

YAGNI Your Requirements Documentation

by Ellen Gottesdiener

EBG Consulting, Inc.: www.ebgconsulting.com

*(This article first appeared in EBG Consulting's eNewsletter,
"Success with Requirements" Vol. 1, No. 9(2007). - [Subscribe for free here](#))*

YAGNI, or **ya ain't gonna need it**, comes from XP (eXtreme Programming), an agile method. YAGNI is a smart, minimalist approach to product development. It means you don't create any artifact that doesn't provide business value.

But it's important not to misinterpret YAGNI. It *doesn't* mean **don't document your requirements** because you'll never need or use it.

This is an extreme interpretation of a sensible principle. Instead, YAGNI means that you'll benefit from paying **close attention to what you document and how**.

Form and Format

For every unique project, you should adjust both the form and the format of your requirements.

Form: The form of **business requirements** might be a list of goals and objectives. **User requirements** might take the form of scenarios, business rules, and a data model. The form of **quality attributes** might be text statements, a subset of "Planguage" (Planning Language) elements, or quality attribute scenarios.

Format: The format of **scenarios** might be handwritten statements on index cards or templated scenarios recorded in a spreadsheet or Word document. The format of your **data model** might be a rough draft sketched on a poster or whiteboard, or a diagram drawn in a tool such as Visio. If you write your **quality attributes** using Planguage, you might format them in a Word document that will be traced to test cases in a requirements management tool.

Deadly Assumptions

Don't assume that detailed requirements specifications are necessary, even if your organization typically creates them for each project. On the other hand, **don't assume** that the discarded index cards that become acceptance tests and code will be sufficient documentation for your maintenance and support programmers—or for that regulatory agency that might want to audit your systems.

Be practical. If your developers are co-located with the customers and are very familiar with the business domain, they won't read or work off of detailed requirements specification documents. But if you plan to **outsource or offshore development**, it's **essential** to write formal documentation supplemented by visual models.

On an agile project, you know that requirements will evolve. So it **doesn't provide business value** to write a detailed specification with all the requirements that will be needed. Indeed, a study (Standish, 2003) showed that 45 percent of specified requirements were never delivered. While there may be many reasons for undelivered requirements and even a rationale for some documentation of requirements you know you won't deliver, you should strive for a minimalist approach whenever possible. In some situations you will find it more **efficient** to write brief requirements in the form of stories, accompanied by "doneness" tests (such as acceptance tests).

Thinking YAGNI

Here are some questions to help you think in a YAGNI-like way about your requirements documentation.

Question the business value of your documentation:

- Is our customer **willing to pay** for the time it takes to document requirements?
- What is the **risk to the project** and product if we do not create this particular documentation?
- What specific documentation is needed by **external entities** (regulatory, governmental, oversight agencies)?
- How can we reduce the time we spend **revising** documentation?
- How will we know the requirements documentation we have is "**good enough**"?

Question the users' value of your documentation:

- **Who** will use each requirements document we produce, and **how** will they use it?
- Is this documentation **necessary** to architect, build, test, or maintain the product?
- What models (text and visual) can we use to elicit and validate requirements with customers **before** we invest in building and testing?
- What's the **minimum** amount of documentation we need to create a prototype or to demonstrate a partial solution?
- Which requirements documents are most often used to **enhance or maintain** the product?
- Will anyone use this documentation **after** the product is built?
- Who is going to **maintain** the documentation?

In Sum

The YAGNI principle will serve you well if it helps you **question your assumptions** about requirements documentation. Before you invest time and money documenting requirements, make sure you know what the business owners, delivery team, and users **ain't gonna need**.

Standard citation for this article:

Ellen Gottesdiener, "YAGNI Your Requirements Docs," *Success with Requirements*, Vol. 1, No. 9(2007).

Reference

Standish Group International, Inc. 2003. *Standish Group Study*. Reported by Jim Johnson, chairman, XP 2002 Conference.

More Info

Planguage is keyword-driven method for writing precise quality attribute statements (and other project elements). For the ultimate source, see Tom Gilb, *Competitive Engineering: A Handbook for Systems Engineering, Requirements Engineering, and Software Engineering Using Planguage* (Addison-Wesley, 2005).

Writing *quality attribute (QA) scenarios* is a technique recommended by the Software Engineering Institute for documenting quality attributes. You can write QA scenarios freeform or by using templates. Like stories, they are powerful and engaging. They can guide architecture and help you prioritize requirements. The most efficient way to define them is to use workshops. See: Mario R. Barbacci et al., *Quality Attribute Workshops, Third Edition* (CMU/SEI-2003-TR-016) (Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2003).

Author Information

Ellen Gottesdiener, Principal Consultant, [EBG Consulting](#), helps you get the right requirements so your projects start smart and deliver the right product at the right time. Her book *[Requirements by Collaboration: Workshops for Defining Needs](#)* describes how to use multiple models to elicit requirements in collaborative workshops. Ellen's most recent book, *[The Software Requirements Memory Jogger](#)* describes essentials for requirements development and management.

In addition to providing training and consulting services and coaching agile teams, Ellen speaks at and advises for industry conferences, writes articles, and serves on the Expert Review Board of the International Institute of Business Analysis (IIBA) Business Analysis Body of Knowledge™ (BABOK™).

You can subscribe to EBG Consulting's offers a [free monthly eNewsletter "Success with Requirements"](#) offering practical guidance and requirements-related news. When you sign up, you'll receive a free .pdf article by our Senior Associate, [Mary Gorman](#), on an essential requirements modeling technique.