

VOLUME I

RE-ARCHITECTING GOVERNMENT

DR. SAJJAD BABAMALEKI

REDESIGNING ADMINISTRATIVE DECISIONS

Re-architecting Government

Volume I

VOLUME I

Re-architecting Government

Redesigning Administrative Decisions

A Method for the Digital Transformation of Government

Sajjad Babamaleki

Reference Edition

Copyright © 2026 Sajjad Babamaleki. All rights reserved.

Translated from the Persian by the author's arrangement.

Published by GOVOG, Dubai.

Typeset in TeX Gyre Pagella and TeX Gyre Heros.

Contents

About This Book	1
Chapter 1. The Public Problem and the Starting Point	4
Chapter 2. Antecedent Methods and the Place of the Book's Proposal	10
Chapter 3. The Administrative Decision as the Unit of Redesign	24
Chapter 4. Process Discovery and the Measurement of Administrative Burden	29
Chapter 5. Law, Authority and the Implementability of Regulation	37
Chapter 6. Building the Redesign Decision File	43
Chapter 7. Comparing Interventions and Correctable Choice	54
Chapter 8. Risk, Exceptions, Appeals, and Redress	62
Chapter 9. The Whole Service and Inter-Agency Responsibility	71
Chapter 10. Data Architecture, Rules, Artificial Intelligence, and Resilience	82
Chapter 11. Power, Staff, and Implementation Capacity	91
Chapter 12. Financing, Procurement, and the Transition to Operation	102
Chapter 13. Evaluating Outcomes and Testing the Method	116
Chapter 14. Real Cases and a Comparison Program	135
Chapter 15. Transferring the Method and Differences of Context	162

Chapter 16. The Correctable State and a Research Program	171
Closing Proposal: A One-Page Decision Change Standard	180
Appendix A. The Sepidar Teaching Case	191
Appendix B. Working Tools and Running the Cycle	200
Appendix C. Glossary	210
References and Limits of Use	213
Subject Index	228
Index of Names and Institutions	230

About This Book

This book begins the redesign of government from a unit that most digital transformation programs never see: one specific change in an administrative decision. Instead of “making the system smart” or “integrating the service,” the book asks a different question. When the source of verification, a referral condition, a decision rule or responsibility for implementation changes, on what authority, on what evidence, at what cost to whom and with what means of correction is that change accepted? The book’s answer is an operational method, “the Administrative Decision Re-architecture Method,” which turns every change into a versioned, reviewable file.

The text contains three kinds of content: explanations of concepts and evidence with precise references, the author’s method and prescriptive rules, and teaching examples. The city of “Sepidar” and the other teaching examples use fictitious rules and figures and are designed to show the logic of the method. The real cases in Chapter 14 are reconstructed from official documents, audits, court rulings and published empirical studies.

A Guide to the Status of Claims

The book distinguishes three kinds of statement and explains the distinction once, here, so that the main text can proceed without interruption and in a prescriptive voice. A “finding” is a statement quoted from a cited source, and its scope is the scope of that source. A “method rule” is this book’s prescription: what a redesign team should record, compare or stop. An “example” is a situation constructed for teaching, and its figures are not field data.

The method rules rest on three kinds of evidence: evidence of the problem (errors that have occurred in real cases), evidence of mechanism (why a link or a control has an effect) and evidence of intervention (the results of similar reforms in experiments and evaluations). A fourth level, a direct comparison of the method’s own performance against professional alternatives, is a research program whose implementation plan Chapters 13 and 16 set out.

Wherever a figure is hypothetical or a source has limited scope, the text says so once, at that point.

A Guide to Reading

Chapters 1 to 3 define the problem and the unit of analysis. Chapters 4 to 7 move from understanding to the decision file and the comparison of options. Chapters 8 to 12 explain the conditions for responsible implementation. Chapters 13 to 16 are devoted to evaluation, historical critique, transfer and the research program. An executive reader can follow the full Sepidar case in Appendix A alongside Chapters 6 and 7 and put the tools in Appendix B to use. The book ends with a closing proposal: the “Decision Change Sheet,” a one-page standard that is the floor beneath all of the book’s tools and is proposed for every change to a decision that affects people.

