The Complexity Matrix

Complexity doesn't get resolved, it travels. Who absorbs it, what it costs them, and where it should go instead.

01 · THE PRINCIPLE: WHERE THE IDEA COMES FROM, AND WHAT SIRAG ADDS

ORIGIN · LITERATURE
"Simplicity is a secret complexity."
Jorge Luis Borges, as quoted by Sirag

→

PARALLEL · SOFTWARE, MID-1980s
Tesler's Law: complexity can't be removed. Developer or user carries it.
Larry Tesler, Xerox PARC · context, not from the episode

→

SIRAG'S VERSION
Complexity doesn't get resolved. It travels elsewhere.
Five possible 'payers' instead of two · allocate on purpose

02 · THE MATRIX: WHO ABSORBS IT, AND WHAT THEY PAY

BEYOND "TEAM VS. USER": FIVE POSSIBLE 'PAYERS'

	3rd Party / Vendor e.g. tech enabler · streamers	Platform Team shared services	Feature Team the product team/squad shipping it	User	Company (ROI) long-term health of the business
FROM THE EPISODE · TELCO SIGN-UP Does the user's phone support eSIM? one decision · several possible owners	**OPTION A** Tech partner builds the compatibility check (API)	not covered in the episode	**OPTION A (w/ 3rd Party)** Integrates the check into the sign-up flow	**OPTION B** Checks it alone: settings, specs, Google, ICCID codes ROI: sign-up drop-off	**OPTION C · NOBODY OWNS IT** High-volume support calls, or P&L impact
KINGS LEAGUE Traffic acquisition the price can arrive later	**SOLVED HERE** Streamers bring their audience: acquisition "easy"	**the complexity moves** →			**PAID HERE, LATER** Lower defensibility: copycats can contract the same streamers
ILLUSTRATIVE EXAMPLE Address entry at checkout not from the episode · for your own work	**OPTION A** Address-autocomplete API, integrated by the feature team ROI: Licence fee + vendor dependency	**OPTION B** Shared validation service, built once ROI: Upfront build, reused by every team	**OPTION C** Builds its own inline validation ROI: Fast for one team, duplicated across many	**OPTION D** Types and checks the full address alone ROI: Friction → failed deliveries, returns, support	**RESULT** Where every option lands Better or worse depending on A–D. No option is free.

03 · HOW TO RUN IT ON YOUR OWN WORK

01 **Pick one thing** — a feature, flow or process
02 **Map who pays what** — hours, fees, effort, cost, tickets
03 **Ask who can stop** — and who should carry it
04 **Move it on purpose** — towards long-term ROI
05 **Revisit, iterate** — improve as you learn

Sirag's recommendation: run it on any feature, process, onboarding or action.

READING THE MATRIX: WATCH-OUTS

Suspect every "net positive".	If it looks free, find who absorbs it.
Easy now can cost later.	A shortcut can turn into a dependency.
Money helps, risk resists a formula.	Sometimes, the final call is "not an Excel formula".