

# **Effective Design Principles**

# Overview

A principle is a fundamental truth or proposition that serves as the foundation for a system of belief or for a chain of reasoning.

— New Oxford American Dictionary, 2010

Principles support alignment and decision-making, and to a lesser extent creativity. However, they are often so broad and consensual that they are de facto ineffective.

This document tracks best practices for eliciting and writing useful design principles.

## Best Practices

Good design principles are memorable, evolving, meaningful, outbound, specific, prioritized, applicable, determinative and encompassing (mnemonic: MEMOSPADE).

### Memorable

Clear, concise and impactful so that they can be internalized, thus generative, rather than referred to, thus conformative. Use short, unambiguous, and well-illustrated statements.

For example, Android's "make important things fast".

### Evolving

Reasonably iterated upon, to remain relevant as the product grows. Adjust their substance and scope, especially in earlier phases when things are still in flux.

For example, Gravity replaced "be universal" with "inclusive".

### Meaningful

Imparting a human-oriented sense of why, so that they're memorable and motivating. Make the rationale interesting to the reader (designers or non-designers) rather than just true to the system.

For example, "small things matter, good and bad" from Windows UX.

### Outbound

Expressed in terms of end-user benefits rather than internal benefits. Don't mention tools, technical capabilities or other similar internal properties.

For example, replace "use push notifications" with "keep the user apprised timely".

## **Specific**

Not a truism, indistinguishable from competitors, that doesn't help making decisions. Replace "always true" statements with opinionated visual preferences, focused guidelines or business idiosyncrasies.

For example, "clarity above all" rather than "easy to use".

## **Prioritized**

Ranked against one another, so as to provide a path to resolution for difficult use cases. Write principles as "x over y over z", or make an explicit different between must have and nice to have.

For example, as per Salesforce: "clarity > efficiency > consistency > beauty".

## **Applicable**

Easily connected to designs and acted upon. If a statement isn't actionable it's a goal, not a principle; principles must support design activities. Make sure principles aren't completely abstract, demanding a lot of thinking to be applied to the product. Provide an high-level "how".

For example, "use shape to communicate actionability" rather than "make controls obvious".

## **Determinative**

Allowing to say no, or to fairly pick between options. Otherwise the effort devolves into lowest common denominator without style nor conceptual integrity. Break down big statements into smaller, unambiguous ones. Write down explicitly their known consequences, positive or negative. Quantify if possible.

For example, "icons must have a single meaning over" rather than "meaningful icons".

## **Encompassing**

Applicable to entire classes of problems, now and in the future. Otherwise they're a pass/fail criterium, not a principle. Avoid being too detailed, for example writing about one use case or component in particular.

For example, "start with smart defaults" rather than "placeholder text in input fields".

# **Furthermore**

- Principles can come from external factors like business strategy, market constraints, technical limitations; or internal factors like brand or personal preferences
- Principles can inform entire strategies, new products, big redesigns or specific journeys. On smaller efforts like feature-add or limited redesign, they can be too much overhead. It's preferable to use guidelines then.
- Having more than a handful principles is overwhelming; curate actively
- More literature in [Reading notes on design principles](#)