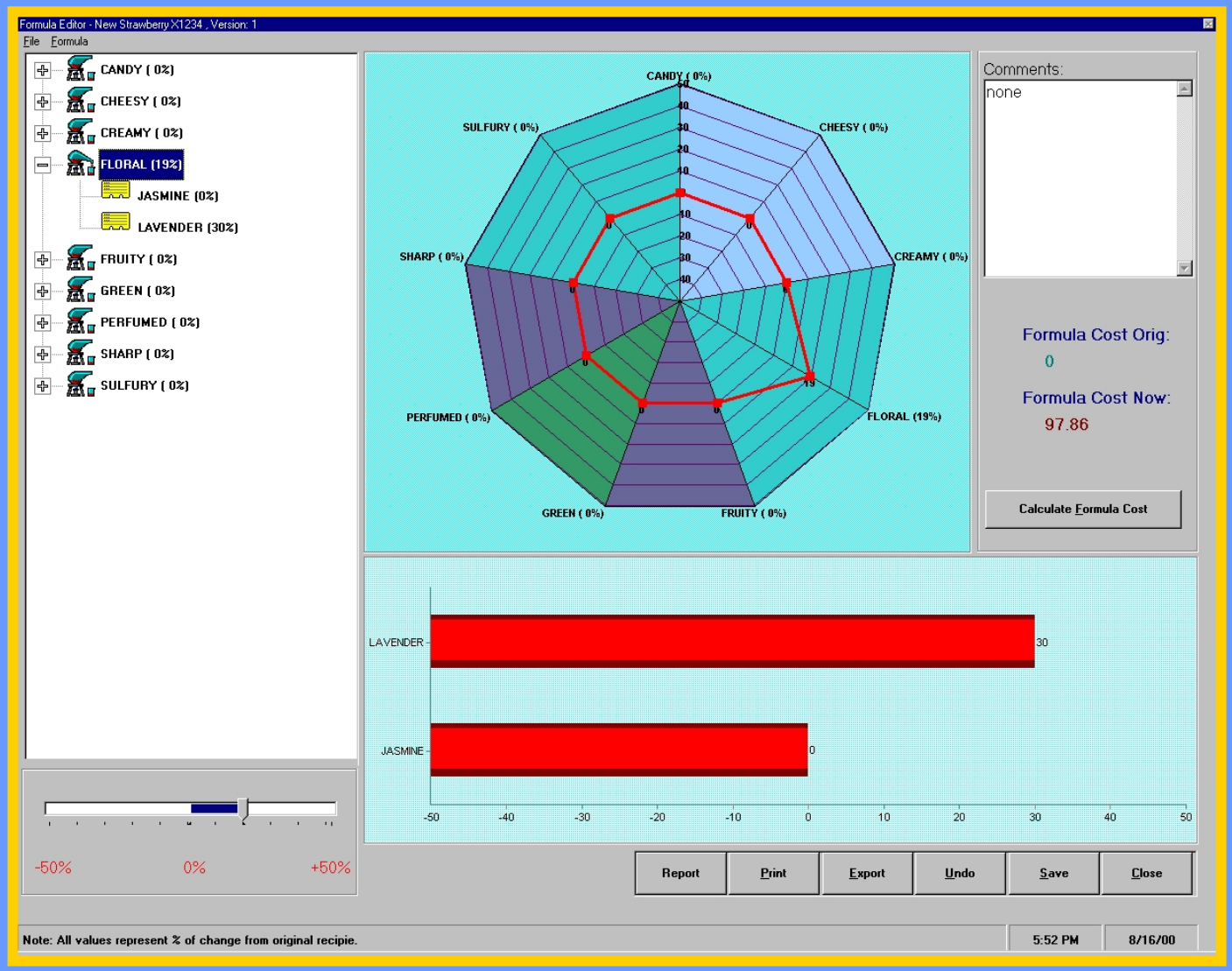
Toolkits must offer users a “solution space” that contains all the design variables and tools they need to create a design.

Example: Hairstyling toolkits:

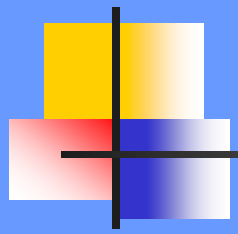
Design variables offered: hair position, length, color, waviness;

Tools offered: virtual scissors, comb, colorants, curlers, straighteners.

Flavor Design Toolbox for Users



Screenshot of Flavor Design Toolbox software courtesy of John Wright, International Flavors and Fragrances. Used with permission.



(3) Offer pre-designed modules

Custom designs are typically not totally unique. Toolkit libraries should contain pre-designed modules and modifiable "default designs" – so that users can concentrate their design work on the novel features of their designs.

Examples:

- "Macrocells" for custom IC designs: microprocessor
- Modifiable "default designs" for hairstyles or for houses.

Modules should make "design sense" to a user-designer. (e.g., not "half a roof plus front door" for house designers, or "sautéed garlic plus onions" for chefs)

(4) Toolkits must enable "first-time," error-free production of user designs

User design language provided by toolkit must translate to production language without error:

Sometimes this is easy:

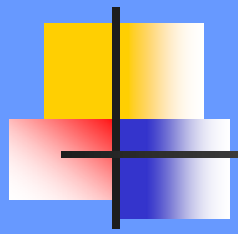
Translation from circuit design language (Boolean algebra) to IC producer's digital device fabrication language.

Sometimes this is hard:

Nestle Mexican Sauces toolkit

Creating Value with Toolkits: Experiences at GE Plastics

- 30 years of in-house expertise on website (tools): $5 mill. cost
- Potential customers can solve their own design problems
  - Helpline calls dropped >50%
  - 400 e-seminar for 8,000 potential customers per year
- About one million visitors p.a.
  - Automatic screening and tracking of potential customers
  - One third of new customers
- Sales threshold dropped by more than 60%



Profiting from toolkits

Users will benefit from toolkits in your industry if user needs are heterogeneous.

If users will benefit, you must offer toolkits – or someone else will and get first mover advantage.

Your business model may change when you offer toolkits – for better or for worse.

Example: ASIC foundries profited from a toolkit approach for the first 15 years – and then began to lose profit to specialist toolkit suppliers.

How to start developing a toolkit

- It's OK to start with something rough as long as it offers sufficient value to entice user experimentation. Simple release of in-house design tools is sometimes a sufficient for a start.
- Work with lead customers that **really** need your toolkit and so will be willing to work with you as you refine it.
- You don't need superhuman insight to design and update toolkits – lead users will bump up against the edges of the solution space your toolkit offers and ask for more – or design toolkit improvements for themselves.