

# Powerful Questions

# Important to remember

# Examples of Evidence

## Step 1: Clear Outcomes

What difference will this/do you want my presence to make?	In [6 months] how will you know that I've been a supporting Scrum Master?
What impact will that have?	How will you know / see / observe that the impact took place?

It's their decision to pull your help.	It's their responsibility to change or keep their behaviour.
Your authentic vulnerability allows them to show vulnerability	Use their word: Pacing

More predictable	In a Sprint Review stakeholder vote (fist to five) and the average improves
	Fewer deadlines missed
	Lead times have smaller variance

## Step 2: Sharing Responsibility

In order to achieve this, what contributions are needed?	When you imagine we achieved this goals, how did we do this?
How can I (you) have your (my) back? How can I (you) make you (me) look good?	How do you want us to behave, when things become difficult?

You don't have to already know, how it's all done. They don't know either.

More self-organised / autonomous	No more requests for X to manager Y, while development works well
	Fewer blocked items per sprint
	Developers vote on how well they could coordinate the work on PBIs, score improves

## Step 3: Establish accountability

What is the first step? How will we know it's successful?	When are we going to check in next?
---	-------------------------------------

Trust & Respect will grow through regular iterations	Prepare them for resistance, that occurs in the middle of any meaningful change
Don't forget: someone needs to gather the data.	

More cross-functional	Average number of developers engaged in a ticket is higher
	Amount of time spent pair-programming per week
	Amount of times a dev work on his not-primary skill
	In the skill table, every skill is owned by at least 3 ppl.

