# A Manifesto for System Administrators

CANDACE RAMSEY
REVISED AUGUST 2026
CANDACE.IT.COM

*How this work goes well*

The structure of four values and twelve principles is borrowed from the 2001 Agile Manifesto, deliberately. The words are mine, written for infrastructure work rather than software.

## PART ONE · FOUR VALUES

We are uncovering better ways of running the systems other people depend on, by doing it and by helping each other do it. Through this work we have come to value:

### Shared knowledge over individual heroics

The person who can fix it alone at 2am is valuable. The team that never needed them to is more valuable. Every system one person can operate is an outage waiting for a vacation.

### Systems that work over documents that describe them

A running service beats a beautiful design doc. But note where the line falls. A concise runbook someone else has successfully followed is part of a working system, not a description of one. It is the forty-page architecture document nobody has opened since the review that we are arguing against.

### The customer's problem over the ticket boundary

We rarely talk to customers directly, and that is usually by design. The help desk does. So do the business analysts and the product owners. They are our line of sight into what is actually happening out there. Treat them as partners carrying real information, not as a queue that hands us work.

### Absorbing change over protecting the plan

Hardware slips, funding shifts, a vendor discontinues the thing we standardized on. A plan that cannot survive that was never a plan. It was a wish with dates on it.

> *There is real value in the items on the right. Tools, documentation, agreements, and plans all matter, and most of ours should be better than they are. The left side wins when the two genuinely collide, which is rarer than people reach for it.*

## PART TWO · TWELVE PRINCIPLES

01 **Availability is the deliverable.** The ticket, the document, the meeting, and the dashboard all exist to protect it.

02 **Change is constant in infrastructure.** The goal is not to prevent it but to make it survivable. Small and reversible beats large and careful.

03 **Deliver improvements continuously rather than saving them for an event.** A quarterly window is a batch, and batches hide risk until they release it all at once.

04 **The people who run the system and the people who hear from the customer should be in regular contact.** Not only during an outage.

05 **Build the team around people who can be trusted with judgment.** Then give them the context to exercise it. Context, not just permission.

06 **The fastest way to move knowledge is a conversation.** The most durable way is a runbook someone other than its author has followed successfully.

07 **No system should depend on one person being reachable.** Single points of failure in people are the ones we tolerate longest and regret most.

08 **Sustainable operations means a pace we could hold indefinitely.** If someone is regularly rescuing us, that is a design defect, not a personality trait.

09 **Attention to engineering quality is what buys future speed.** Deferred maintenance is a loan, and the rate is variable.

10 **Simplicity is mostly about doing less.** The system we did not build cannot page us at 2am.

11 **The best designs come from teams that plan together, not from whoever moved first.** Speed at the start is usually borrowed from the middle.

12 **Stop regularly, ask what is actually going wrong, then change exactly one thing.** A retrospective that produces no change is a meeting about feelings.