

The Skills Pack

Turning Finance Work into
Tested AI Skills



Felix Reyes

grain
PRESS

The Skills Pack: Turning Finance Work into Tested AI Skills

Copyright © 2026 Felix Reyes.

DRAFT — 5 October 2026

All rights reserved for the book’s text and illustrations. You may download, store and print a copy for your own reading and reference. Except as permitted by law or by a separately identified license, reproduction, redistribution, translation or adaptation requires prior written permission.

Companion software, skill files, reusable templates and sample data, including code listings reproduced in this book, are licensed under the MIT License. The downloads include the license texts and scope.

Published by Grain Press, Jacksonville, Florida

ISBN 979-8-9955988-9-3

Companion files: <https://felix.yt/skills/>

This draft is provided “as is”, without warranty. To the extent permitted by law, the author and publisher disclaim liability for loss or damage arising from its use. The applicable licenses state their full warranty and liability terms.

Agent Skills is an open specification published at agentskills.io. Product and company names referenced herein may be trademarks of their respective owners and are used for identification purposes only.

Preface

Recurring finance work depends on rules that are often scattered across spreadsheets, reporting templates and the experience of the person preparing the numbers. Introducing an AI model adds another place for those rules to drift. A useful skill needs a defined task, reliable calculations, clear output requirements and a way to check the result.

The Skills Pack shows how to put those pieces together.

The book is written for people who own recurring finance processes and the maintainers who turn those processes into working tools. Its examples include revenue bridges, budget-versus-actual commentary, close checklists and cash forecasts. Each starts with a business task and follows it through implementation, testing, review and release.

A skill brings together instructions, scripts and supporting material for a task. A host is the program that loads skills and runs a model against them. A pack groups related skills with the rules they share. Those shared rules cover matters such as how figures are presented, how sources are identified and what a deliverable must contain. Keeping them together gives the team a defined place to make a change and a record of which version each pack uses.

All worked examples use InterStationary, a fictional distributor with reproducible data. Its departments format the same revenue figure differently and duplicate the rules for their reports (Section 1.1). The method supplies five controls: explicit rules, repeatable computation, bounded judgment, accountable ownership and evidence for release.

Four rules carry the method through every chapter: numbers come from scripts; structure comes from an output contract a program can check; voice comes from a shared layer every skill in a pack reads; evidence comes from tests that read what a host wrote and report counts a reviewer can check.

Scripts calculate figures and apply explicit classification rules. Where content is fixed, a script can write that content directly. The model handles the parts that require selection and wording within a defined output contract. A reviewer assesses the explanation, checks for important omissions and decides whether the result is ready to use.

Tests support that decision. They can identify specified calculation failures, missing sections, unsupported figure strings and some incorrect associations between figures and names. A note can pass every automated check and still omit the most important movement or offer an explanation the data does not support. The examples include these failures because they show where a check ends and a reviewer's judgment begins.

Each pack has an owner, a version, test cases and a release process. Its rules and recorded results let someone who did not build it review a change before release.

Chapter 12 identifies the configurations tested and records their failures alongside their successes. These results provide a starting point for evaluation on your own data. Whether a pack saves time, improves a report or works on another configuration must be measured in that setting.

Begin with one recurring task. Make its rules explicit, build the smallest useful skill and test what it produces. The rest of the book develops that first skill into a pack your team can review, maintain and improve.