2765 The Product That Grew Past Its Design

Product Book · https://kb.jeffreysummers.com/docs/2765-the-product-that-grew-past-its-design/

Every product starts clean. The menu has a logic. The sequence of service has a shape. The station layout has a reason. Then the operation runs into reality — a Guest request that became a permanent modification, a supply issue that became a permanent substitute, a service step added to fix a complaint that was never removed when the complaint stopped. Each addition was a response to a real problem. The accumulation is [Complexity Decline].

The product that reaches [Complexity Decline] does not look broken from the inside. It looks like a fully-featured operation — a menu with options for every preference, a service sequence with steps for every contingency, a station layout adapted to every workflow that ever ran through it. From the outside, from the Guest's side of the experience, it looks like a system that has lost its shape. Too many options. Too many steps. Too many hand-signals between cast members that were never written down because they started as a one-night workaround and became permanent without anyone deciding they should be.

The alternative position is that product evolution is product improvement — that every addition is a response to feedback and feedback-driven changes make the product better. That position is true for individual additions and false for accumulation. The tenth response to feedback does not improve the product. It complexifies it. The Guest who drove the feedback that produced the tenth addition does not experience a more responsive operation. They experience a more complicated one.

Product design under [Complexity Decline] requires a subtraction discipline alongside the addition discipline. Every new element earns its place by answering one question: does this close the [Trust Arc] or does it add complexity to a system that is already producing the experience the Guest came for? If the answer is addition without arc-closure, the element does not belong. The product that stays clean is not the one that never responds to feedback. It is the one that subtracts as deliberately as it adds.

What changes tomorrow: count the number of modifications your kitchen executes on a standard shift. Not the Guest-requested ones — the permanent ones that started as accommodations and

never went back. Each one is a complexity layer the kitchen is carrying every shift. Audit one. Decide if it still earns its place. The product gets cleaner one deliberate subtraction at a time.