Stakeholder Power: The Key to Project Failure or Success

by Tom Gilb

Main Idea:

Stakeholder analysis is taken for granted in some disciplines, like systems engineering and medicine. But is widely ignored in IT and Software projects, especially the Agile Software culture.

There might be one of few user types, and one or few customer types, but there are probably from 30 to 300 or more stakeholder instances or types. Many with multiple and critical needs. Not to mention need conflicting with other stakeholder needs.

Agile is about 'Users' [3] and 'Customers'. But Agile (Scrum for example) rarely to never speaks about or teaches the broader stakeholder concept.

If we fail to acknowledge critical stakeholders, and *their* critical needs: our projects will continue to fail. Our systems will continue to fail. If not initially, then in the longer term.

Dealing with critical stakeholders early, continuously is mature. Waiting until they cannot be ignored, years later is immature, and unfortunately widespread.

Evidence is that the 'Business Analyst' community is not going to do anything on their own initiative. They have failed us for decades.

Enlightened management is going to have to make 'serious stakeholder respect' *mandatory*. If management fails, and they fail regularly, then we might need pressure from laws, regulations and insurance companies to make people be responsible.

But, does anybody really care if we waste massive time and money on IT systems that fail?

No. (I am not joking, just observing).

But you, the reader might be an exception. So this paper is for you.

