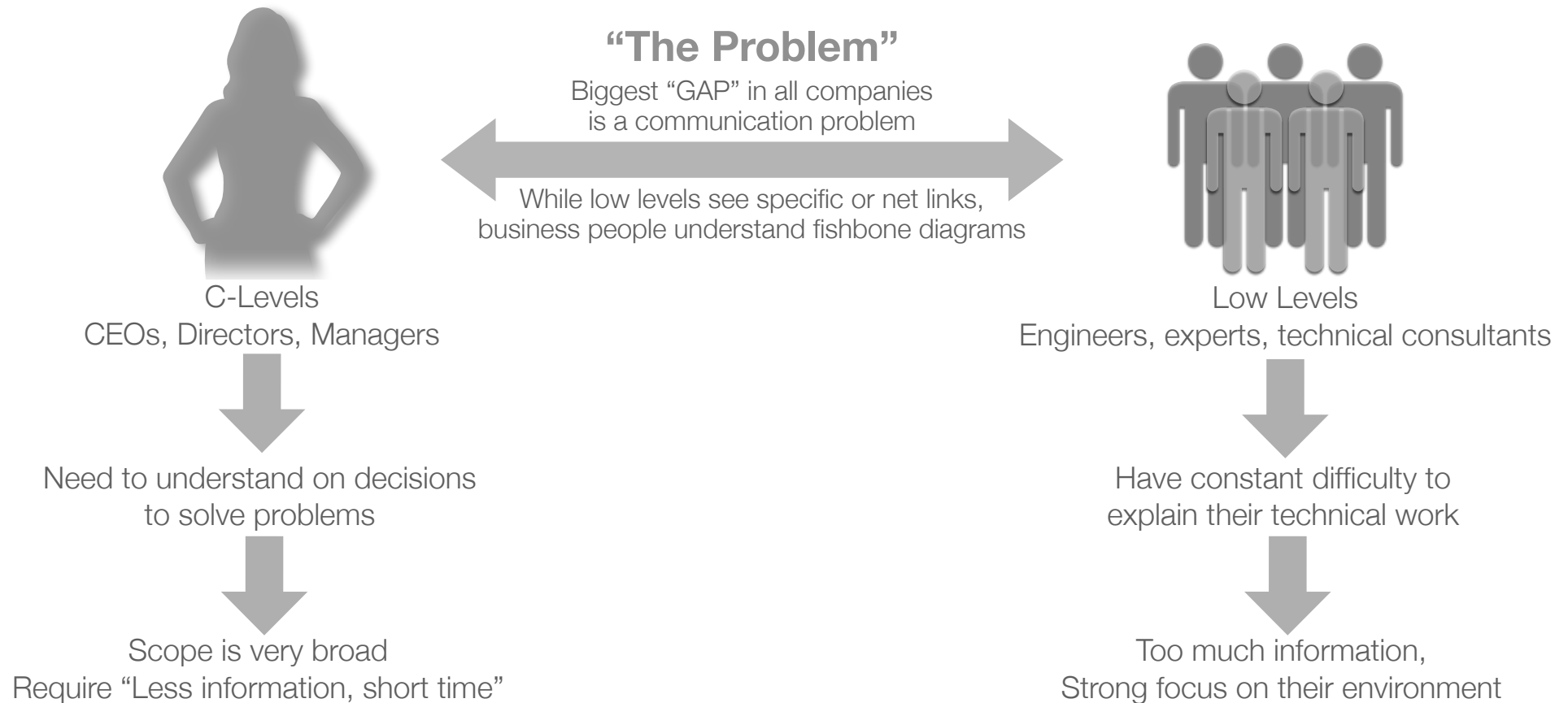


The first GIT interface ever understood by C-Level



Summarizing the Problem:

GIT is a system that is good when working with teams that are all working on the same project but when comes to combine with project management and organizational efforts it becomes a great mess:

- 1 - A "Commit" is hardly a "Task"
- 2 - Lack of RCAs when there are problems
- 3 - GITs require some deep specific training knowledge to fully work on team level environment

With all this 3 "deadly" issues, no C-Level can find value to any GIT system

The first GIT interface ever understood by C-Level

“The Solution”
Geeky GIT Manifesto



	Current GITs interfaces	Solution	What is solved?
1	Abstracted chain representation, hard learning curve to C-Levels	Well explained RCAs & friendly INTERFACE for the chains of coding	Mid & C-Levels know where, what, why and how is the problem and can easier understand on low levels judgement/opinion
2	“Commits” are not “tasks”, “Commits” are pieces of code	Commits are directly related to “tasks” from project plan	Project Managers can easily understand what was done by the developers because it is project related
3	Some GITs have DevOps systems, But never related to project plan	Apply PMI methods for “Disciplined Agile DevOps”	Allows Agile Teams and PMOs work better together at organizational level
4	GITs are fully functional interfaces UX is considerer irrelevant since it's users are normally engineers	Create a UX interface that is C-Level friendly	UX friendly interface allows faster training and faster understanding by mid&C-Levels or even interns and juniors levels