Each chapter opens with an “At a Glance” table that states, in four rows, the problem, the main lesson of the cases, the chapter’s tool and the most common point of failure. Each closes with an “Applying It in Your Unit” table: three actions that the manager of a unit can take within the limits of existing authority, without a new law, budget or permit, each backed by the chapter itself. A manager with limited time should follow this ninety-minute path:

Reading path for the executive manager

Step	What to read	Where	Time
1	The “At a Glance” table of all sixteen chapters, to find the chapters relevant to your unit’s work	Start of each chapter	25 minutes
2	The Sepidar case, which shows the method applied to a hypothetical change from start to implementation	Appendix A	25 minutes
3	The “Applying It in Your Unit” table of selected chapters	End of chapter	15 minutes
4	The Decision Change Sheet and its three stop points	Closing Proposal, Closing Table 1	15 minutes
5	The architecture of the tools and Tool 8, decision gates	Appendix B	10 minutes

References are numbered and appear at the end of the book. A number after a reference, when accompanied by the word page or paragraph,

indicates the location of the material within that source. The definitions in the glossary are this book's working definitions and do not replace the specialist or legal definitions of any field.

Chapter 1. The Public Problem and the Starting Point

Chapter 1 at a Glance

Topic	Summary
Problem	The unit launches the project under the name of a solution, such as “we don’t have a system,” and expands the scope without limit before it has set the level of intervention, the service boundary and a measurable outcome.
Lesson of the cases	The UK “Service Manual” treats the user’s whole journey as the service, but seeing the journey is not the same as owning all of its consequences; reforming business permits shortens the time to start operating, while employment depends on other factors.
Tool	The “problem charter,” with two separate boundaries, the service boundary and the attribution boundary, makes clear who faces what barrier, what lies outside the scope and who measures the outcome.
Common point of failure	A high savings score in a weighted sum neutralizes the absence of an appeal route or an uncontrolled severe risk, or the project absorbs every problem and neither its outcome nor its failure can be explained.

Administrative Output and Public Outcome

Registering an application is an output; receiving the service on time is an outcome closer to the public need. Issuing a permit is not always the end of the problem either: if the electricity connection or the safety approval does not arrive, starting operations remains impossible. The boundary of the analysis should be set by an outcome for which the government can reasonably accept responsibility, not by the last button in one organization’s system.

At the same time, attributing every social outcome to a single process is a mistake. Reforming business permits reduces the time to start operating, but employment also depends on demand, capital and skills. So write out the chain of effect: what behavior or capacity does the intervention change,

and by what route does that change reach the public outcome? Every link in this chain is a testable assumption.

The Service Boundary Is Not the Attribution Boundary

The UK Service Manual defines a service as everything the government collectively provides to help all users reach an outcome, and it distinguishes three levels of outcome, from a process step to a broader outcome [46, section What a service is]. This definition breaks organizational boundaries, but it does not establish a causal relationship: seeing the user's whole journey is not the same as owning all of that journey's consequences.

Write two separate boundaries in the problem charter. The service boundary shows which components, channels and organizations the user deals with to reach the outcome. The attribution boundary shows by what route, and on what assumptions, the intervention under review can explain part of the change in the outcome. The first prevents the experience from being fragmented; the second prevents the project from claiming consequences that other factors produced.

If the two boundaries differ, do not erase the difference. A project that changes only the stage at which a decision is announced must, in order to measure the real time to access, also see the data from the following stage. This dependency is not a reason to expand the scope of implementation automatically; it is a measurement and coordination need whose owner and data source must be specified.

Four Levels of Intervention

Digitizing information turns paper into data. Digitizing the workflow changes how work is transferred and tracked. Process redesign reviews whether steps are necessary and how decisions are made. Institutional reform changes authority, accountability, budgets and organizational rules. These levels sometimes occur at the same time, but they should not be reported in place of one another.

The OECD framework describes digital government through components such as being data-driven, user-driven and taking a platform approach [20]. A practical rule: the project's name is chosen after the level of the problem has been diagnosed. If the organization's mission is unclear, buying software is not the answer; if the only problem is re-entering information, rewriting the entire law is disproportionate.

The Problem Charter

Before mapping the process, write one paragraph: who, in what situation, faces what barrier; what evidence do we have that the barrier exists; and what is the desired outcome? The phrase “we don’t have a system” confines the problem to one solution in advance. The phrase “the applicant cannot understand the status of their case and has to come back repeatedly to follow it up” leaves room to compare several responses.

The output of this section is a problem charter with a clear boundary: what is not within the project’s scope, who measures the outcome and what conditions trigger a review of the scope. A limited scope is a tool for identifying responsibility and effect. A project that absorbs every problem of government can neither attribute its outcome to itself nor explain the failure of its components.

Why One Template Is Not Enough

Processing an application for a benefit, inspecting a hazardous facility and preparing a budget are all government processes, but they do not share the same logic. In the first, the error of denying an eligible person stands out. In the second, overlooking a risk to third parties matters. In the third, the issue is allocating resources among competing goals. A model suited to high repeatability, if transferred to another process, eliminates necessary judgment.

