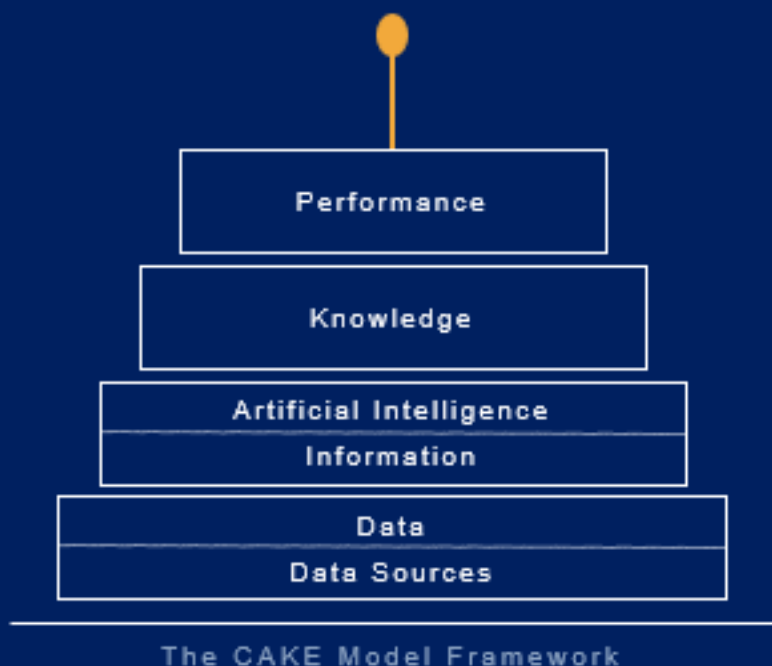


Learning to Bake the Cake In the AI Era



Building the Digital System, Layer by Layer

Annie Green
Mountain Quest Institute

Dr. Annie Green has been a pioneer in Knowledge Management (KM), starting in the early days of KM research and education. Her work on Valuing Intangible Assets has been ground-breaking and is still the benchmark for other researchers in this area. She has produced other insightful books in the KM domain, and this one is the culmination of her decades' long research, teaching, writing, and practice – practices where she played the key role in designing, integrating, and implementing KM systems in several large organizations. She is in a class by herself; and no doubt the readers of this book will benefit from her insights. Enjoy the cake! I found it to be delicious!

—Michael Stankosky, DSc

Former KM Lead Professor, The George Washington University (GW)

Founder, Co-Director, GW's Institute for Knowledge and Innovation

Editor-emeritus, VINE, The Journal of Information and Knowledge Management Systems

For years, organizations have struggled to connect the dots between data, technology, AI, knowledge, people, and performance. Too often, each is treated as a separate initiative, when the real value comes from how they work together. *Learning to Bake the CAKE* gives readers a practical way to see the whole system. It explains the core layers of a digital foundation, the connections between them, and the human behaviors required to make them useful, trusted, and sustainable. For anyone serious about moving beyond fragmented digital transformation efforts, this book offers a clear and accessible framework for understanding what matters, why it matters, and how to make it work in practice.

—Bill Kaplan

Founder-Working Knowledge^{CSP}

Dr. Annie Green has a gift: she can create a narrative that communicates critical elements of knowledge in a way that is folksy, accessible, and memorable. By weaving common sense into a tapestry of knowledge, skills, and ability, she puts forward a way to learn, a way to remember, and a way to implement complicated design needs in a smooth, easy framework. In other words, she connects ideas together in a way that aligns the goals with the knowledge flow of an enterprise (be it one or of many) to enable success. She bakes the CAKE. And, as she says in the preface, "A well-made cake does not happen by accident. It takes intention, balance, and care." Annie brings those attributes, and more, to this book, which readers will find equally fun and informative.

—Dr, Julie J.C.H. Ryan

CEO, Wyndrose Technical Group

Learning to Bake the CAKE is a rare, practical masterpiece that turns the complexity of digital transformation into a clear, actionable recipe. With the elegant *CAKE Model – Connect, Align, Knowledge, Enable* - this book reframes technology, data, AI, and people as a single, interdependent system and shows exactly how leaders and practitioners must design, govern, and sustain that system to create trustworthy outcomes. It reads like a strategist's playbook and a practitioner's checklist at once: rigorous where it must be, refreshingly pragmatic where others stay abstract.

Imagine a Michelin-star kitchen where every dish depends on perfect *mise en place*: the farmer's produce (inputs), the pantry and cold room (data management), the line cooks and stations (information systems), the head chef's secret sauces (AI management), the recipe book and passed-down techniques (knowledge management), and the dining room's feedback loop (performance management). In that kitchen, the CAKE Model is the head chef's system – no single ingredients or station can save a bad meal; only coordinated craft, standards, and timing produce a memorable service. The book's "jelly" between layers and its "icing and piping" governance metaphors make this culinary analogy vivid and useful: interfaces, metadata, policies, and controls are the invisible techniques that turn raw ingredients into a plated experience.

What makes the book indispensable is how it pairs architecture with humanity. The three human additives—*Digital Savvy*, *Digital Mindset*, and *Light and Value*—are treated not as optional garnishes but as essential culinary skills: curiosity and verification are the tasting spoons, ethical judgment is the quality control, and transparency is the plating that lets stakeholders see and trust what's on the plate. This humane emphasis ensures the CAKE Model is not just about building systems that work, but about building systems people can understand, steward, and improve.

For executives and technical leaders, the CAKE Model is a diagnostic lens and an investment roadmap. It helps you see where trust is leaking, which capabilities to shore up before scaling, and how to move from fragmented pilots to resilient, explainable systems that deliver measurable outcomes. For teams and educators, it supplies a shared language and concrete practices that bridge the gap between strategy and day-to-day engineering, turning scattered projects into a coherent, governed capability.

Read *Learning to Bake the CAKE* and you'll stop chasing shiny tools and start preparing a foundation that lasts. This is not theory dressed up as strategy; it is a disciplined, humane guide for anyone who wants digital capability to be deliberate, defensible, and continuously improving. Use it as a recipe, a checklist, and a manifesto—and watch your organization move from improvisation to mastery.

—Marvin Powell
Founder & CEO Coach Powell Training and Development

Every once in a while, a book comes along that reframes how I see a familiar challenge. This was one of those books for me. It completely changed how I think about data, AI, and business value. More importantly, it answered a question I've wrestled with for months: Why do organizations invest so heavily in technology, yet still struggle to create measurable business outcomes?

What resonates with me most is the concept of connective tissue, which links data, systems, people, processes, and decisions. It sounds obvious once you see it, yet it's something I've seen many organizations overlook. The CAKE framework brilliantly demonstrates that success isn't just about building great technology; it's about ensuring everything is connected and aligned end-to-end. Without shared definitions, clear ownership, effective governance, and a common source of truth, even the best systems can end up working against one another.

I also appreciate the emphasis on knowledge, capturing decisions and making them accessible so both people and AI can learn from and build upon them. Dr. Annie Green brings these concepts to life through compelling real-world examples. The Heathrow Terminal story, in particular, powerfully illustrates that technology alone doesn't solve problems; it must enable people to do meaningful work under real operating conditions.

As you read, don't skip the Baker's Notes to become a Master Chef. Think of them as your concise executive cheat sheet that reinforces each chapter and makes the key ideas easy to remember and apply. I lead AI products and can confidently say this book has given me a fresh perspective on creating value from technology.

— Bolade Olawale
Product Manager, Dell Technologies

In *Learning to Bake the CAKE*, Dr. Annie Green transforms the often-complex world of digital transformation into a remarkably accessible and insightful framework. By blending systems thinking, knowledge management, artificial intelligence, and leadership principles, she demonstrates that successful digital ecosystems are built not merely on technology, but on intentional design, human judgment, and organizational trust. Green's CAKE model offers readers a practical roadmap for understanding how data, information, knowledge, and performance must work together to create lasting value. This is a timely and important contribution for leaders, practitioners, and anyone seeking to navigate the complexities of the digital age with greater clarity and purpose.

—Jesse Vaughan
President & Creative Director Vaughan Creative Partners, LLC
35-Time Emmy Award-Winning Storyteller | TEDxRVA Speaker

"Learning to Bake the CAKE" is a timely and thoughtful perspective on what organizations often miss in digital transformation. This framework is very approachable, making this a valuable resource for both technical and business audiences.

My favorite section is "Cleaning the Kitchen." My grandmother always taught me to clean as I cooked. To her, a meal was not truly finished until the kitchen looked as though no one had cooked there at all. That lesson translates very well into digital spaces. Sustainable digital transformation is about continuously maintaining, refining, and retiring what no longer serves the organization."