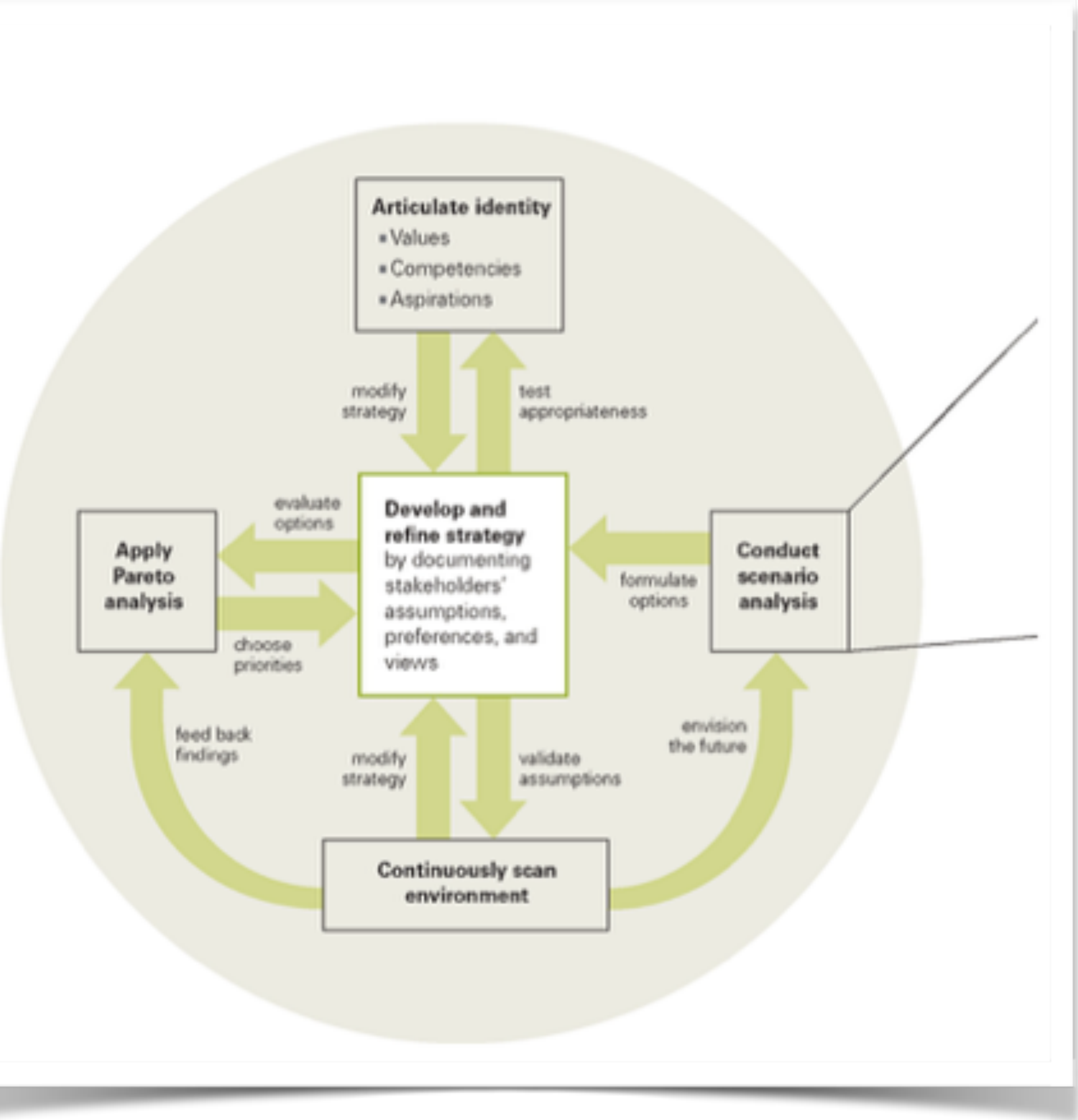

Figure 1: A stakeholder process that can remind us of the continuous and dynamic work needed to keep on top of emerging stakeholders, their needs, and their changes. Source https://hbr.org/2008/05/strategy-as-a-wicked-problem ('Read More' diagram). PPG's Framework.

Definition
A stakeholder is any person, group or object, which has some direct or indirect interest in a defined system.

Stakeholders can exercise control over both the immediate system operational characteristics, as well as over long-term system lifecycle considerations (such as portability, lifecycle costs, environmental considerations, and decommissioning of the system). [4]

Notice:

'or object'.

This includes laws, regulations, plans, policies, customs, culture, standards. Inanimate. you cannot ask them or discuss with them. But you can analyze them, their priority, the degree of relevance. They can determine if your system is illegal, or acceptable. Determine success or failure.

Ten Stakeholder Principles.

1. Some stakeholders are more critical to your system than others.

2. Some stakeholder needs are more critical to your system than others.

3. Stakeholders are undisciplined: they may not know all their needs, or know them precisely, or know their value. But they can be analyzed, coached, and helped to get the best possible deal.

4. Stakeholders may be inaccessible, unwilling, inanimate, oppositional, and worse: but we need to deal with them intelligently.

5. Stakeholders might well ask for the wrong thing, a 'means' rather than their real 'ends'. But they can be guided to understand that. Or their requests can be interpreted in their own real best interests.

6. Stakeholders do not want to wait years, get delays, invest shitloads of money, and then little or no value. They want as much 'value improvement' of their current situation, as they can get, as fast as they can get it. For as little cost as possible,

7. Stakeholders cannot have any realistic idea of what their needs and demands will cost to satisfy. So their adopted requirements need to be based on value for costs, not on value alone. Delivering small increments, based on high value-to-cost, is one smart way to deal with this.

8. If you think you have found 'all critical stakeholders', I think you should assume there is at least one more, and when you find that one, …. They will emerge, and they are not all there at the beginning.

9. If you think you have found all critical needs of a stakeholder, there will always be at least one more need hiding.

10. If you do not understand, and act on the principles below; you will blame your failure on ‘system complexity’, and the unexpected and wicked problems. But in reality it is your own fault and responsibility; deal with it - up front and constantly.

Finally:

It is tempting to write a page or so for every principle above. But there is some virtue in a very short paper too! So, I promise you that I have argued all the points in depth in my book [1]. So if you really want a deeper understanding, you know what to do.

References

1. “**Value Planning: Practical Tools for Clearer Management Communication**”
http://gilb.com/dl853. A manuscript pdf copy.

Search the pdf on ‘Stakeholder’: plenty of hits.

Value Planning, book (free core, cheap the rest).
leanpub.com/ValuePlanning. eBook. Auto new versions.

2. Gilb, Tom, **Competitive Engineering, A Handbook For Systems Engineering, Requirements Engineering, and Software Engineering Using Planguage**, ISBN 0750665076, 2005, Publisher: Elsevier Butterworth-Heinemann. Sample chapters will be found at Gilb.com.

Free CE Pdf: http://www.gilb.com//dl540

3. **User Stories**
User Stories: A Skeptical View.
Paper: http://www.gilb.com/tiki-download_file.php?fileId=461

User Stories paper by Tom and Kai Gilb
In Gilbs' Mythodology Column, Agilerecord.com March 2011

VIDEO: http://vimeo.com/53159408

Smidig 2012 Talk Video Recording, English
8 Minutes
Published on Dec 15, 2012
Tom Gilb discusses the dangers of User Stories at Smidig 2012 (annual conference of Agile software development in Oslo.)

4. Planguage **Glossaries**

CE Planguage Glossary
http://www.gilb.com//dl387

Master Planguage Glossary
http://www.gilb.com//dl830

5. **CONFRONTING WICKED PROBLEMS:
and some Planguage Tools to deal with them**.
http://www.gilb.com/dl866

Version
Start 9 Feb 2016 Eve
This edit = Tuesday Feb 9 2016, 23:42