For the purposes of understanding, we separate five families of process: policymaking and rulemaking; regulation and supervision; determining eligibility and delivering services; back-office operations; and handling complaints and resolving disputes. This classification is a tool for organizing analysis, and a single service can span several families.

Decisive Dimensions

Describe the process in terms of volume, variety of cases, reversibility of the decision, severity of the consequences of error, degree of judgment, number of agencies involved and data quality. A “simple case” must also be defined. Simplicity for the software does not necessarily mean simplicity for the applicant. A person without a standard document may have clear eligibility and yet be classified as complex by the system.

In a high-frequency, low-risk process, investing in a shared component makes sense. In a low-frequency, high-consequence decision, documenting judgment and independent review take priority. In an inter-agency process,

the main problem is often the transfer of responsibility, even if each unit acts quickly on its own.

The Product of Analysis

Build a profile for each process: its family, the expected outcome, the affected population, the legal owner, the implementer, the delivery channels, its dependencies and its main risks. Then record where there is disagreement. If the legal unit regards it as a discretionary decision and the technology unit as a fixed rule, this disagreement must be resolved, or explicitly managed, before design.

In this method, a ministry is not assumed to be equivalent to a process. Procuring medicines flows among the ministry, hospitals, the treasury and suppliers. The unit of redesign is the chain that produces the defined outcome. Organizational structure is one of the constraints on that chain; the choice of the boundary of analysis must be justified separately.

The Minimum Diagnosis Before Choosing a Project

Before scoring, run three elimination tests: Has the problem been defined by an observation that goes beyond general dissatisfaction? Does the change fall within the scope of authority or a valid referral route? And has at least one observable outcome and one high-consequence harm been named? A negative answer does not reject the project; it sends it back from the solution-selection stage to the stage of research or of resolving the question of authority.

This threshold must remain light. Its purpose is not to build a complete file before starting; it is to prevent the selection of a project whose problem is solution-driven, whose authority is unknown and whose outcome cannot be measured. If this preliminary check alone takes weeks for a low-risk change, the method has created new administrative burden before creating any value.

High Value Is Not Always the Right Place to Start

A large process has great potential benefit and at the same time creates many unresolved dependencies. A small process is suitable for learning but is sometimes not representative of the main problems. The first choice should weigh a combination of public value, feasibility and learning value.

First examine the irreversible constraints: legal authority, a minimum budget, the ability to protect people and the existence of someone responsible. Then compare the permissible options. A high savings score

must not, in a weighted sum, neutralize the absence of an appeal route or an uncontrolled severe risk. Some criteria are conditions of entry, not tradable scores.

Multi-Criteria Decisions Without False Precision

For the remaining options, record the range of benefit, cost, time, risk and uncertainty. If the weights of the criteria change, does the ranking of the options change? An option that wins only under one favorable set of weights is a fragile choice. Differences in the values of decision-makers should not be hidden behind a decimal number.

Look at dependencies separately from the score. Fixing a base register sometimes shows little direct benefit but makes several services possible. Conversely, infrastructure for which no specific service is ready to use it builds idle capacity for years. Plan infrastructure and services together on the basis of evidence of shared use.

The Learning Portfolio

A reasonable portfolio includes a quick low-risk fix, a limited inter-agency problem and an infrastructure prerequisite. Their number is determined by the team’s real capacity. Running ten experiments simultaneously without the capacity to evaluate them does not create ten sources of learning; it creates ten ambiguous projects.

For each stage, set an output and a decision gate. After discovery, “do not build a system” is a justified decision. After an experiment, a simple fix is sometimes preferable to a complex architecture. A reasoned stop must have a clear place in the budgeting and reporting mechanism; without one, the incentive to continue the project and to interpret the evidence optimistically grows stronger.

Applying It in Your Unit: Chapter 1

Action within the unit's authority	Backing in this chapter
Write a one-paragraph “problem charter” for the unit's main problem: who, in what situation, faces what barrier, what evidence you have, what the desired outcome is and what lies outside the project's scope.	Section “The Problem Charter”: the solution-driven phrase “we don't have a system” confines the problem to one solution in advance.

Action within the unit's authority	Backing in this chapter
At the next project-selection meeting, ask the three elimination tests: the problem is defined by observation, the change is within the scope of authority, and one observable outcome and one high-consequence harm are named; send a negative answer back to the research stage.	Section ``The Minimum Diagnosis Before Choosing a Project": a negative answer does not reject the project; it sends it back to research or to resolving the question of authority.
Before choosing among the remaining options, recalculate their ranking with two different sets of weights, and record in the minutes any option that wins only under one set of weights as a fragile choice.	Section ``Multi-Criteria Decisions Without False Precision": an option that wins only under one favorable set of weights is a fragile choice.