— **Dr. Tori Reddy Dodla**

Digital Transformation Executive, Author, Editor IGI Global

Organizations are at an inflection point in AI adoption. The technology is capable enough to be genuinely transformative, yet unreliable enough to be genuinely dangerous when mismanaged. The temptation is to treat AI as a product to be purchased rather than a system to be governed. This book makes the case through framework, case study, and sustained argument that AI without strong data foundations, knowledge management, performance oversight, and human judgment at every critical point is not a capability. As such, AI is not an advantage; it is a risk. That argument needs to be made clearly, persuasively, and repeatedly, and this book makes it well.

The greater good, in the context of digital systems, is not faster automation or more impressive AI outputs. It is an enterprise that can be trusted: by the people who work in them, by the customers who depend on them, and by the communities they affect. Trust is built through exactly the disciplines this book describes: clean data, reliable systems, responsible AI, preserved knowledge, honest performance measurement, and human beings who bring savvy, mindset, and a commitment to visible, valuable outcomes. A book that helps organizations build trust at scale is a book that contributes to something genuinely important. This book does just that. Bottom line, a cake relies on its ingredients to taste good as AI depends on data to be a system. The kitchen is ready.

—**Dr. Glenda Arrington**

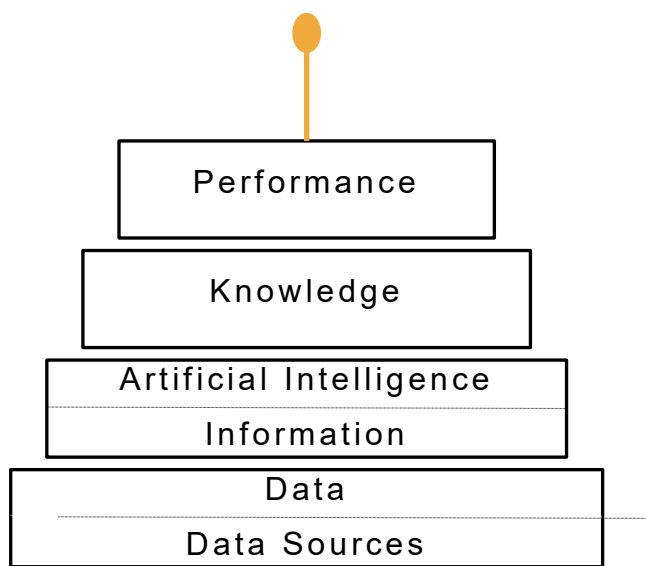
Inspector General (Retired)

At a time when organizations are struggling to make sense of accelerating technological change, *Learning to Bake the CAKE* offers a practical framework for understanding how data, AI, knowledge, and people must work together to create digital systems that are both effective and trustworthy.

—**Deborah Westphal**

*CEO, Karen Toffler Charitable Trust
Futures & Systems Think*

Learning to Bake the CAKE In the AI Era



The CAKE Model Framework

Building the Digital System, Layer by Layer

By Annie Green
(Edited By Alex Bennet)
Mountain Quest Institute

Copyright © 2026 Annie Green

All rights reserved. This publication may not be reproduced, stored or transmitted in any form without permission from the publisher or author.

FIRST EDITION (Hardback)

MQIPress

Frost, West Virginia
303 Mountain Quest Lane, Marlinton, WV 24954
United States of America
Telephone: 304-799-7267
eMail: drgreen2004@gmail.com
www.mqresearchcenter.com
www.mountainquestinn.com

ISBN 978-1-968006-49-5

Dedication

This book is my personal learning journey through education, research, and practice. I dedicate this book to all the learning institutions I attended, all the organizations in which I was employed, all the colleagues who I collaborated with and learned from, known and unknown, and the global population that is a continuing part of my professional digital journey.

Appreciation

I am forever grateful to Dr. Alex Bennet's guidance, mentoring, and contributions to my journey. She kept my view clear as I navigated the paths. I would like to extend a special thanks to Dr. Michael Stankosky for all our long conversations and for him making the statement "Know the Content of the Cake Mix" and blessing me with its use. In addition, I would like to give a shout out to my colleague and sorority sister Dr. Glenda Arrington for always spreading the word about my work.

Definition



- **C**onnect/Connective (tissue)
- **A**lign/Alignment
- **K**nowledge (Capture)/Knowledge Flow
- **E**nable/Enablement

Preface

Most organizations struggle because everything they build digitally feels fragmented or disconnected. Data lives in one place, applications in another, AI shows up as a pilot somewhere else, and knowledge sits in documents no one can find. Everyone hopes it will all come together. But too often, it doesn't.

“Learning to Bake the CAKE” is a book about how all the parts of a digital system fit together and how people make that system work. It is about moving away from fragmented efforts and toward something more integrated, intentional, and trustworthy. This book uses a simple CAKE metaphor to facilitate better understanding.

A good cake is not just a collection of ingredients. It requires the right layers, the right connections between those layers, and the right finishing touches. If one of these three requirements is missing, or poorly prepared, the whole cake suffers. A digital system works the same way, and this book is your recipe.

We are living in a time in which data, AI, KM, and digital systems shape everything we do. From how decisions are made to how organizations operate and even how we interact with each other. But while the technology keeps advancing, our understanding of how to use it has not kept pace. Too often, organizations treat digital transformation as a collection of projects instead of a system, which results in confusing data, broken workflows, AI that is not trusted, knowledge that gets lost, and decisions that don't improve.

The path away from this fragmented existence begins with a simple but powerful question: What does a complete, functioning digital system look like and how do we make it work in real life?

This book guides the reader step-by-step through a structured, practical way to think about digital systems using the CAKE model. The CAKE model is built on six foundational technical layers that make up every digital system: Data Sources - where everything begins, Data Management - where trust is built, Information Systems - where work happens, AI Management - where intelligence is created, Knowledge Management - where learning is preserved, and Performance Management - where decisions improve.

Each layer is explained in clear, practical terms, with real-world examples and stories that show what happens when things go right and when they go wrong. The reader is guided through how these layers depend on one another by understanding the “jelly between the layers” or the interfaces, metadata, and logic that allow information to move clearly and safely from one layer to the next.

No matter how well a system is built, it will not succeed without supportive human actions. So, this book highlights three human elements or “additives”: Digital Savvy - how people use tools, data, and systems effectively, Digital Mindset - how people *choose* to think, question, and act, Light & Value - how we make systems visible and outcomes meaningful.

Finally, the book focuses on something many organizations overlook, sustaining and improving, judging whether a system is working, cleaning up what no longer serves, capturing and preserving knowledge, and improving instead of repeating mistakes.

This book is not just meant to be read. It is meant to change how the reader thinks. As you read, you will start to notice patterns: where things break, where trust is lost, where decisions weaken, and where opportunity exists. More importantly, you will start to see how to fix them by understanding the system.

This book is written for anyone who plays a role in the digital world, including students learning the fundamentals, professionals trying to make systems work better, technical teams building and managing platforms, and leaders making decisions about digital investments.

Whether you are deeply technical or not, the goal is the same: To help you see what matters, and why it matters, so you can use digital systems more wisely and more effectively.

A well-made cake does not happen by accident. It takes intention, balance, and care. The same is true for digital systems. If we understand the ingredients, connect the layers, and apply the right mindset, we can build systems that are not only functional, but trustworthy, usable, and meaningful. Let's Start Baking!

Annie Green

Contents

Preface | i

Foreword | 1

Introduction: | 3

The “CAKE” Model | 3

The Digital System Recipe | 4

Section One: Six Mixing Bowls | 8

Layer 1: Input (Data) Sources (The Bottom) | 9

Collect Raw Ingredients | 9

Prepare Raw Ingredients | 10

Ingredients: Input (Data) Sources | 12

Combine Ingredients | 15

When Bad Inputs Become Bad Decisions | 17

Baker’s Notes | 17

Layer 2: Data Management Systems | 19

Select Utensils | 19

Prepare Utensils | 20

Ingredients: Data Management System | 21

Combine Ingredients | 23

When Poor Data Management Delays Real-World Action | 27

Baker’s Notes | 27

Layer 3: Information Management Systems | 29

Select Utensils | 29

Organize Utensils | 30

Working the Mixture | 31

Ingredients: Information Management System | 33

Combine Ingredients | 35

When Weak Information Systems Turn Work into Chaos | 40

Baker’s Notes | 41

Layer 4: Artificial Intelligence (AI) Management Systems | 42

Select Utensils | 42

Prepare Utensils | 43

Working the Mixture | 45

Ingredients: Artificial Intelligence System | 47
Combine Ingredients | 49
When there is Intelligence and Not Enough Real-World Info | 53
Baker's Notes | 53

Layer 5: Knowledge Management (KM) Systems | 55

Select Utensils | 55
Prepare Utensils | 56
Add Some Flavor | 58
Ingredients: Knowledge Management Systems | 60
Combine Ingredients | 62
When a Knowledge System was Undermined | 65
Baker's Notes | 65

