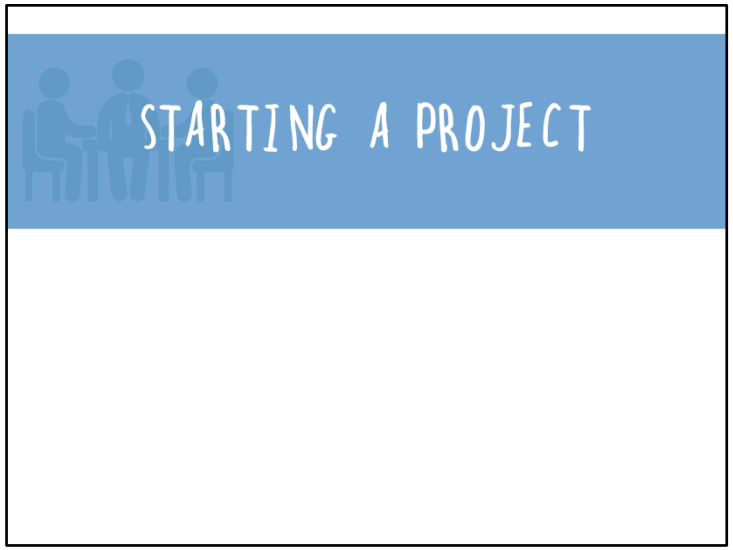
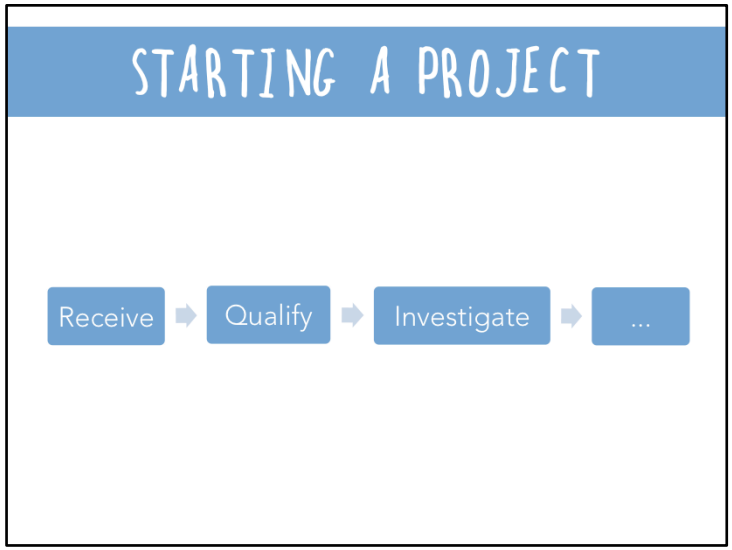
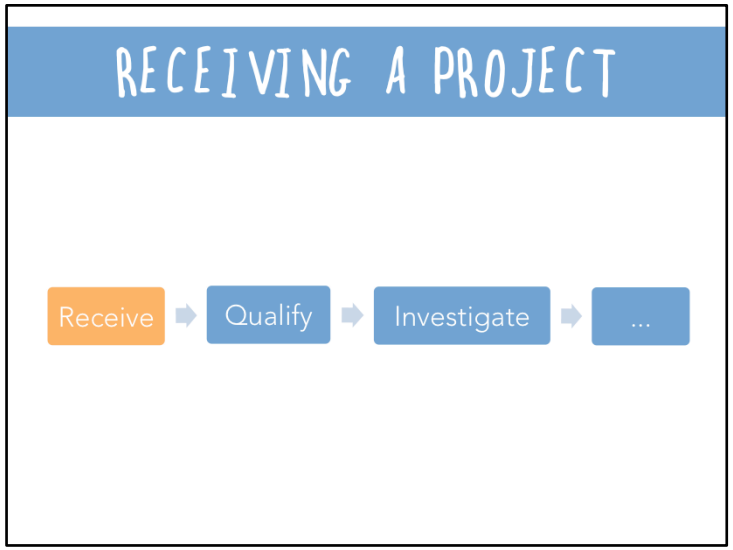