Chapter 2. Antecedent Methods and the Place of the Book’s Proposal

Chapter 2 at a Glance

Topic	Summary
Problem	Every specialism involved in reforming a service, from experience design to law, operations and technology, sees a real part of the problem, and one specialism's solution is taken to be everyone's solution without its assumptions being made explicit.
Lesson of the cases	In “Sepidar,” the data completeness report confirms that an address record exists, but the data definition document shows that some of the records are correspondence addresses; this contrary evidence changes both the scope of the recommendation and the economic calculation.
Tool	The “reviewable link,” with six components, namely the change identifier, the statement relied on, the source and its range of validity, the connecting assumption, the person responsible for review and the review event, creates a reviewable trail for every important recommendation.
Common point of failure	Contrary evidence is added to an annex without the recommendation changing, or the fact that one route is faster in the event data is treated as permission to choose that route without the added assumption being recorded.

One Problem, Several Languages

In a meeting to reform a service, the experience designer talks about repeat visits, the legal officer about the basis of verification, the operations manager about the case queue and the technology lead about connecting systems. Each sees a real part of the problem. The difficulty begins when one of them’s solution is treated as everyone’s solution without its assumptions being made explicit. Removing the document upload is an improvement

for the experience designer; for the person responsible for the decision, it is a change in the source of proof. Both descriptions are true at the same time.

To create a shared language, the existing literature is the starting point; user-centeredness or decision recording is not reinvented. The central question is more precise: for a specific change in an administrative decision, what information and what reasoning must be placed side by side so that the change can be accepted, rejected or deferred? The answer must be usable both by the decision-maker and by whoever later reviews the decision.

Service Design: Setting the Boundary of Experience

The second point of the UK Service Standard emphasizes solving the user's whole problem, collaboration among organizations and choosing a solution proportionate to the need. The same guidance warns that solving a whole problem does not mean building an oversized service or fixing everything in one go [1, sections Why it's important and What it means].

This antecedent curbs a common error: the boundary of an office is not the boundary of the applicant's problem. In the Sepidar teaching case, obtaining a permit is not the same as starting economic activity; but the project is not meant to resolve every barrier to starting activity either. Its operational boundary runs from the registration of the application to the announcement of a reasoned decision, and dependencies outside this boundary are recorded separately.

At this point the book's method adds one requirement to the change register: the project lead writes down which part of the experience the reform changes and about which outcome it makes no claim. This requirement turns a well-known principle of service design into a reviewable commitment.

Decision Modeling: Making Dependencies Visible

The DMN standard is designed for the precise description of decisions and business rules, and it covers the link between decision design and its execution and its use alongside process modeling [2]. So "see the decision separately from the workflow" and "record the decision's dependencies" come from DMN, and this book puts them to use.

In a simple example, the decision to issue a permit depends on verifying the premises, verifying the type of activity and completing the relevant conditions, and the decision model displays this dependency. The redesign file adds a question about the change itself: if the source used to verify the premises changes, why is that change permissible and reliable, who bears its cost and under what conditions must it be reviewed? DMN describes the

logic of the decision; the redesign file describes the reason for changing that logic.

The real competitor of the book's method is not a weak diagram; it is a team that uses a decision model well, together with legal review and an evaluation log. Chapter 13 shows how a comparison with such a team is designed.

Machine-Consumable Rules: Bringing Rule and Implementation Closer

The report "Cracking the Code" defines "Rules as Code" as building a machine-consumable version of some government rules alongside the natural-language text, and it links this to the relationship between drafting and implementing rules [3, Overview]. This discussion exposes the risk of scattered and inconsistent translations of a rule.

In the decision file, three things are recorded separately: the reference text, the interpretation in use and the implementation logic. Suppose the text requires verification of the place of activity; the internal instruction demands a paper certificate; and the system treats a missing file as equivalent to the condition not having been verified. Before automation, it must be made clear which part is a requirement of the reference text and which is an implementation choice. Fixing the form does not resolve a dispute over interpretation.

Administrative Burden: A Question of How Costs Are Distributed

Moynihan, Herd and Harvey explain administrative burden through three costs, learning, psychological and compliance costs, and show that the distribution of this burden is the product of political choice [4]. So a lower cost for the organization cannot be taken, without examination, to mean a lower burden for the public.

The book's method turns this discussion into an operational duty: for every option, the cost to the government, the cost to the applicant and the non-monetary effects are recorded separately. If automatic recording of information reduces staff contact but makes it harder for the applicant to correct an address error, both changes are visible. Do not convert anxiety or exclusion into money without a clear basis; report them with a description, an appropriate measure and a range of uncertainty.