Layer 6: Performance Management System (The Top) | 67

Select Utensils | 67
Prepare Utensils | 69
Count the Candles | 71
Ingredients: Performance Management System | 73
Combine Ingredients | 75
When Performance Suffered | 78
Baker's Notes | 79

Bake the Layers | 80

Six Cake Layers | 82

Section Two: Additives | 84

Additive 1: (Jelly, Icing & Piping) Digital Savvy | 86

Stack the Layers with Jelly Between Them | 87
Icing and Pipe the Cake | 91
When Digital Savvy was Missing from the Loop | 92
Baker's Notes | 93

Additive 2: (Flowers) Digital Mindset | 95

Embrace the Mindset | 96
Prepare for the Occasion | 99
Share the Nutrition Facts | 103
When Human Judgement Stop Asking Questions | 104
Baker's Notes | 105

Additive 3: (The Candles) Light & Value | 106

Shape the Candles | 106

Prepare to Light the Candles | 108

Light the Candles | 111

When Candles Burn Bright but Point in the Wrong Direction | 113

Baker's Notes | 114

The Digital "CAKE" | 116

Section Three: Serve the Cake | 117

The Decision to Serve | 118

Slice/Serve the Cake | 119

Judge the Cake | 121

Come Back for Another Slice | 123

When the Slice was not Ready | 125

Baker's Notes | 126

The Empty Cake Plate | 127

Section Four: The Recipe | 128

Save the Recipe | 129

Share the Recipe | 131

Clean the Kitchen | 133

A Cake Could Not Hold | 135

Baker's Notes | 136

Preserve, Share, Sustain Learning | 137

Conclusion | 138

A Final Word | 138

Afterword: The Digital Leader as Master Chef | 141

Appendix A: Glossary of Terms | 146

Appendix B: List of Acronyms | 150

References | 153

About the Author | 163

Additional Readings | 164

Foreword

I've spent most of my professional life helping organizations capture, share, apply, and grow their institutional knowledge. Often, we'd uncover treasured troves of valuable knowledge that had slipped through the cracks before ever having a chance to reach the point where it could do its best. It wasn't because the people and the tools weren't capable. It was because the many parts were rarely if ever put together as an integrated whole. Data lived in one place. Applications in another. The knowledge itself was encoded into digital artifacts scattered all over the place in different formats, hiding in fragments nobody could find. Somehow, this thing called Knowledge Management (KM) was to bring it all together. But after years of throwing good money into KM and seeing little if any improvement, many gave up in frustration. Now, with the rapid growth of AI, the problem has become more overwhelming.

Learning to Bake the CAKE is right on time. It shows that the parts were never really the problem. Rather, finding, aligning, and enabling the connections between them was the real challenge. It was what we'd suspected all along. It is *integration*, getting all the parts to work together harmoniously as one living system.

Enter Dr. Annie Green, whom I've known for more than two decades, starting with her doctoral research on intangible asset valuation to her later work on the organizational brain, and more recently, the nexus of human and artificial intelligence. Throughout the entire period, she has remained one of only a few in the KM community that has dedicated their career not only to seeing the whole system, human and digital, but also finding a way to make it understandable and actionable. Fortunately, she has taken a major step forward in her endeavor by applying a simple metaphor anyone can grasp; *baking a cake*. Make that a *six-layer cake*.

Unlike today's health-conscious cookbooks, additives are strongly encouraged—mostly of the human variety, of course. These include cultivating a savviness and mindset, as well as “lighting the candles” to make their performance clearly visible and the value of outcomes meaningful and well-understood. Also, no cake would be complete without some tasty *jelly* between the layers: the interfaces, the metadata, and the governance that most organizations treat as an afterthought, when in fact are what holds the entire system together.

Guided by the editorial hand of Mountain Quest Institute's co-founder Alex Bennet, Annie brings us something the field has needed for a long time. Just as "the proof of the pudding is in the eating," the same holds true for intelligent/knowledge systems. They have been hiding on the "systems engineering" side of the dessert menu. This book brings them front-and-center as a featured "special," considering not only what's best for the people doing the baking, but also for the people doing the eating. So, whether you're deeply analytical, intuitive, or both, you'll benefit greatly from this new type of recipe book.

Speaking of intuition, whether it's judgment, context, or purely gut instinct that comes when something is about to go terribly wrong (or wonderfully right)—the very things no digital dashboard can fully capture, this book covers it all. At every layer, you'll see that human beings are not obstacles to digital performance. They are the *source*. Data does not clean itself. AI does not govern itself. Knowledge, which is critical to effective decision-making, does not organize itself. Qualified, trustworthy adult supervision is needed. Someone must care, and ensure that the system is working, notice when it's drifting, and take responsibility for what it produces. This is the real essence of digital leadership, another topic Annie has studied and written about extensively. This book is the very best guide to true digital leadership I've encountered to date.

Pay close attention and let Annie show you how to become your organization's Master Knowledge Pâtissier(e). You'll become fluent in every layer, asking the right questions, "taste testing" what the system is producing, and acquiring the discipline needed to remain up to date with the inevitable changes in the market and the environment.

Open the book and step into the kitchen. But, be prepared. You'll find yourself noticing the missing layers in your own organization, the gaps you've been working around for years, then reaching for the recipe to finally begin putting things into alignment, while measuring the outcomes and making the necessary adjustments. The pantry is stocked. The kitchen is ready. The ingredients are labeled. It's time to start baking.

Art Murray, D.Sc.
*CEO, Applied Knowledge Sciences, Inc., First Fellow, International Institute for
Knowledge and Innovation, June 2026*

Introduction

Most organizations do not struggle because they lack ideas, tools, or talent. They struggle because their digital efforts are built like a stack of disconnected projects; data over here, apps over there, a pilot Artificial Intelligence (AI) assistant somewhere else, and everyone hopes it magically adds up to business value. But it rarely does. This book introduces an integrated approach to digital systems.

The CAKE Model

We integrate for outcomes when we *Connect* systems end-to-end, *Align* on definitions and ownership, *Capture Knowledge* so decisions and Standard Operating Procedures (SOPs) are reusable, and *Enable* teams with the operating muscle to adopt, govern, and improve the system. Let us stop creating fragmented silos and start building scalable digital systems. The CAKE Model is built on four integrated principles:

- **C**onnect/Connective (tissue) — Emphasize linking layers, systems, and handoffs end-to-end.
- **A**lign/Alignment — Share definitions, ownership, governance, and “one version of the truth.”
- **K**nowledge (capture)/Knowledge flow — Capture decisions/SOPs and make them usable across humans and AI.
- **E**nable/Enablement — Make integration usable in practice (operations, adoption, training, mindset).

Connect is linking the layers of a digital system so source inputs, data, information (services), intelligence, knowledge, and performance do not operate as isolated parts. In practice, connection is what allows signals to move across sources, data to flow into systems, AI to work within real operations, and learning to improve the whole architecture rather than one disconnected tool. Strong connections reduce friction, duplication, and blind spots by making the handoffs between layers explicit and dependable.

Align is creating shared definitions, ownership, priorities, and governance across the digital system so people and technology work from the same understanding of what matters. Alignment ensures that data means the same thing across teams, systems support the same operating goals, and decisions are made with a common view of truth, accountability, and risk. Without alignment, well-built systems create confusion because each part optimizes for itself instead of the larger outcome.

Knowledge is capturing, organizing, sharing, and reusing what the organization learns so decisions, procedures, and lessons do not disappear between people, projects, or systems. In a digital system, knowledge gives continuity across the layers by preserving what worked, what changed, and what should guide future action. It turns experience into institutional memory that supports both human judgment and AI grounding, making the overall system more consistent, teachable, and resilient.

Enable is giving people and teams the practical ability to adopt, operate, govern, and improve the digital system. Enablement includes the training, operating discipline, support, leadership, and cultural readiness that turn architecture into real performance. A digital system may be well designed on paper, but without enablement it becomes underused, poorly governed, or unable to improve as conditions change. Enablement is what makes the full system usable, sustainable, and scalable in practice.

The “CAKE” model has six technical layers and three human additives, which supports a digital system foundation to stop treating digital systems as a collection of tools and start running them like a complete system. Executives get a clean investment and governance lens. Practitioners get a shared checklist for delivery and operations. The organization gets outcomes people can trust, repeat, and scale.

Every digital experience, an app you use at school, a customer service portal, a payroll system, a dashboard, or an AI assistant/agent depends on a complete digital system. This book presents a practical view, with nine components to form a viable digital system. Six align with the underlying architecture, how digital work is designed, built, and managed. Three align with the human-in-the-loop, because people decide what to trust, what to act on, and what is ethical. Whether you are a student learning the basics, a user trying to get your job done, a staff member improving operations, a technical specialist building and running platforms, or an executive investing for impact, this “recipe” helps you see what matters and where to focus.

