

## **Asking effective questions: a collaborative problem solving technique**

*If you're stuck for answers then you need someone to ask you effective questions*

Do you know that sinking feeling that comes when you think you've bitten off more than you can chew? There might be so many things to do that you can't choose where to start, or there might be so many hurdles to overcome that failure seems inevitable. Either way you feel paralysed by the sense of impending doom. But don't throw up your hands in despair! You probably already have the answers to your problems: all you need is for someone to ask you some effective questions.

*What makes a question effective?*

An effective question cuts through the mental logjam that we create when things go wrong. Answering it gives us an insight into our problems and stimulates our search for solutions. An effective question is phrased in an open-ended fashion that does not suggest a particular answer and warrants a response of more than just a few words. It usually starts with a "what" or a "how", as "why" questions are easy to misinterpret and can trigger a defensive response. Questions that are intimidating, repetitive, or rambling are obviously ineffective. As an effective questioner it is important to listen without interrupting, waiting respectfully for each answer and considering it when it arrives before giving an opinion or launching into another question.

*What do effective questions achieve?*

Effective questions clarify our goals and help to steer us from a state of confusion or aimlessness toward a set of concrete actions. The dialogue that effective questioning initiates also increases the number and quality of interactions within a team and makes it more effective.

*Some categories of useful effective questions*

The following examples illustrate some categories of question that I have found effective while working as a coach for a software development team:

- “Where are we going?”
- “Where are we now?”
- “Have we been here before?”
- “Whose input are we missing?”
- “What steps ought we to take?”
- “How should we start?”
- “Which are the real obstacles?”
- “Are we ignoring something?”

These can be used in any sequence that makes sense for your team and your situation.

*“Where are we going?”*

If we are totally floundering then effective questions will explore the motivation behind our current activity and focus us on the desired outcome. Simply asking "What are we trying to achieve?" or "How will we know when we're done?" can be enough to get us back on track. Posing Dale Emery's value question<sup>1</sup> - "If you had that, what would that do for you?" – is particularly powerful, as it diverts attention away from a single solution and back to the root of the problem.

*“Where are we now?”*

As programmers well know we can come to new insights by simply retracing the unsuccessful steps we have taken. Asking "What seems to be causing the problem?" or "What have you tried so far?" can elicit this kind of productive 'aha!' moment.

*“Have we been here before?”*

Problems will recur and we can often benefit from effective questions such as "What was the solution last time?" or "What happened before when...?" These questions remind us of the likely effect in the present of repeating actions that were taken in the past.

*“Whose input are we missing?”*

We can often benefit from connecting with people on different teams (or on our own team!) whose strengths and achievements we are perhaps unaware of. Effective questions such as "Have you talked to [her]? What did she say about...?" help to overcome this kind of limiting distance. Alternatively if there is a discussion in which some people remain silent then asking them "So, what do you think?" at appropriate intervals will encourage alternative points of view to be raised.

*“What steps ought we to take?”*

Sometimes we can see both where we are and where we want to get to. In this situation we need effective questions that help to bridge the gap, such as "How could we achieve that when we are starting from here?" If this prospect is too daunting however then it is more helpful to ask "Can we break this down into smaller chunks?" or "What's the first step we are able to take? Then what could we do next?"

*“How should we start?”*

If there are many tasks to be done then we need effective questions that identify our priorities. Asking "If you could only do one thing today then what would you do?" or "What is most important right now?" will help us to focus on the most urgent courses of action.

*“Which are the real obstacles?”*

Sometimes we can seem to hit a dead end where there is nothing more that we can possibly do or try. When this happens we need effective questions that give us courage and allow us to determine which of the perceived constraints are real and which are the product of our fear, miscomprehension, or haste. By asking "What if this was true?" or "What if that were to happen?" we can create options where previously none seemed to exist.

*“Are we ignoring something?”*

If a particular course of action seems too difficult or if we are prone to procrastination then we need effective questions that challenge us to confront what we would otherwise ignore. These questions run the risk of being provocative but if handled thoughtfully then asking "What are you going to do about this?" or "How would you justify doing nothing about that?" can overcome many causes of inertia.

*Asking follow up questions*

Sometimes the first answer we give or receive is too hasty or tentative. Effective follow up questions explore whether a response has been fully thought through, for example "Can you explain how that would work?" or "How can you be sure that...?" Follow up questions that require our understanding to be confirmed can also be effective, such as "So if I heard you correctly then what you're saying is..."

*Summary*

Software development is a difficult activity and as such it is easy to get bogged down in its problems. But we can get unstuck by asking effective questions that facilitate the discovery of solutions. Effective questioning can also benefit software development teams by promoting discussion, co-operation and collaboration.

*About the author*

Tim Bacon works as a consultant for ThoughtWorks Inc in the United Kingdom. The views expressed are his own and are not necessarily those of his employer.

---

References:

<sup>1</sup> [http://www.dhemery.com/cwd/2003/06/the\\_value\\_question.html](http://www.dhemery.com/cwd/2003/06/the_value_question.html)