Learning and Evaluation: Two Separate Questions

PDIA presents problem-solving as an iterative process of breaking down the problem, finding an entry point, acting and learning [5]. The decision

file plays the role of memory in this cycle: what we thought, what we did and which assumption was corrected. Recording a form does not by itself build institutional capacity or the authority to change; it builds a memory on which that capacity is built.

The Magenta Book treats a theory of change as a means of explaining the path of effect, the assumptions and the evidence, and it separates the additional effect of an intervention from merely observing an outcome [6, section 2.2.1]. For this reason, two separate evaluations are needed: did the service reform produce a better outcome; and did the method used to choose the reform produce a better decision than an alternative method? The success of a project does not answer the second question.

Professional Discretion and Its Transfer to the System

Lipsky showed that frontline workers, under the pressure of high caseloads, ambiguous goals and limited resources, build routines that open a gap between formal policy and day-to-day implementation [77]. A design rule: examine variation in how cases are handled, and do not call differences error or waste before examining them.

Bovens and Zouridis analyzed the transition from “street-level bureaucracy” to “system-level bureaucracy”: with digitization, software designers and system analysts become the real holders of discretion, while the question of fairness and of handling hard cases remains [78]. Zouridis, van Eck and Bovens later extended the same analysis to the era of automated decision-making: discretion has passed from caseworkers to programmers and data analysts, while the mechanisms of democratic oversight over this group have not yet been strengthened to the same degree [87]. Discretion does not disappear with digitization; it moves and must again be made subject to accountability.

The book's method therefore records where discretion lies before and after the change: which judgment is assigned to the caseworker, the supervisor, the rule designer or the data owner; who is authorized to grant an exception; and how a change to a rule is approved and reviewed.

The Claim This Book Takes On

The book's claim is this: systematically connecting change, authority, evidence, options, the distribution of burden and correctability reveals decision errors before implementation. Its mechanism is clear: opaque dependencies become visible and the team is forced to compare options in earnest. The cases in Chapter 14, from Robodebt to the Dutch childcare benefits scandal, broke down precisely at these connections.

The method also takes its own cost seriously. If the same problems are revealed by a shorter method, if filling in fields becomes a formality, or if recording delays a clear, low-risk correction for no reason, the method must be shortened. That is why three levels, short, standard and full, are part of the method itself.

Reengineering Before Digital Government

GAO's 1997 guide to assessing reengineering placed the decision to reengineer, the development of a new process and the assessment of implementation and results within a single framework; it included examining practical options, the business case and change management, and it rejected checklist use [22, contents and printed page 12]. Reengineering should not be reduced to removing steps and cutting costs.

This book takes a smaller and more precise scale of analysis: when a source of verification, a referral condition or a decision rule changes, how should the link among the components of its justification be preserved? Reengineering looks at the program; decision re-architecture looks at each change within the program.

Process Mining and the Gap Between Observation and Justification

The IEEE Task Force defines the scope of process mining as going beyond path discovery: conformance checking, organizational analysis, case prediction and building simulation models also belong to it [18]. Process mining is a powerful tool for seeing actual execution.

But observing that one route was faster is not permission to choose it. In the Sepidar example, event data show that applications with a registered address are resolved sooner; this does not yet say that the groups were comparable to begin with or that the registration is sufficient to verify the place of activity. The book's method records this gap: what claim the evidence supports and what assumption has been added to move from observation to recommendation. This question complements data analysis.

Antecedent Principles That Cannot Be Called New

In introducing Rules as Code, the OECD set out principles such as accountability, traceability and contestability [3, Overview], and the UK Service Standard requires understanding legal constraints and working with policy specialists [1, What it means]. These principles are foundational, and the book builds on them.

PDIA likewise counts learning with feedback from evidence and the participation of actors from several organizations among its principles [23, PDIA Principles]. The book’s contribution is not in renaming these principles; it is in turning them into the operational file of a redesign decision.

Regulatory Impact Assessment Is Not a Marginal Competitor

The European Commission’s Better Regulation guidelines and toolbox organize problem analysis, options, impacts, implementation, monitoring and evaluation within the legislative cycle [50]. If an administrative change stems from amending a regulation, regulatory impact assessment is the first frame of reference.

The difference lies in the level and unit of analysis. Impact assessment starts from a policy initiative or regulation; the re-architecture file follows a limited change in the evidence, the processing route or the implementation responsibility of that same regulation. For a large change, the book’s file is an input to, or an annex of, the impact assessment, not a parallel track. For a small change, if an existing assurance mechanism answers the same questions, rely on it.