The Digital System Recipe

If you have ever heard, “The data is wrong,” “The system is down,” “The model is hallucinating,” or “Nobody can find the latest policy,” you’ve seen the same root issue of a missing ingredient in a layer or the whole layer is missing. Each layer can look “fine” on its own, but the gaps between layers create friction, risk, and slow adoption, which shows up as delayed programs, ballooning operational costs, and a steady erosion of confidence in anything labeled ‘digital transformation.’

The technical foundation of any digital system is not a single platform, it is an interconnected architecture that moves from inputs (data sources), to

trusted data (data management), to reliable delivery (information systems management), to intelligence at scale (AI management), to institutional memory (knowledge management), and finally to cognitive control (performance management). These layers provide practical checklists to help navigate the evolution of a digital system, because if any one of them is missing or weak, the overall digital experience becomes slower, riskier, and harder to scale. The six-layer technical foundation is as follows:

- **Layer 1: Input (Data) Sources:** This is where information first enters the system. It includes things like transactions, sensor readings, documents, events, prompts, and human input.
- **Layer 2: Data Management:** This is where raw inputs are cleaned up and made trustworthy. Here, data is organized, protected, governed, and given shared meaning so people and systems can use it with confidence.
- **Layer 3: Information Systems Management:** This is where trusted data is turned into working digital services. Applications, workflows, infrastructure, security, and operations all come together here to help the organization run reliably.
- **Layer 4: AI Management:** This is where digital capability becomes usable intelligence. Models, prompts, grounding, monitoring, and governance work together to make AI helpful, dependable, and safe enough to use in real work.
- **Layer 5: Knowledge Management:** This is where what the organization learns is captured and reused. Decisions, procedures, practices, and lessons learned are organized and kept current so people and AI can stop reinventing the same answers.
- **Layer 6: Performance Management:** This is where the entire system learns and improves. Measures, feedback, judgment, and learning come together here to create shared awareness, support better decisions, and help the organization keep getting better.

The six technical layers are represented as a stacked cake. A way to think about the six technical layers is that each one turns the output of the previous layer into something more usable and more scalable. Integration is less about “new tech” and more about making sure the handoffs are explicit, governed, and operationally owned.

The technical foundation layers work in tandem: data enters through sources, is made trustworthy through data management, is delivered reliably through well-run information systems, is amplified responsibly through AI

management, is preserved and reused through knowledge management, and is improved through performance management. When the technical foundation is cohesive, teams can scale digital services with fewer outages, lower risk, faster delivery, and clearer accountability; when it is fragmented, even “good” application/apps and AI initiatives struggle to produce consistent results.

However, even the best technical architecture underperforms without the right human capabilities around it. Human-in-the-Loop, the people foundation, enables the behaviors, norms, and shared language that help individuals interpret information, question outputs, and make responsible decisions, especially when automation and AI are involved. These three additives work together as follows:

- **Additive 1: Digital Savvy:** This is the practical know-how people need to use digital tools well. It includes things like cyber awareness, data literacy, good judgment, and the ability to adapt as tools and platforms change. Digitally Savvy people can separate signals from noise, understand the limits of data, recognize risk, and use technology with more confidence and care.
- **Additive 2: Digital Mindset** – This is the way people choose to work in digital environments. It includes the habits and attitudes that make technology use responsible, truth-seeking, collaborative, and adaptable. A strong Digital Mindset shows when people verify before sharing, respect context and data quality, question manipulation, and use technology in ways others can trust.
- **Additive 3: Light & Value** – This is the way organizations make digital work both visible and worthwhile. Light helps people see what is happening in data, pipelines, models, and decisions so the system can be understood, governed, and improved. Value helps people focus on the benefits they are trying to create, such as productivity, speed, trust, and differentiation. Together, they connect visibility with results, helping answer both “can we see it?” and “did it matter?”

The people foundation, or human-in-the-loop enablers determine whether the technical foundation is used well. Digital Savvy builds practical skills and equips people to navigate tools, data, and risk; Digital Mindset shapes how the skills are applied and enforces the habits of truth-seeking, collaboration, and responsible judgment; and Light & Value provide the visibility and outcome orientation that turn intangible assets into measurable impact. Together, they make adoption stick, strengthen trust, and ensure that automation and AI improve decisions rather than amplifying errors. The three additives are what make the system usable by humans.

Even a beautifully engineered stack underperforms if the people using it do not know how to interpret outputs, challenge questionable results, or collaborate around shared facts. That is why the “additives” matter. They are not soft add-ons; they are the adoption and trust engine.

These nine foundational components explain why digital initiatives succeed or struggle. They also create a shared language across roles: users can describe pain points clearly, staff and technical teams can pinpoint where controls or capabilities are missing, and leaders can invest in the right improvements. Memory Hook: Think **6 + 3**. Six technical layers build the digital architecture, the CAKE itself. Three human additives make it safe, trusted, and ready to serve. Together, they are the complete recipe.

SECTION ONE

Six Mixing Bowls



Figure 1: A Bowl for Each Layer - Six Mixing Bowls

Section One is where we roll up our sleeves and begin building the digital “cake” layer by layer. Before any organization can trust AI, make better decisions, preserve knowledge, or improve performance, it must first understand what the full digital system is made of and how each part depends on the others.

This section follows the journey of a digital system from the bottom up. It begins with raw ingredients; the data sources, signals, prompts, and human inputs that first enter the system; and traces how they are cleaned, managed, shaped into information, refined into intelligence, preserved as knowledge, and finally used to improve performance.

A strong digital system is created by carefully preparing each layer and making sure it can support the next. The six layers of the CAKE Model starts with the six-mixing bowls. What goes into each one determines what comes out of the one above it.

Each layer in this book follows a recipe structure. First, we look at the ingredients; what each one is and what enters the layer. Then we look at the method; how those ingredients are captured, connected, and prepared. Finally, we look at the tradeoffs; what each choice costs and what it risks. You will see some repetition between the narrative and the ingredient lists. That is intentional. In a recipe, the story and the checklist serve different readers at different moments.

Layer 1

The Bottom: Input (Data) Sources

Layer 1 sits at the bottom of the cake because every digital system begins with what first enters it. Before there is trusted data, usable information, AI-driven intelligence, reusable knowledge, or measurable performance, there are sources. Data sources are the points where the digital system first receives prompts, raw signals, records, instructions, and human input from the world around it.

If these inputs are weak, incomplete, noisy, or poorly understood, those problems do not stay at the bottom. They rise through every layer above, corrupting reports, misleading AI outputs, and undermining decisions. When the sources are clear, dependable, and professionally managed, they give everything that follows a stronger foundation.

Collect Raw Ingredients

The raw ingredients of Layer 1 are four types of input: prompts, raw signals, data, and instructions. Each one enters the system differently and carries different risks if it is poorly designed or carelessly managed.

Prompts are inputs from a human or another system that supply intent and context — the direction a system cannot “sense” on its own. For example, a supervisor copies a customer complaint email (source) into a support copilot and enters the prompt: *“Draft a reply in a calm tone, acknowledge the issue, propose next steps, and include our refund policy. Do not promise anything outside policy. Output 5 bullets plus a 2-sentence email draft.”* The prompt becomes an input that the system uses to transform the source content into a structured response. Prompts strongly shape behavior because they specify the task, scope, and rules. The *purpose* of prompts is to make intent explicit, repeatable, and support governance by making instructions reviewable, testable, version stamped, and auditable. Poorly designed prompts, those with unclear goals, missing constraints, or unsafe instructions, can produce misleading or off-policy results even when the underlying data and model are sound.

Raw signals are the unprocessed cues that come directly from the world. They are what a human sense or a machine detects before anything has been cleaned, labeled, or interpreted. A microphone picks up sound waves during a meeting. Those waves carry no meaning until speech-recognition software converts them into words. Raw signals are inherently noisy and imperfect. They

must be interpreted and validated before anyone, human or AI, treats them as dependable.

Data is what raw signals become once they have been captured, structured, or recorded in a form that humans and systems can store, analyze, and use. When speech-recognition software converts sound waves into written words with timestamps, those words are data. The AI can now analyze them, summarize them, or search across them. The key questions to ask about any data source are: Where did this come from? How was it collected or cleaned? What might be missing or distorted?

Instructions are the rules or logic that tell a human or a system how to use data and what actions to take. A prompt that says, “*Summarize this meeting transcript and flag unresolved decisions*” is an instruction that guides how the system processes the data and what it produces. Instructions shape outcomes. Poor rules or logic can produce misleading or unsafe results even when the underlying data is sound.

