



Capable teams, each playing its own tune? One book gets everyone in time.

Architecture and governance means one shared map of which software does what and how it connects, plus a few agreed house rules that are checked while the work happens, not the night before launch.

One book, everyone in time.

The story, decoded. In our story, the orchestra is your teams and the software each one runs, the music book is one shared map of your software, the shared beat is the few choices everyone agrees, and the listening gadget is an automatic check.

SOUNDS FAMILIAR?

Every team picked its own tools, and now nothing quite fits together.

We pay for several tools that do the same job.

Nobody can draw how our systems connect, or the drawing is years old.

Our house rules live in a long document, and every reviewer reads it differently.

THE IDEA IN PLAIN WORDS

One shared map of which software does what

One current map of your software, who looks after each piece and how they connect. Each team zooms into its part, and new projects plan to fit before they start.

Experts call this: enterprise architecture, solution architecture

A few agreed choices, and the reason written down

Everyone agrees a short list of choices, such as which tools to use for common jobs and where customer details are kept. Each group decides its small things and writes down why.

Experts call this: standards, principles, architecture decision records

House rules checked as you go

The rules that matter most are written so a computer can check them each time someone changes the software. A bit off: a hint. Way off: a person decides. A few big rules simply stop it. You set, rule by rule, which of the three a slip gets.

Experts call this: policy as code, guardrails, continuous compliance

Three questions for anything new

Before a new tool or supplier joins: does it fit, what does it replace, what if it breaks? The pieces you already rely on get a plan B too.

Experts call this: technology governance, technology risk

Worth knowing

It's a journey, not a quick fix: we start with one map and one rule, and build from there with your teams.

Worth knowing

Teams can make a call after asking the people the decision affects; asking is not the same as everyone agreeing. Writing down why means fewer old arguments come back, not none.

Worth knowing

Automatic checks catch many problems, not all of them. Only rules written so a computer can read them can be checked this way, and people still make the hard calls. Most rules start as hints; only a few become a firm stop.

HOW WE WORK WITH YOU

STEP 01

Listen

Sit in, talk to the people who build, and list where things clash or wait.

STEP 02

Map it with you

One map of what you run and how it connects, drawn with the people who know it.

STEP 03

Agree the rules

Pick the rules that matter most, and how strict each one is.

STEP 04

Set up checks

Turn the top rules into automatic checks with one pilot team.

STEP 05

Coach as you go

Coach your people, and hand over the map, rules, notes and checks.

Start small: one map of your software, one house rule checked automatically with one team, or one stuck decision settled in writing.

WHAT WE HELP WITH

One map of which software does what

A shared, current map your teams can read and keep up to date.

A few house rules that matter

A short list people can find and follow.

Worth knowing: Laws and regulators' rules are not optional; "few and clear" is about your own house rules.

New projects planned to fit

A new system designed to fit what is already there.

Three questions for anything new, and a plan B

A clear view of what comes in, why, and which pieces are risky.

Everyone in time yet?

Start with one map: email contact@algoshred.com with what your teams run and where things clash, and we'll suggest the first corner to map.

Vignesh T.V., CEO & CTO, Algoshred Technologies Pvt. Ltd.

WATCH THE FILM

algoshred.com/architecture-governance