A Comparison Map and the Limits of Inference

The table below places the central question of each approach beside its place in the book’s method; the third column shows where each approach is applied.

Table 1 Comparison of redesign approaches

Approach and source	Prominent question in the selected source	Where it is used in the book’s proposal
Reengineering in the GAO guide [22]	Is the process change justified, and is it being implemented?	The file for a specific change within a reengineering program
Service Standard [1]	How is the user’s whole problem to be solved?	Defining the outcome and the dependencies outside the team’s authority
DMN decision modeling [2]	How are a decision and its rules to be stated precisely?	Connecting the decision model to the reason for accepting a change to it

Approach and source	Prominent question in the selected source	Where it is used in the book's proposal
Process mining [18]	What path and conformance does the recorded execution show?	Setting the limits of inference from events to recommendation
Rules as Code [3]	How is a rule to be presented for humans and machines?	Recording the link among the reference text, interpretation and implementation of the change
PDIA [23]	How is problem-solving capacity built through action and feedback?	Keeping assumptions and reasons for review within the learning cycle

A Stress Test Against the Strongest Alternatives

The table below assesses each approach in its best professional form and shows the gap that the book’s method is designed to fill.

Table 2 The strongest professional alternatives

Professional alternative	Its strong capability	When it is sufficient on its own	The gap the book's method fills
Process management and reengineering	Governance of the process life cycle, modeling with tools such as BPMN, measurement and redesign	The problem is mainly workflow, and authority and distributional effects do not remain outside the change	The reason for changing a rule or evidence gets lost among the model, the law, the cost and the review
DMN and decision modeling	Defining the decision, inputs, rules and dependencies	The decision logic is precise, and the legal justification and evaluation of the change are recorded in the existing system	There is a need to link the model version to evidence, burden, redress and review triggers
Process mining	Discovery, conformance and analysis of recorded behavior	The question is about actual execution, and the meaning of events and the quality of the log are valid	Moving from observation to recommendation requires an unrecorded legal or policy assumption
Service design	User needs, the whole journey, channels and accessibility	Professional design covers law, authority, data, the distribution of burden and outcome follow-up within the same project	Improving the experience may shift burden, risk or authority to another point

Professional alternative	Its strong capability	When it is sufficient on its own	The gap the book’s method fills
Regulatory impact assessment	Problem, options, costs and benefits at the level of policy or regulation	The change is examined at that same level and through a valid RIA process	A small implementation change in data, evidence or inter-agency handoff remains hidden from macro-level analysis [50; 76]
Rules as Code	Alignment of the rule text, the conceptual model and machine execution	The main problem is translating and publishing rules, and a complete accountability mechanism exists	The decision to change must also be connected to cost, exceptions, redress and the service outcome [3]
PDIA	Iterative problem-solving, learning and adaptation to context	The team has the authority, the decision memory and the mechanism for assuring rights and data	Learning needs a versioned file of claims, evidence, authority and stopping conditions [23]
Assurance case and evaluation	Linking claim, argument and evidence; testing mechanism and effect	The scope of the change, the cost of implementation and the route to redress are already covered within the same system	The file must be operationalized and adapted for administrative decisions and the implementation cycle [26; 44]

The result of this comparison is clear: the components of the method have antecedents, and the book’s contribution lies in combining them. This contribution has two parts: the formulation of a precise unit of analysis, namely a specific change in an administrative decision, and the requirement to preserve a reviewable trail among change, authority, evidence, options, burden, implementation, appeal, redress and review. None of the alternatives in the table requires this complete chain for an administrative decision in the form of a single file.

The Proposed Contribution at Three Levels

The book contributes at three levels. At the conceptual level, it draws an explicit distinction between a decision about an application and a decision about changing the system for processing applications. At the operational level, it turns each specific change into a traceable chain of basis, evidence, options, consequences and review conditions. At the empirical level, it offers a testable hypothesis: recording these links reveals important errors sooner at an acceptable cost; Chapter 13 sets out a design for testing it.

The precise form of the book’s contribution is this: an integrated method for justifying and reviewing specific changes in administrative

decisions, which adapts the structure of the safety-engineering “assurance case” (linking claim, argument, evidence and assumption) to administrative redesign and supplements it with authority, the distribution of burden, appeal and redress.

The Distinguishing Unit: A Reviewable Link

For the method to be more than a set of forms placed side by side, every important recommendation is turned into a “reviewable link” with six components: the change identifier; the statement relied on; its source and range of validity; the assumption connecting the statement to the decision; the person responsible for review; and the event that triggers review.