Before any of these inputs can be stored, analyzed, or acted on, they must be in digital form; converted into the bits and bytes that computers can process. A sound wave becomes a sequence of numbers. A paper form becomes a structured database record. A handwritten note becomes searchable text. The conversion step matters because errors made here such as poor image capture, lost context, bad formatting, quietly damage accuracy at every layer above. Only what is in digital form can be processed by software. Everything else remains invisible to the system.

In summary, collecting raw ingredients is about more than gathering signals, prompts, data, and instructions. It is about recognizing that what enters the system first shapes everything that follows. When organizations understand their inputs clearly, capture them carefully, and treat them as the beginning of a larger chain of trust, they give the entire digital system a stronger start. This first layer matters because good ingredients do not guarantee a good outcome, but poor ingredients always guarantee trouble before the cake ever reaches the oven.

Prepare Raw Ingredients

Raw inputs do not arrive through a single door. They enter a digital system through four different interfaces and understanding how each one works changes how you design, govern, and trust the system.

Picture a hospital that has just deployed an AI assistant to help clinical staff manage patient intake. In the space of a single morning, four things happen simultaneously.

At the front desk, a nurse enters a patient's reported symptoms into an intake form. That is a human interface at work — a person providing judgment, context, and qualitative information that no sensor could supply. Human input can add nuance and interpretation, but it can also introduce inconsistency, delay, and error. The system depends on people not just at setup and oversight, but at every point where judgment matters more than speed.

Down the hall, a patient monitoring unit measures heart rate, oxygen levels, and temperature every thirty seconds. That is a sensor interface — a device detecting signals from the physical world and converting them into data without any human involvement. Sensors are fast and continuous, but they do not understand what they measure. A faulty sensor, or one placed incorrectly, introduces error before any AI model is ever involved.

In the background, the hospital's intake system is exchanging data with the regional health authority's records system through a secure connection. That is a network interface — the channel that carries information between systems, devices, and external sources. Networks enable scale and integration, but they also introduce dependency. If the connection is slow, insecure, or unavailable, even accurate data and capable AI cannot function reliably.

And running quietly beneath all of this, the hospital's software automatically logged every system event, triggering scheduled checks, and applying configuration rules about which alerts require immediate escalation. That is a software interface — the layer where programs, rules, and schedules generate inputs on their own. Software-driven inputs are consistent and scalable, but they embed assumptions. What the software is designed to do, and not do, shapes what the system sees and what it misses.

By mid-morning, the AI assistant is working with a stream of inputs that came from all four interfaces at once. None of them arrived the same way. None of them carry the same risks. And none of them can be trusted without understanding where they came from. See Table 1.

Table 1. Interfaces and Risks

Interface	What It Does	Primary Risk
People	Provide Judgment, Context, and Decision	Inconsistency, Delay, and Human Error
Sensors	Detect and Measure the Physical World	Noise, Placement Error, and Equipment Fault
Networks	Move Information Between Systems and Sources	Dependency, Latency, and Security Exposure
Software	Generate Inputs Through Rules and Schedules	Embedded Assumptions and Logic Errors

The goal of understanding these four interfaces is not to memorize a taxonomy. It is to ask for any input entering a system: Who or what is creating this? What can go wrong at that point? And who is responsible for catching it?

Ingredients: Input (Data) Sources

The “ingredients” for the first layer are the prompts, raw signals, data, and instructions that are captured and turned into digital form so a system can store, process, analyze, and act are described and explained below. The bulleted list below maps the major input sources found in most digital environments and aligns each one with the interface through which it typically enters the system. This is reference material. You do not need to memorize it. What matters is developing the habit of asking for any digital system you work with: *Which of these sources are we relying on? Do we understand them well enough to trust what they produce?*

- **Business Transactions** (records of what people and organizations do): Orders, payments, schedules, tickets/cases, inventory moves, HR actions, and more.
- **Captured-Physical-World Signals** (sensors): Devices that measure the real world and turn them into data (temperature, motion, location/Global Positioning System (GPS), images, sound, biometrics).
- **Commercial/Third-Party Data Providers:** Data and services a company license or buys (address verification, credit checks, shipping status, market pricing).
- **Configuration and Policy inputs** (settings and rules): Rules that tell systems what to do — permissions, thresholds, routing rules, pricing rules, and reference/master data.
- **Data from Other Systems** (integrations): Data moved between systems through databases, Application Program Interfaces (APIs), Extract, Transform, & Load (ETL), and message queues (system-to-system sharing).
- **External/Public Data Sources:** Data from outside the organization (weather, maps, demographics, regulations, public research datasets).
- **Feedback Loops** (outputs that become new inputs): What the system can produce fed back in — user ratings, accept/reject decisions, A(Alpha)/B(Beta) test results, sensor feedback in robotics, and monitoring metrics.
- **Files and Documents** (text, images, spreadsheets): Portable Document Format (PDF), email, chat messages, images, spreadsheets, and

scanned forms — often converted into data using extraction tools (like Optical Character Reader (OCR)).

- **Human-Entered Inputs** (people typing/tapping/talking to a system): Data and commands entered through keyboards, touchscreens, mice, and voice assistants.
- **Machine-to-Machine (M2M) and Internet of Things (IoT) messages:** Devices and systems sending data to each other automatically (smart devices, industrial controllers, vehicles, meters).
- **Network and Cybersecurity Telemetry** (system health & security signals): Logs and alerts from networks and security tools (logins, blocked traffic, malware detections).
- **Software Events** (apps creating inputs): Inputs created by software when something happens — click events, error messages, logs, and performance metrics.
- **Time and Scheduled Triggers:** Inputs created by time such as scheduled jobs, reminders, timeouts, and “end of day/week/month” processing.
- **Web and Social Signals:** Website clicks, app usage analytics, reviews, and social interactions (subject to privacy rules and terms of service).

Table 2 below groups common input sources by the main interface through which they enter a digital system: humans, sensors, networks, and software. This is helpful because it shows that digital inputs do not all arrive the same way. Some come from people entering information directly, some come from devices detecting signals, some flow across connected systems and networks, and some are created automatically by software itself. Reading the table this way makes it easier to see who or what is introducing prompts, signals, records, feedback, and instructions into the system and where quality, speed, and risk need the most attention.

Table 2. Groups of Common Input Sources

Interface	Input Source
People	• Business Transactions • Human-Entered Inputs • Files And Documents • Feedback Loops with User Ratings, Accept/Reject Decisions, and Other Human Responses
Sensors	• Captured Physical-World Signals • Machine-To-Machine (M2M) and IOT Messages

Interface	Input Source
	• Sensor-Based Feedback Loop
Networks	• Data From Other Systems (Integrations) • Network and Cybersecurity Telemetry • Web and Social Signals • Commercial/Third-Party and External/Public Data When Received Through Connected Services
Software	• Configuration and Policy Input • Software Events • Time and Scheduled Triggers • Application-Generated Feedback Loops and Automated Processing Events

Table 3 below shows the typical situations in which each interface is most useful for bringing information into a digital system. Instead of grouping inputs by source, it groups them by the kind of job the interface is best suited for. Humans are used when judgment or explanation is needed, sensors when direct measurement matters, networks when information must move between connected systems, and software when inputs should be created automatically through rules or schedules. Reading the table this way helps clarify not just *where* inputs come from, but *why* a certain interface is the right choice for speed, control, consistency, or context.

Table 3: Typical Use of Interface

Interface	Typical Use
People	• Judgment, Interpretation, Approval, or Qualitative Context is Needed • The Information has not Yet Been Captured in a Structured System • You Need Prompts, Decisions, Feedback, or Document Submissions from People
Sensors	• You Need Direct Measurement of the Physical World • Signals Must Be Captured Continuously or Automatically • Speed, Frequency, or Environmental Monitoring Matters More Than Human Entry
Networks	• Information Must Move Between Systems, Devices, Services, or External Sources • You Rely on Integrations, Telemetry, Web Activity, or Third-Party/Public Feeds

Interface	Typical Use
	• Connectivity, Exchange, and Secure Transmission are Central to the Input Flow
Software	• Inputs are Created by Applications, Rules, Schedules, or Automated Workflows • You Need Repeatable Processing Based on Configuration or Policy • The System Must Generate Events, Triggers, Logs, or Control Actions on its Own

These tables are a starting point, not a complete inventory. Every organization has sources that are specific to its industry, its technology stack, and its history. The value of mapping them is not comprehensiveness, it is awareness.

Combine Ingredients

Each input source brings a different mix of strengths, limits, and tradeoffs. Human input can add judgment and context, but it may be slower or more inconsistent. Sensors can capture events quickly and continuously, but they may also introduce noise or errors. Connected systems and external feeds can improve speed and scale, but they can raise dependency, security, or quality concerns if they are not validated. Software-generated input can make work more consistent and efficient, but only if the rules and logic behind them are sound.

