Nic Carroll, Ed.S., CISSP, CETL



# Reimagine IT

## Frame it. Build it. Loop it. Launch it. Tend it.

A new way to run projects in IT and beyond. Define the end game, plan and build at the same time with AI as a partner, refine in short loops, launch to a test group or to everyone, and keep improving. It works for an app, a process, or any project, and fits a day, a week, or a month.

| 01 | 02 | 03 | 04 | 05 |
|---|---|---|---|---|
| **Frame It** | **Build It** | **Loop It** | **Launch It** | **Tend It** |
| A shared end game | A tangible first draft | A version ready for real people | A live result in test or production | A result that keeps getting better |

**A project methodology by Nic Carroll.** One cycle, any pace. Tend it leads back to Frame it.

THE SHIFT

# Build It Together, Now

A continuous improvement cycle for any project, at any pace.

THE OLD WAY

### Plan for Months, Then Build

Scope for weeks, build behind closed doors, deliver once, and hope it fits. Every change pulls the whole team back together.

THE NEW WAY

### Build It Together, Now

Frame the end game, build a real draft in the room, refine in short loops, and launch in days or weeks. Changes are small and fast.

THE FIVE MOVES

| MOVE | WHAT IT MEANS | WALK OUT WITH | PACE |
|---|---|---|---|
| 01 **Frame It** | Define the end game: what we are solving, for whom, and what success looks like. You may not know what it will look like yet. | **A shared end game** | An hour or less |
| 02 **Build It** | Plan and build at the same time. AI handles the detailed work so the group creates a real first draft together. | **A tangible first draft** | Hours to a day |
| 03 **Loop It** | Refine in short, fluid bursts: a five minute conversation, a 30 minute check in, or test group notes. Sprints, only far shorter. | **A version ready for real people** | Days to weeks |
| 04 **Launch It** | Release to a test group or to everyone, with plain language communication planned along the way. | **A live result in test or production** | When it is ready |
| 05 **Tend It** | Improve continuously. Small changes happen fast. When the need changes, Frame it again. | **A result that keeps getting better** | Ongoing |

**One cycle, any pace: a day, a week, or four weeks. Tend it leads back to Frame it.**

WHO IS IN THE ROOM

# Experts Build It In, Not Bolt It On

Bring everyone the project needs to get it done. When the right experts help build it, you walk out with something safe, secure, reliable, and real enough to touch.

**Project Owner**
Builds with an AI partner, runs the loop, and owns the result.

**Project Manager**
Protects a short timebox, picks the room, and owns the launch message.

**Business Analyst**
Turns reactions into the next build list, live in the room.

**IAM and Security Partner**
Sets up access in the room so it works on day one.

**User Partner**
Speaks for the people who will use it and tests the workflow.

**Program Partner**
Brings the real process and often becomes the owner.

**Applications Partner**
Connects to core systems and supports it after launch.

**Development Partner**
Brings the development team into the room to review the build and finish what AI cannot.

**UX Design Partner**
Shapes layout, language, and accessibility during Build it.

**Communications**
Writes the launch message in the words users use.

**Security, access, development, and applications shape the work in the room, supported by automated checks where possible, so what leaves the room is ready to trust.**

SAFE BY DESIGN

## Fast Does Not Mean Reckless

- We work only in organization approved AI tools
- Access uses organization accounts, so only the right people get in
- We decide up front what information the project may use
- Experts review during Build it and Loop it, not after launch
- Every project has one named owner and a backup

RULES OF THE ROOM

## How We Work Together

- If you cannot see it or use it, it is not done
- Right people, short time
- Show, do not spec
- The owner builds. The experts check.
- Change is cheap, so change often

RUNNING SEVERAL AT ONCE

# Many Projects, Every Area in the Room

The Project Owner submits the need. Leadership chooses which projects run, and several can run at the same time. Every project needs one person from each area it touches who has the authority to decide and the access to act in the room. Those people are few, so four habits keep them from becoming the new bottleneck.

### Stagger Build It.

Each expert group sits in only one Build it at a time. Loop it and Tend it need far less of their time, so overlap there is fine. Plan Build it days so they do not collide.

### Name a Backup for Every Partner Role.

Each project already has an owner and a backup. Do the same for access, applications, and UX, so no room waits on one person who is booked somewhere else.

### Keep One Shared List.

One row per project: the owner, the end game in one sentence, the current move, and the partners in the room. It is the only view leadership needs, and it shows at a glance when one expert is in too many rooms.

### Let Automated Checks Carry the Routine.

Scans, access checks, and standard tests run on their own, so experts join for decisions, not for every loop.

*When the list shows an expert in too many rooms, leadership decides which project waits.*

FACILITATOR GUIDE

# Starting the Conversation

For the person leading the room. Ask, then let them answer.

FACTS TO SAY OUT LOUD

- The cycle fits any pace: a one day project, a one week project, or a four week build.
- The room holds everyone needed to finish the project, so nothing waits on someone who was not there.
- People leave Build it with a tangible draft, not a plan about a draft.
- Guardrails are set once, up front, so no project waits on approvals to start.
- Loops can be five minute conversations. Sprints are now days, not months.

FACILITATOR GUIDE

# Questions to Ask

Two questions for each move, and three more for the people in the room.