In Sepidar, the recommendation “remove the certificate for everyone” differs from the recommendation “remove the certificate only for cases with a relevant and valid registration.” The statement “an address exists” states only that a record exists; the connecting assumption is that it matches the place of activity. If the type of address is unknown, the link has not been confirmed, and the general removal of the evidence does not follow from it. The registry owner explains the data quality, and the person responsible for the decision examines whether it is sufficient for this use; neither is relieved of their own responsibility by the other’s approval.

The review event is also linked to this statement: the discovery that a batch of records were correspondence addresses triggers a review of all dependent decisions. The value of this record is that everyone can see which argument has collapsed and which decisions must be examined again.

Comparison With the Strongest Practical Alternative

The book’s method must be measured against the strongest practical alternative: a team equipped with a service map, a process or decision model, a review of authority and risk, a cost comparison of options and an evaluation plan. Another group analyzes the same information using the book’s links, and the two groups have equal resources, time and training.

This design separates two questions: does a shared template help compared with scattered documents; and does explicitly recording links add anything to that same shared template? Comparing three conditions (the usual documents, a shared template and a template with links) separates these two effects, and Chapter 13 gives the details.

The result of this comparison makes the method’s scope of application precise: if recording links yields the greatest benefit in multi-source, high-consequence changes, the full level of the method is for those changes, and the rest are handled at the short level.

What This Comparison Gives the Reader

With this chapter, the reader can trace each component of the method back to its antecedent and know where the book has built on it and where it has gone beyond it. In a project with documented and coordinated review, the book provides a shared language for training and recording; in a project where documents are disconnected, it is the main tool for connecting them.

The Assurance Case: The Closest Structural Antecedent

The NIST glossary defines an assurance case as a reasoned, auditable artifact that supports a top-level claim with structured argument, evidence and explicit assumptions [24, from SP 800-160v1r1]. This definition overlaps directly with the structural core of the book's file and is its closest antecedent.

In Annex 5 of the ICO guidance on explaining decisions made with AI, the assurance case is linked to goals, impacts and risks, implementation actions, and monitoring and re-evaluation [25, introduction and Main components]. So review also has a place in the assurance-case tradition.

The OMG's SACM has standardized the structured representation of assurance cases [26]. The book's seven-component model is not an implementation of SACM and makes no claim of technical conformance with it; it draws inspiration from its logic.

Do Not Mistake a Difference in Application for a Difference in Method

An example of the main claim of a re-architecture file: "Changing the source of verification for this group of cases, under these conditions, is acceptable." Its subject is a change in the processing system, and an assurance case could also be drawn up for the same claim. So the book's distinction lies not in the governmental subject matter but in the operational rules it builds for that subject matter.

The book supplements the general structure of the assurance case with four questions specific to public administration: what is the source of authority; for which fact is the evidence sufficient; to whom is the burden of the change shifted; and how is an error linked to the affected cases and to corrective action? The assurance case in safety engineering does not ask these four questions by default, because its subject is system safety, not administrative justice.

From Decision Narrative to Decision Argument

Recording the reasons for a choice also has antecedents. Compendium, a tool supporting the IBIS method for recording and organizing discussion, is one example of this tradition [27].

The book's method separates three outputs. Minutes show who said what. A decision log records which option was chosen. An argument file shows why the evidence presented supports the conclusion and what weakens that support. A good decision log can do all three; the name of a document does not determine the quality of its reasoning.

Technological Due Process and Automated Administrative Law

In her article "Technological Due Process," Danielle Citron showed that automating administrative decisions erases the boundary between adjudicating cases and rulemaking: when programmers translate policy into code they are in effect making rules, without notice-and-comment procedures or judicial review applying to them. Citron's answer is a new procedural doctrine: an audit trail for every decision, systematic testing of systems, and making the rules embedded in code reviewable [80].

This book's method brings the same logic down from the level of legal doctrine to the day-to-day work of a redesign team. The version of the rule and the data consumed in each case are recorded (Chapter 3), a rule is tested against boundary cases before release (Chapter 10), and every change in decision logic has its own redesign file. The book adds two components to this agenda that are seen less often in the legal debate: the distribution of the burden of change between government and the public, and an architecture of redress for cases decided before an error was discovered.

Karen Yeung analyzes "algorithmic regulation" as a form of social control in which data are continuously collected and behavior is shaped through automated decisions, and she shows that this form of governance intensifies questions of legitimacy and accountability [81]. Through three field studies (the automation of eligibility determination in Indiana, the housing allocation system for homeless people in Los Angeles and the child abuse risk screening tool in Allegheny County), Virginia Eubanks showed that the cost of errors in automated systems falls mainly on the poorest groups [82]. This book's multi-party burden ledger and distributional measures are an operational response to this very finding.