In practice, choosing and managing input sources always involves tradeoffs across four dimensions: quality (accuracy and bias), timeliness (real-time vs. batch), cost (effort to collect and maintain), and risk (privacy, security, and compliance exposure). Understanding those tradeoffs is what allows people to design systems with intention rather than assumption, `deciding which sources are worth the speed, where extra validation is needed and where a slower or lower-cost input is genuinely good enough for the job as follows:

- **Quality**
 - Accuracy: Shows how closely an input reflects reality, because better decisions, automation, and analysis depend on data that is correct enough to trust.
 - Bias: Shows whether an input systematically overrepresents, underrepresents, or distorts reality, because unmanaged bias can produce unfair, misleading, or poor decisions at scale.

- **Timeliness**

- Real-time: Shows whether an input is captured and available almost immediately, because fast-changing situations need current signals to support rapid action, monitoring, and automation.
- Batch: Shows whether an input is collected and processed on a schedule or in groups, because many reporting, reconciliation, and back-office uses gain efficiency and stability without needing instant updates.

- **Cost**

- Effort to Collect: Shows how much time, labor, money, and coordination are required to acquire an input, because even valuable data may not be practical if collection cost is too high.
- Maintenance: Shows how much ongoing work is needed to keep an input current, usable, and dependable, because stale, broken, or unmanaged input sources lose value.

- **Risk**

- Privacy: Shows how much input can reveal about individuals or sensitive activities, because stronger privacy controls are needed when collection or use could expose personal or confidential information.
- Security: Shows how vulnerable an input or its flow is to loss, theft, tampering, misuse, or disruption, because weak security can turn a useful source into an operational and organizational risk.
- Compliance Exposure: Shows how likely an input is to trigger legal, regulatory, policy, or audit obligations, because some sources create added accountability for consent, retention, access, reporting, and proper use.

Balancing the mix is about designing input choices with intention. Not every digital system needs the same combination of speed, accuracy, cost, and control, just as not every recipe calls for the same ingredients. What matters is knowing which sources fit the purpose, where tradeoffs are acceptable, and where they are not. When organizations make those choices deliberately, they build systems that are more practical, more trustworthy, and better matched to the people and outcomes they are meant to serve.

When Bad Inputs Become Bad Decisions

Imagine a company that makes money by trading stocks at very high speed. That company was Knight Capital Group. One day, the firm rolled out a software update that was supposed to support a new trading function. But something went wrong in the setup: one part of the system still had old code sitting inside it, and when the update went live, that old logic was triggered by mistake.

What makes this such a powerful data story is that the system did not stop and ask, “Does this input make sense?” It simply accepted the incoming instructions as valid and started acting on them. In a matter of minutes, the system began sending vast numbers of unintended buy and sell orders into the market. Those bad instructions became tainted operational data inside the firm’s live trading environment, and once they were flowing, the damage spread fast. Traders and leaders were suddenly looking at a system that appeared to be functioning, but it was acting on corrupted inputs.

The result was chaos. Knight Capital Group lost \$440 million in less than an hour. The company did not just have a “technology glitch;” it had a failure at the point where raw machine input entered the organization’s system without enough validation, control, or safeguards. A bad input source stream was allowed to become trusted system behavior.

The lesson is simple and important: raw inputs should never be trusted just because they arrive from inside a system. If organizations do not validate what is coming in, old logic, broken signals, or corrupted data can quickly become actions, decisions, and losses. Bad inputs do not stay contained for long. Once they enter a live system, they can spread faster than people can react.

Baker’s Notes

So, What (Executive View)?: The source of an input affects cost or how expensive it is to capture, risk or the privacy/security exposure, quality or accuracy and bias, and speed or real-time vs. batch. Strong digital systems make inputs reliable through clear ownership, validation, and secure data flows.

Source inputs arrive through many channels: people, sensors, connected devices, software events, transactions, documents, integrations, telemetry, external data, and even time-based triggers. Each input source carries different reliability, latency, cost, and risk characteristics. The practical takeaway is to treat “inputs” as a first-class design concern: know your critical sources, instruments, and validate them. Establish clear ownership and safeguards so bad, late, or biased inputs don’t propagate upstream.

This is where the discipline begins, not in the AI model, not in the dashboard, not in the strategy document, but in the first signal that enters the system and whether anyone thought carefully about where it came from. Layer 2 of the CAKE model takes what Layer 1 collects and does the harder work of making it trustworthy enough to use.

Section Three is where the finished digital “cake” finally leaves the kitchen and meets the people it was made to serve. After the layers have been built and the human additives have been added, this section asks the main test question: will people eat it, enjoy it, and come back for more? Serving the cake is more than launching a system or making it available. It is about making sure the slice fits the user’s needs, role, readiness, workflow, and expectations. A system may be technically complete, but if people find it confusing, unreliable, poorly supported, unsafe, or disconnected from work, they may reject it. This section shows that true success happens when deployment turns into adoption, users can judge the value for themselves, feedback is welcomed, and the system continues improving until people see it as useful, trustworthy, and worth using again.

With the layers baked, the additives applied, and the system brought into full view, the next step is to slice and serve the cake. This is the moment where the digital system moves from design and preparation into real use. Serving the cake represents deployment, access, and the first true interaction between people and the system in their everyday work. It is where all the hidden effort, the ingredients, layers, jelly, icing, and candles, must now hold together under real conditions. Up to this point, the question has been, “Did we build it well?” Now the question becomes, “Can people use it?” Now, the system must be clear, stable, trustworthy, and practical enough to be accepted, adopted, and relied upon. A digital system only proves its value when people are willing to take a slice.

The Decision to Serve

If building the digital system is like baking and decorating a cake, then the decision to serve is the moment the cake is finally placed on the table, sliced, and offered to people to eat. This is where all the hidden work becomes visible. People can see the quality of the cake, how well it holds together, whether the slices are clean, and whether they look worth tasting. In a digital system, this is the moment of deployment and operational use. The serving communicates that the system is ready, trustworthy, and truly usable. If the cake is messy, unstable, dry, or hard to cut, people may hesitate to eat it no matter how much effort went into making it. The same is true for a digital system. If it feels confusing, unreliable, poorly explained, or difficult to use, people are less likely to trust it or adopt it. Presenting the system with quality, clarity, and readiness increases the probability that people will use the system. The decision to serve is not just about saying the system is done. It is about showing that it is good enough, safe enough, and useful enough for people to accept the system into their work. Adoption begins when people see that the slice in front of them is worth trying.

Section Four

The Recipe



Figure 2: The Digital System Recipe

Section Four is where the digital cake moves from being something that was built and served into something that can be preserved, repeated, and improved. By this point, the system has layers, human additives, and users who have had a chance to judge its value. Now the question becomes: how do we keep value from fading?

This section focuses on the recipe, the documented knowledge, decisions, standards, lessons learned, controls, roles, and care instructions that help the organization sustain the system after the first launch. It also makes clear that every kitchen requires cleaning. Outdated content, broken workflows, stale data, weak controls, technical debt, and forgotten lessons can spoil a strong system. Section Four shows that lasting digital value depends on preserving what works, retiring what no longer serves, learning from mistakes, and keeping the whole environment healthy enough to bake, serve, and continually improve.

The recipe is what turns a single successful cake into something that can be made again with consistency, clarity, and trust. In a digital system, the recipe is more than a list of steps; it is the documented knowledge, standards, decisions, and lessons that explain how the system works, how and why it was designed, and how others can use, support, and ensure continuous improvement. Without a clear recipe, organizations are forced to rely on memory, scattered files, and individual heroics, which makes the system fragile and difficult to sustain. But when the recipe is preserved and shared, the

organization gains continuity, resilience, and a repeatable way to keep baking value long after the original builders have stepped away.

Save the Recipe

Saving the recipe is what keeps that knowledge of how the cake was made from being lost, damaged, forgotten, or changed carelessly. A recipe is not useful if it disappears after one celebration, or if the next person finds three different versions and no one knows which one to trust. In a digital system, saving the recipe means preserving the documentation, decisions, standards, lessons learned, and operational knowledge in a way that is stable, current, and protected. It is about keeping the recipe versioned, maintained, and trustworthy so people can return to it later with confidence. It is not about writing things down once.