### FRAME IT

- What problem costs your team the most hours each week?
- Can you say it in one sentence someone outside your office would understand?

### BUILD IT

- What would you want to see or use by the end of Build it?
- Who needs to be in the room to finish this project safely?

### LOOP IT

- What is the fastest way to get feedback back into the work?
- What is good enough for version one?

### LAUNCH IT

- Should this launch to a test group first, or to everyone?
- Who needs to hear about it, and in whose words?

### TEND IT

- If a change took 30 minutes, what would you ask for first?
- Who owns the result when the owner is out?

### YOUR ROLE

- Which part of your job is a gate today, and could it become a guardrail?
- What do you know that a builder cannot learn without you?

### THE FUTURE

- Six months from now, what should take a day instead of a semester?
- What will you do with the hours these tools give back?

### THE HABIT

- Which of our steps reduce real risk, and which exist only because we have always done them?
- What would you need to see to trust work built by someone without tech in their title?

WHAT IT LOOKS LIKE IN PRACTICE

Each case shows a part of the work the cycle assumes: where records live, how a stuck room gets help, how projects share experts, and when a project is complete. The cases are illustrative examples.

USE CASE 1

# A One Day Project

Where the records live

**The situation.** A registrar's office receives hundreds of emails each term asking the same transcript questions: how to order, how long it takes, where it will be sent. Staff spend hours answering the same messages. The Project Owner, an associate registrar, submits the need, and leadership picks it.

**Who is in the room.** The Project Owner, the Project Manager, a Program Partner who answers those emails every day, an IAM and Security partner, an Applications partner, a Development partner, and a UX Design partner. Each one can say yes for their area on the spot.

**How it moved.** Frame it took 45 minutes. The end game: students get transcript answers and submit a request in one place without sending an email. The room decided the form may collect only name, student ID, and request type. Build it ran until midafternoon. AI drafted the form and the answers from the ten most common emails, the access partner set up sign in, and Applications connected the form to the existing request queue. Loop it was one hour: five staff members tried it and the room fixed a dozen wording problems. It launched the next morning to one college as a test group.

**What gets recorded.** The Project Manager saves to the shared folder as the day goes: the end game, the data decision, each draft, the loop notes, the test group, and the launch message. The shared list shows the current move. Nobody writes a separate status report.

> **The result.** If an auditor or an open records request asks about this project a year later, the folder answers the question in minutes.

---

USE CASE 2

# A Room Gets Stuck

How a stuck project reaches leadership

**The situation.** A financial aid office is building a portal that tells students when their award changes. It is a four week build. In week two, the Applications partner learns the portal needs a data feed from an outside vendor, and the current contract does not allow that use of the data. A contract change is required, and no one in the room can approve it.

**How it works.** The Project Manager tells leadership that same day in two sentences: what is blocked, and the decision needed. The message offers two options: pursue the contract change first, or launch without the feed and add it later. Leadership answers the next day: launch without the feed and start the contract change through the normal purchasing and legal process at the same time. The room keeps building everything it can: layout, student messages, and access.

**What gets recorded.** The Project Manager logs the block, the options, the decision, who made it, and the date in the shared folder. The shared list shows the project as waiting on leadership for one day only.

> **The result.** The project loses one day instead of two weeks. The feed is added during Tend it once the contract change is signed, and the decision trail is in the folder.

USE CASE 3

## Three Projects at Once

Every area in the room, without burning out the experts

**The situation.** Leadership approves three projects in the same month: a one week housing check in app, a two week onboarding checklist, and a four week research data request process. All three need the same IAM and Security partner, and two need the same UX Design partner.

**How it works.** The Project Manager uses the shared list to spot the overlap before it causes a delay. Build it days are staggered: housing on Monday, onboarding on Wednesday, research the following Monday. The backup access partner covers the housing Loop it sessions. Automated checks for access, security scans, and accessibility run on every build, so the shared partners join only when a check raises a decision. If the load is still too heavy, leadership decides which project waits.

**What gets recorded.** The shared list shows each project's owner, end game, current move, partners, and backups, so anyone can see who is committed where.

> **The result.** All three projects move forward in the same month, and no expert is in two Build it rooms on the same day.

USE CASE 4

## Completion, Then Tend It

Completion is recognized. Tend it keeps the product working.

**How completion works.** The Project Manager confirms four things: the end game is met, people are using it, support is handed to the Applications partner, and every open item is either closed or moved into Tend it. The project is marked complete on the shared list, the final version is saved to the folder, and leadership recognizes the team. The partners return to other rooms.

**How Tend it works.** Tend it is a regular pulse check, not a full room. The Program Partner and the Applications Partner own it. On a set rhythm they ask three questions: Is it working? Is it current? What needs to change? Small updates are made quickly as needs shift. Each update goes out as a short note, and when a change touches a specific area, that area's partner reviews it before it goes live.

**What gets recorded.** Every change is logged in the change catalog: what changed, why, who made it, when, and who was told. The catalog sits in the project folder next to the original records.

> **The result.** The project has a clear finish line, the product stays current, and its full history is in one place. If the need changes in a bigger way, the Project Owner submits it again and it starts as a new Frame it.