For more information see gilb.com for books, videos, blogs, papers and slides. See especially new book 'Value Planning' for depth.

The older 12 tough questions paper
is at
http://www.gilb.com/dl24

## 20 Tough Questions: Your Superpowers for successful projects

Dramatically increase your influence on the projects

1. we tend to get stuck in building the detailed technical design, you can learn how to focus on the really high-priority outcomes of a project.

2. we tend to look like fools, when people ask smart questions about our projects, you can learn to ask the smart questions, early.

3. you can make those smart-ass architects look silly.

*"You have got to take these questions seriously, otherwise it can damage our stock market price."* the CEO, Kari, told the

senior technical directors. A few months later they fired their junior Director for not having answers to these questions.

These questions can be a cultural shock to your colleagues; so you might want to exercise considerable caution and diplomacy in asking them. Maybe do like my wife would do with me, make me think I thought of it myself.

## 10 Tough Questions You can ask about Requirements

1. Have you **agreed** a set of top-10 critical-value objectives for the product?

2. Are those objectives **unambiguously clear** to all who might have to understand them, the intended readership?

3. Is it clear which requirements the **stakeholders support,** and are interested in?

4. Are the requirements **really values, qualities and results**: not the technology we think will get us there?

5. Is it clear what the **worst acceptable value** delivery level is?

6. Is it clear what the **Wish level** is, and that this is not a commitment yet (Goal level): until we find technology and resources to reach it?

7. Is it clear what the requirement's knock-on **value** is, for example 'economic', or in terms of higher level objectives, if we reach the Wish or Goal level. What is it worth?

8. Do we know the defect density of our specifications? If you can see more than 10 unclear or ambiguous words on a requirements page, is this a threat to understanding your project?

9. Do we have other major stakeholder **levels** that need a *separate* specification of requirements? Like; Business Level, Stakeholder Level, Product Level or Sub-Product Level.

10. Is there any requirement arguably **more-critical** than the top-ten, that we failed to include or specify? Now that we *think* we have a complete set: what is missing?

**10 Tough Questions You can ask about Solutions, Design & Architecture**

11. Are the **designs/solutions specified so unambiguously and clearly,** so that nobody can inadvertently misunderstand them, including *what* to estimate and *what* to implement?

12. Have you estimated the short-term and life-cycle costs, in both time and money, for each major solution idea?

13. Have you looked at the ratio of solution impacts over their costs (solution impacts/solution costs): so you can select the **most efficient** solutions?

14. Have you looked at the **worst-worst case** (credibility and uncertainty) for all value impacts, and all resource impacts?

15. Can you consider implementing the most efficient (effects/costs) solutions **early, to get feedback, learning, and possibly deliver real value** to the field?

16. Can you decompose any design idea, into **smaller, independently implementable, sub-solutions**? High value decompositions can be done earlier.

17. Have you invited **competitive imaginative engineers,** to come up with far more cost-effective solutions than you can show the on your Value Decision Tables? Using the Value Decision Table as a **provocative baseline for discussion.**

18. Is it possible to improve **the Value Decision Estimate**, and improve certainty, by better research on existing experience of the solutions, or by experiments, or pilots? Can you get better solution credibility for deciding what to do first.

19. Can we conduct simple, short-term, this week, A/B **experiments to get better data and experience** on some of the solutions?

20. What can we do to **motivate the best design engineers (and architects)** to analyze our ideas, and come up with *better* ones? Both up front, and after delivery cycle feedback.

**Summary:** rigorous and rich specification of requirements and design will deliver much better value for resources.

Tough questions are instantly usable, at a meeting for example. They do not require expertise to understand and ask, but they require ‘experts’ to answer. They will quickly unmask ‘false salespeople’.

Because these questions work very well, they cause people to get interested in investing more time with my methods.

In the case of the 20 Tough Questions, I was in April 2016 invited to do a keynote for the largest (20,000 trained engineers) user of our methods. These people are engineers not managers, so I made this version of the tough questions to speak to the engineering audience.

————-

*“Do you think we should bring up the long term technical debt costs at this point Judy” ?*

*“Would you like me to suggest early initial wins for our team Bob?”*