This matters because organizations do not sustain value through memory alone. People change roles, teams shift priorities, systems evolve, and new requirements appear. If the recipe is not saved, the organization starts to lose continuity. The same questions get asked. The same mistakes get repeated. Governance becomes weaker because no one is sure of the approved approach. But when the recipe is saved, the organization protects its institutional memory and makes it easier to adapt without starting over. Saving the recipe is necessary because it helps the organization keep what has been learned, preserve the effort, and to continue baking value long after the original bakers/builders have stepped away. Saving the recipe is what protects continuity and resilience, because it keeps the organization from having to relearn essential knowledge every time people, priorities, or technologies change. What helps keep a recipe well preserved:

- **Clear Ownership:** Someone should be responsible for keeping the recipe current, trustworthy, and easy to find.
- **Version Control:** The organization should know which version is current, what has changed, and why it has changed.
- **Regular Review:** A recipe stays useful when it is checked often enough to prevent drift, confusion, or stale guidance.
- **Secure Storage:** The recipe should be kept in a place that protects it from loss, damage, or unauthorized change.
- **Easy Access for the Right People:** A preserved recipe only helps if the people who need it can find and use it at the right time.
- **Clear Structure:** Well-organized instructions, definitions, and dependencies make the recipe easier to follow and maintain.

- **Documented Decisions:** It helps to record why important choices were made, not just the final instructions.
- **Integrated Lessons Learned:** A strong recipe improves by incorporating what experience has taught the organization.
- **Consistent Standards:** The recipe should reflect shared rules, naming conventions, and expectations so it does not drift from the larger system.
- **Protection Against Single-Point Dependence:** A recipe is better preserved when it does not depend on one person’s memory to remain intact and usable.

Table 24 contrasts two very different outcomes for saving and not saving documentation: a saved-recipe system or a lost-recipe system.

Table 24: Saved-Recipe versus Lost-Recipe System

Saved-recipe system	Lost-recipe system
Preserves critical knowledge in a stable, trusted form.	Rely too heavily on memory, habit, or scattered files.
Help new people learn without starting from guesswork.	Forces people to piece things together through trial and error.
Supports continuity when staff, tools, or priorities change.	Becomes fragile when key people leave or circumstances shift.
Makes improvement easier because decisions and lessons are visible.	Repeats mistakes because prior learning is hard to recover.
Strengthens resilience by keeping the organization grounded in what it already knows.	Weakens resilience because the knowledge base erodes.

Saving the recipe is about protecting the organization’s ability to continue baking value with consistency, confidence, and continuity. When knowledge is clearly owned, versioned, reviewed, secured, and made easy to access, it becomes more than documentation, it becomes a durable foundation for resilience, learning, and trust. Without that discipline, organizations slowly drift into confusion, repeated mistakes, and unnecessary dependence on individual memory or individual heroics. But when the recipe is preserved well, the organization gains something far more powerful than a set of instructions: it gains a reliable way to carry forward what it has learned, adapt without losing direction, and keep producing strong outcomes when experiencing change.

Share the Recipe

If baking the cake is like building the digital system, then sharing the recipe is what allows other people to make the same cake again without starting from guesswork. A good cake is not scalable if only one person knows how to make it. The recipe is what turns one successful effort into something that can be repeated, supported, improved, and shared across more generations. In a digital system, the recipe is the documentation, guidance, standards, decisions, and know-how that explain how the system works, what ingredients it depends on, how the layers connect, what the guardrails are, and how people should use and maintain it. Sharing the recipe means the system is no longer trapped in one team's memory or in the head of one expert. It becomes something that others can understand, trust, operate, and build upon.

This matters because digital value fades quickly when knowledge stays trapped in pockets of experience instead of being made usable by others. Without a shared recipe, onboarding takes longer, support becomes inconsistent, improvements are harder to coordinate, and the system becomes fragile when key people move on or leave. But when the recipe is shared clearly, the organization gains repeatability, resilience, and continuity. New people can learn faster. Teams can work from the same understanding. Governance becomes easier to apply, and the system is more likely to keep delivering value because the knowledge needed to sustain it is no longer hidden. The organization can move from one good cake to a reliable way of baking value again. This is how institutional memory is preserved. What a good digital recipe should include:

- **Purpose:** A clear explanation of what the system is meant to do and why it exists.
- **Ingredients:** The key inputs, data sources, knowledge sources, and dependencies the system relies on.
- **Steps:** The main processes, workflows, and handoffs that show how the system works.
- **Roles:** Who owns what, who maintains what, and who is responsible for decisions, support, and change.
- **Rules:** The standards, policies, controls, and guardrails that keep the system consistent and safe.
- **Definitions:** Shared terms, metadata, and naming logic so people interpret the system the same way.

- **Access Guidance:** Who should use the system, what access they need, and how use should be controlled.
- **Limits:** What the system is not designed to do, where caution is needed, and when human judgment still matters.
- **Care Instructions:** How the system should be monitored, updated, supported, and improved.
- **Lessons Learned:** What has already been discovered through use, so others do not have to start from zero.

There are challenges that can occur in undocumented systems. These challenges include:

- **Knowledge Trapped in People:** When key know-how lives mostly in individual memory, the system becomes fragile and hard to sustain.
- **Onboarding Takes Longer:** New people must learn by trial and error instead of being guided by clear, reusable instructions.
- **Support Becomes Inconsistent:** Different people solve the same problem in different ways, which creates uneven results.
- **Mistakes Repeated:** Without documented lessons, the organization keeps relearning the same hard lessons.
- **Scaling Becomes Harder:** It is difficult to extend, replicate, or transfer the system when no shared recipe exists.
- **Accountability Becomes Blurry:** When processes and decisions are undocumented, it is harder to know who owns what and why choices were made.
- **Trust Weakens:** Users lose confidence when the system behaves differently depending on who is involved.
- **Improvement Slows Down:** It is harder to refine what is not clearly written down, compared, and reviewed.
- **Risk Increases When People Leave:** Turnover can take away essential knowledge that no one else can easily recover.
- **The System Dependent on Few Experts:** Instead of becoming an organizational capability, it stays tied to a handful of individuals.

Table 25 shows the difference between a system built on a shared, preserved recipe and one that depends too much on undocumented memory.

Table 25 shows that real organizational resilience comes from preserving the recipe so success can be repeated, sustained, and strengthened.

Table 25: Shared-Recipe vs. Undocumented Systems

Shared-Recipe System	Undocumented System
Can be understood and supported by more than one person.	Depends too heavily on what a few people remember.
Onboards new people faster because the knowledge is visible.	Slows down onboarding because people must learn by trial and error.
Scales more easily across teams and situations.	Becomes fragile when the original builders are unavailable.
Preserves lessons, standards, and decisions.	Repeat mistakes because knowledge is hidden or lost.
Turns one successful build into a repeatable way of creating value.	Risks become a one-time success that is hard to sustain.

Sharing the recipe is what keeps a good system from becoming a one-time success that fades with time, turnover, or memory loss. When the recipe is clear, the organization can repeat what works, improve what does not, and carry its learning forward with more confidence. Sharing the recipe protects institutional memory, supports consistency, and helps the organization keep baking long after the first cake is served.

Clean the Kitchen

If building, serving, and sharing the cake represent designing, deploying, and using a digital system, then cleaning the kitchen is what keeps the whole operation healthy enough to keep producing value. No matter how good the cake is, the kitchen cannot stay useful if the counters are covered in spilled batter, old ingredients are left out, pans are never washed, and nobody knows what can still be trusted for the next bake.

In a digital system, cleaning the kitchen means the ongoing work of maintaining what has been built by removing clutter, fixing what is broken, retiring what is outdated, tightening controls, updating knowledge, correcting drift, and making sure the environment stays safe, usable, and ready for the next cycle of operations. Digital systems do not lose value only through failure, they also lose value quietly through neglect, duplication, stale content, weak controls, technical debt, and growing confusion. Cleaning the kitchen is how

the organization protects clarity, trust, and readiness after the excitement of building and serving has passed. How to clean the kitchen:

- **Retire outdated content** so people and systems are not relying on old instructions, stale reports, or expired knowledge.
- **Fix broken workflows** so tasks do not keep getting stuck, routed poorly, or worked around in inconsistent ways.
- **Remove duplication** so teams do not maintain multiple versions of the same data, guidance, or functionality.
- **Refresh metadata and labels** so search, navigation, and retrieval continue to work clearly and consistently.
- **Tighten access and controls** so the right people keep the right permissions, and unnecessary exposure does not grow.
- **Address technical debt** so temporary fixes, shortcuts, and aging components do not slowly weaken the system.
- **Review feedback and incidents** so recurring problems become improvement opportunities instead of permanent frustration.
- **Keep documentation current** so people can still understand how the system works and how it should be used.

This is what happens when systems are neglected, or lacking digital savvy and adult judgement:

- **People are no longer sure which version to trust**, so they keep asking around instead of relying on the system.
- **Search returns too much clutter or too little value**, making useful information harder to find when it is needed.
- **Teams keep building workarounds** because the official process feels broken, outdated, or too hard to use.
- **Duplicate content and duplicate functions keep growing**, so effort is being spent maintaining the same thing in too many places.
- **Metadata, labels, or definitions are drifting**, which makes navigation, reporting, and interpretation less consistent.
- **Access no longer matches real need**, so some people cannot get what they need while others can see more than they should.
- **Small issues keep recurring** because feedback is noticed but not turned into fixes.

