Table of Contents

How Do We Improve How We Write Requirements and Goals?

As Lean and Agile thinking takes hold in how we get work done, many people have come to downplay the importance of written requirements and other documentation. This makes sense—iterative and incremental approaches don't benefit from exhaustive upfront documentation. But that doesn't mean we stop writing things down, or that clarity in our writing becomes less important.

In fact, when we do less documentation, clarity matters more. We still need to clearly describe problems, express requirements, and define objectives, especially in distributed environments. Fewer words means each word carries more weight.

Natural Language is Slippery

Most of us write in natural language (like English), which is inherently ambiguous. Consider the sentence:

- "Mary had a little lamb."

Depending on your emphasis or context, this could mean:

- Mary (a person) had a pet lamb.
- Mary (the mother) had a baby she called "little lamb."
- Mary ate a small serving of lamb for dinner.
- A sheep named Mary gave birth to a lamb.
- A wolf named Mary ate a lamb.
- Mary's lamb was unusually small.

And so on.

As humans, we're surprisingly good at interpreting these ambiguities - when we speak. But when we write, we strip away contextual cues. This creates space for confusion—especially in technical or goal-oriented writing, where precision matters.

As someone once said in a requirements meeting:

> "The only thing wrong with what I'm saying is what you're hearing."

That's the problem we're trying to solve.

What Guidelines Help Improve the Clarity of Written Requirements?

Years ago, I first heard about "Requirements Engineering" in a conversation with Dan Walsh. Back in the waterfall era, requirements clarity was a big deal so people developed multi-step processes (inception, analysis, modeling, specification, validation, management) to manage it.

While those processes have limited use in today's discovery-driven, product-led organizations, some of the insights are still helpful especially the practical guidelines for improving written clarity. Below are some common traps and how to avoid them.

Avoid Weak and Ambiguous Words

Words like these are vague, subjective, or impossible to measure:

- quickly
- user-friendly
- appropriate
- effective
- normal / regular
- as possible
- support
- reliable
- streamlined
- etc.

> **Guideline**: Replace weak words with precise, measurable terms. What does “quickly” actually mean? How will you test for “user-friendliness”?

Bound Your Lists

Unbounded lists introduce uncertainty about scope. Watch for phrases like:

- “at least…”
- “such as…”
- “including but not limited to…”
- “…by xx/yy/zz or later”
- “etc.”

> **Guideline**: Define clear boundaries. Unbounded lists are impossible to design for or test against.

Define Collections Explicitly

Ambiguity often comes from undefined groupings:

- “The next release must support 802.11 and other network protocols supported by competing applications under Linux.”

Questions immediately arise:

- What counts as “other protocols”?
- Which versions of 802.11?

- What are “competing applications”?
- Which Linux versions are in scope?

> **Guideline**: Either define collections clearly or agree on a shared definition.

Watch Out for Verb Choice and Logical Ambiguities

Subtle changes in verb use can lead to big differences in meaning:

- Does “support” mean enable, allow, ensure, or enforce?
- Does “allow access to registered users” mean others are blocked?

Also:

- Be precise with “each”, “all”, and “only”.
- For example, the word “only” can dramatically change meaning based on placement:
  - Condsider “Mary only gave John a dollar.” (She gave him nothing else.) versus “Mary gave only John a dollar.” (She didn’t give a dollar to anyone else.)
- Avoid ambiguous operators like the EVIL slash (/):
  - “The system shall report/log improper access attempts…” - Does it report? Log? Both? Either?

> **Guideline**: Use verbs that imply specific responsibilities and outcomes. Clarify logical operators (and/or) and access rules explicitly.

Final Thoughts

We don’t need to return to the world of 100-page spec documents. But in a world of fast-moving teams and remote collaboration, clear writing is a powerful enabler of alignment.

Even small improvements in how we write requirements and objectives can reduce misunderstanding, improve delivery, and build trust across teams.

Everything here applies just as much to goals and objectives - not just technical requirements. Whether you’re setting a Sprint Goal, a PI Objective, or even OKRs, vague language creates misalignment.

Consider:

“Improve login performance” — vague.

vs.

“Reduce login response time by 30% for 95% of users by end of PI” — measurable and testable.

The same clarity rules apply. Be specific about outcomes, timeframes, and success criteria—even if you're dealing with higher-level goals.

Want to Know More?

Note:

- This thinking originally came from Dan Walsh and Erik Simmons, both thoughtful product leaders

Additonsl ideas:

- How Do We Write Great PI Objectives
- How Do We Write Good Features?

FAQ, PortfolioManagement, WorkingAgreement, Requirements, PIObjectives, SAFe

From:
https://www.hanssamios.com/dokuwiki/ - **Hans Samios' Personal Lean-Agile Knowledge Base**

Permanent link:
https://www.hanssamios.com/dokuwiki/how_do_we_improve_how_we_write_requirements_and_goals

Last update: **2025/07/30 07:22**