Contestability by Design

The literature on “contestable AI by design” argues that a decision system must respond to human intervention throughout its life cycle: not only after a decision is issued, but in design, training, deployment and review. Alfrink and colleagues have turned this approach into a framework of design features and practices [83]. The architecture of appeal and redress in Chapter 8 is the administrative version of this idea, with one extension: the book carries contestability from the individual decision to the “decision about change,” so that a change in the source of verification or in a rule itself has a file that can be criticized and reversed.

Decision Records and Argument Notation

In software engineering, the “architecture decision record,” introduced by Michael Nygard in 2011, has become a common practice: every important decision is recorded in a short document with its context, decision, status and consequences, so that later teams know why the system is the way it is [84]. The book’s redesign file follows this tradition of brevity and versioning, but it makes five components mandatory that an architecture record does not require: the basis of authority, the sufficiency of the evidence for the fact being verified, the distribution of burden, the route to redress and the event that triggers review.

For mapping the chain of argument, Goal Structuring Notation (GSN) is an established standard in safety engineering that links goals, strategies, solutions or evidence, context, assumptions and justifications in diagram form [85]. The book’s seven-component model can be mapped onto this notation: the claim that the change is acceptable in the position of the goal, the evidence in the position of the solution, and the assumption connecting the statement to the decision in the position of an assumption. A team familiar with GSN can draw the redesign file with the same tools.

The Gap Between Policy and Implementation

In *Recoding America*, drawing on her experience in the federal government and on the crisis of California’s unemployment insurance during the pandemic, Jennifer Pahlka shows that the failures of digital government stem less from technical problems than from the separation of policymaking from implementation and from an accumulation of rules that no one has the authority to simplify [86]. Chapter 5 of this book targets this very gap: an implementation impact file before a regulation is issued, and the redesign file

as a channel that returns the lessons of implementation to the policymaker, with evidence and a responsible person.

A Teaching Example of Contrary Evidence

In Sepidar, a group proposes removing the proof-of-address certificate for cases with a registration. The data completeness report confirms that the record exists, but the data definition document shows that some of the records are correspondence addresses. If the purpose of the decision is to verify the place of activity, the second document is contrary evidence against the sufficiency of general use. Adding it to an annex without changing the recommendation is not a response to the critique.

The revised file shows how the recommendation changed: removal of the certificate is limited to cases in which the type of address and its link to the place of activity have been verified; the rest go to the review route. More referrals raise the cost, and the calculation in Chapter 7 is redone with the new referral share. The contrary evidence changes both the permissible scope of the recommendation and its economic calculation.

This example is also a clear training exercise: give the analyst the contrary document after the initial proposal and see whether they merely file it or revise the claim, the scope of implementation and the cost.

Testing the Method Against the Assurance Case

Comparing the three conditions, usual documents, a shared template and a template with links, measures the effect of organization and of making links explicit. The second question requires a separate comparison: one team uses an assurance case suited to the problem and the other uses the book's administrative tools, with comparable evidence, expertise and training.

The variable being measured is not the number of fields filled in; it is which method leads more often to correcting an unsupported recommendation, identifying the scope of an error's effect and choosing a feasible action, and at what cost. The book expects the four questions specific to public administration (authority, sufficiency of evidence, distribution of burden and redress) to make the difference at exactly this point.

Applying It in Your Unit: Chapter 2

Action within the unit's authority	Backing in this chapter
For every important recommendation under review, record the six components of the reviewable link in a table, and write the assumption connecting the statement to the decision and the event that triggers review in separate columns.	Section ``The Distinguishing Unit: A Reviewable Link": the statement ``an address exists" states only that a record exists, not that it matches the place of activity.
At the proposal review meeting, give the analyst the contrary document after the initial proposal has been presented and see whether they merely file it or revise the claim, the scope of implementation and the cost.	``A Teaching Example of Contrary Evidence" in Sepidar: adding a contrary document to an annex without changing the recommendation is not a response to the critique.
When reviewing completed files, count in how many cases recording the links actually corrected an unsupported recommendation; if the fields have been filled in as a formality, return the template to the short level.	Section ``The Claim This Book Takes On": if filling in fields becomes a formality, the method must be shortened.

End of the Free Sample

Thank you for reading this sample of *Re-architecting Government, Volume I: Redesigning Administrative Decisions*.

The full book is available on Leanpub in PDF and EPUB formats.

For organisational purchases, multi-user licences or official invoices, contact ceo@govog.tech.