Table 26 shows the difference between a clean system and a chaotic one, the condition in which people must work and make decisions. Table 26 makes clear that cleanliness in a digital system is really about maintainability and readiness: a clean system stays usable and prepared for the next cycle of work, learning, and improvement, while a chaotic one becomes harder to sustain because disorder accumulates faster than value.

Table 26: Clean versus Chaotic System

Clean System	Chaotic System
Uses current guidance, current data, and trusted controls.	Leaves people guessing which version, rule, or source is still valid.
Makes work easier because clutter and duplication are reduced.	Creates friction because people work around stale content, broken steps, and repeated confusion.
Supports trust because the environment feels maintained and intentional.	Weakens trust because neglect makes the system feel unreliable or confusing.
Stays ready for the next cycle of use, learning, and improvement.	Becomes harder to sustain because disorder keeps accumulating faster than value.

Cleaning the kitchen is what keeps the digital system from slowly losing its value after the first successful bake. A clean kitchen makes the next round of work safer, faster, clearer, and more dependable. That is why cleanup is not a side task. It is part of sustainment, and sustainment is what protects value. Cleanup is part of digital maturity because mature organizations do not just build systems; they sustain them by keeping the environment clear enough to trust, improve, and use wisely.

A Cake That Could Not Hold

Southwest Airlines had an operational meltdown in late 2022. Severe winter weather hit many airlines, but Southwest Airlines had a much harder time recovering because its crew-scheduling and operational systems could not keep up with the disruption. Schedulers were forced to work manually, matching crews to aircraft by hand, because of the scheduling technology. The airline had planes, crews, and customers, but the information systems that were supposed to coordinate all the work were not effective or efficient enough to restore order quickly.

On the surface, it looked like a weather story, and weather certainly lit the match. But weather alone did not cause Southwest to unravel the way it did. The deeper issue was that the airline's digital and operational "cake" could not hold together under pressure. Its crew scheduling and recovery tools had known limitations and could not keep pace with the scale of disruption, forcing people into exhausting manual work just to figure out where aircraft and crews were supposed to be. In CAKE terms, the technical layers were not strong enough where they mattered most: the information systems layer could not reliably support operations at scale, and the performance layer could not help the organization observe, interpret, decide, and recover fast enough once the disruption spread. The result was not just delayed flights. It was a system-wide collapse in coordination, visibility, and response that stranded travelers, separated crews from aircraft, and made the organization temporarily unable to serve its own purpose.

This was not only a technology failure. It was also a human-foundation failure. A strong technical stack can still wobble under stress, but organizations recover when people share a strong digital mindset, clear decision structures, healthy cross-team communication, and the discipline to escalate problems before they become chaos. Southwest's own post-incident analysis pointed to communication gaps, weak escalation, and recovery processes that depended too heavily on people improvising around brittle tools instead of being supported by resilient systems and shared awareness.

The cake was weak, but so was the way people had to work around the cake. This shows that digital failure rarely belongs to "the technology people" alone or "the operations people" alone. It happens when weak layers and weak human additives meet at the same time. The lesson is clear: organizations do not become resilient by buying better tools, or by telling people to work harder. They become resilient when the technical foundation is strong enough to support real work and the human foundation is strong enough to use that system wisely, collaboratively, and under pressure.

Baker's Note

So, What (Executive View)?: A strong recipe reduces dependence on memory, speeds onboarding, improves consistency, and protects the organization from losing critical knowledge when people, roles, or technologies change. Without it, even well-built systems become fragile because too much depends on undocumented know-how and individual recall.

The recipe is the part of the digital system that makes success repeatable instead of accidental. It preserves what was built, why key choices were made,

how the layers connect, and what others need to know to maintain, support, and improve the system. The recipe is documented knowledge made usable: standards, decisions, dependencies, lessons learned, and instructions that help new people learn faster and help experienced people avoid starting over. When the recipe is clear, current, and shared, the organization gains continuity, resilience, and a stronger foundation for learning. When it is missing, scattered, or outdated, the system becomes harder to sustain because too much of its value lives in individual memory instead of in a form others can trust and reuse.

Preserve, Share, and Sustain Learning



Figure 3: Recipes for a Digital System

Section Four explains that a digital system becomes truly valuable when its “recipe” is saved, shared, and maintained. The recipe represents the documented knowledge, decisions, standards, lessons learned, roles, controls, and care instructions that allow a system to be understood, repeated, supported, improved, and trusted after the original builders move on. Without a clear recipe, organizations become dependent on individual memory, scattered files, and individual heroics. With a preserved recipe, they gain continuity, faster onboarding, stronger governance, and a reliable way to keep producing value.

This section shows that sustaining digital value requires more than documentation. The kitchen must be cleaned. Outdated content must be retired, workflows repaired, metadata refreshed, access reviewed, technical debt addressed, and lessons from incidents turned into improvement. A mature digital organization does not simply build and serve the cake; it keeps the environment healthy enough to bake again. The lesson is clear that repeatable value depends on a recipe people can find, trust, use, improve, and protect.

CONCLUSION

This book has shown that a digital system is never just a collection of tools, data, or isolated projects. It is a layered, connected, and human-dependent system that must be designed, protected, used, and improved with intention. Through the CAKE model, the reader has seen that strong digital foundations begin with trustworthy inputs, move through disciplined data and information management, expand through responsible AI, preserve value through knowledge management, and improve through performance management.

What is made clear is that no technical architecture succeeds on its own. Digital Savvy, Digital Mindset, and Light & Value are what make the system usable, ethical, visible, and worth trusting. The main learning point is powerful: organizations do not gain lasting value from digital systems by accident. They gain it when they understand the **content of the cake mix**, connect the layers well, protect the structure, and choose to use the system with care, truth, and shared responsibility.

A Final Word:

Digital transformation projects remain difficult to sustain because many organizations still approach transformation as a set of technology upgrades rather than as an integrated operating system. Research widely cited in the field of digital transformation, including studies published by McKinsey & Company, notes that roughly 70 percent of transformations fail to achieve their stated goals, often because organizations do not build the capabilities, engagement, and cross-functional coordination needed to make change stick¹.

Disconnected systems make the problem worse. When data, applications, workflows, AI tools, and knowledge repositories are not connected, teams work from different definitions, duplicate effort, loose context between handoffs, and struggle to see one reliable version of reality. As a result, transformation slows down, user trust weakens, and promising digital investments produce fragmented outputs instead of measurable value. Disconnected systems turn digital transformation into a collection of isolated slices rather than a complete cake: each tool may function on its own, but the organization cannot reliably

¹ See McKinsey & Company, *Unlocking Success in Digital Transformations* (2018), which reports that fewer than one-third of transformations succeed in improving performance and sustaining gains.

connect, align, capture knowledge, or enable people to improve performance across the whole system. Five core takeaways from this book are:

1. A digital system only works well when all six layers are present, connected, and strong enough to support one another.
2. Metadata, interfaces, and governance are not side issues; they are what hold the layers together and make the system trustworthy.
3. AI is only as useful as the information, knowledge, and human judgment that ground and govern it.
4. Digital Savvy and Digital Mindset are essential human additives because tools alone do not create trust, truth, or good decisions.
5. Performance improves when organizations observe, interpret, decide, act, and learn as a continuous cycle rather than treating digital work as one-time implementation.

The CAKE is worth making. The discipline is in making it well, serving it with care, and keeping the recipe so it can be made again.

This emerging cognitive framework is where humans and digital systems collaborate as interconnected partners. This relationship flourishes via symbiotic intelligence, which offers humanity a new horizon where human possibility is expanded through partnership with AI. In this digital era, digital systems can illuminate patterns, accelerate discovery, and open doors that once seemed beyond reach. Those capabilities become significant when humans bring the vision, conscience, compassion, and courage needed to decide where those doors should lead.

Together, this partnership becomes more than innovation; it becomes a shared learning journey that challenges us to grow wiser, more responsible, and more united as we shape technologies that reflect our highest values rather than our deepest fears. Symbiotic intelligence teaches us something profound--the future will not be defined by what machines can do alone, but by what humanity chooses to become when empowered by them.

Integration is the word that appears most often when organizations describe what their digital systems lack. Not more data, better AI, or newer platforms, although all are at some point necessary. Integration is the ability to make the parts work together.

The CAKE Model in this book is, at its core, a model of integration. It shows that data sources only have value when they are managed. Managed data only has value when it flows into reliable information systems. Information systems only have value when they support the kind of intelligence that AI can provide responsibly. AI only has value when its outputs are preserved as knowledge. Knowledge only has value when it is applied to improve performance. And performance management only has value when it sends learning back through every layer below it, strengthening the whole.

Like a layer cake, a system is assembled tier by tier — each one only as strong as the layer beneath it.