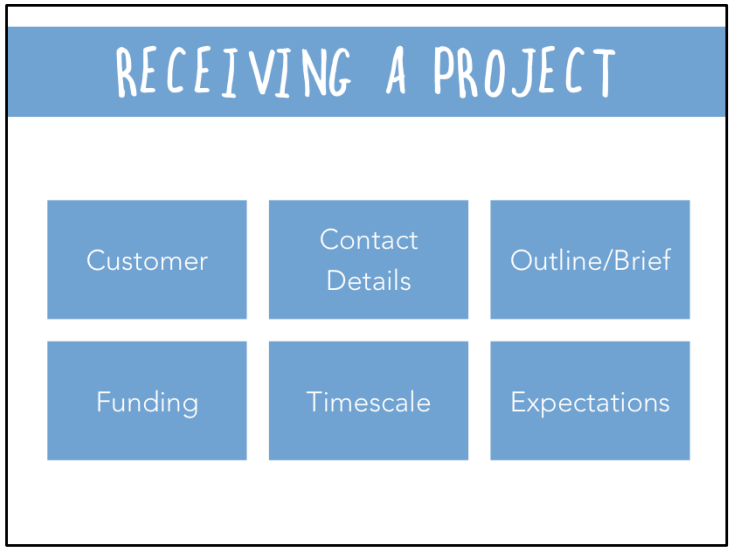
In this video we will look at the process of starting a software development project.



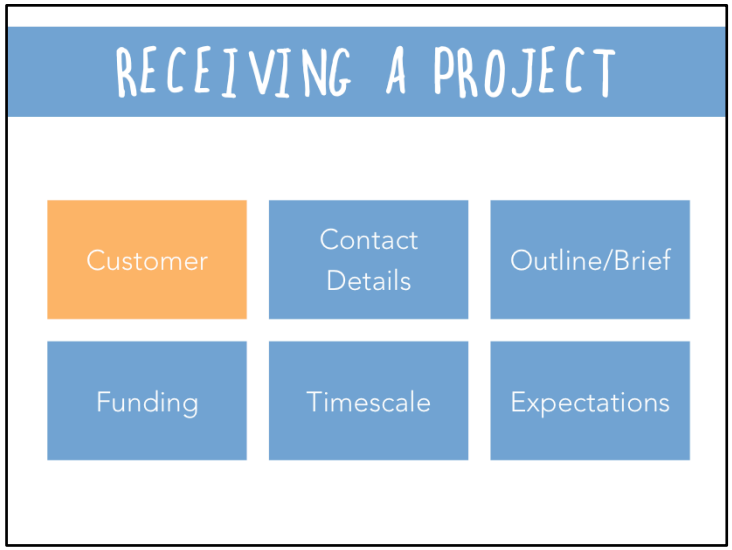
We will consider the first three tasks in the process used by epiGenesys.



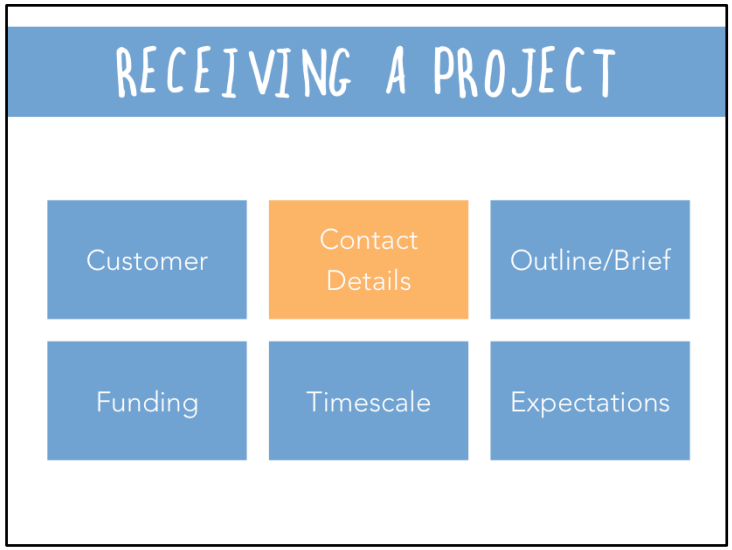
The first task is to receive an initial contact from a prospective customer. At epiGenesys this would usually involve an email or telephone call.



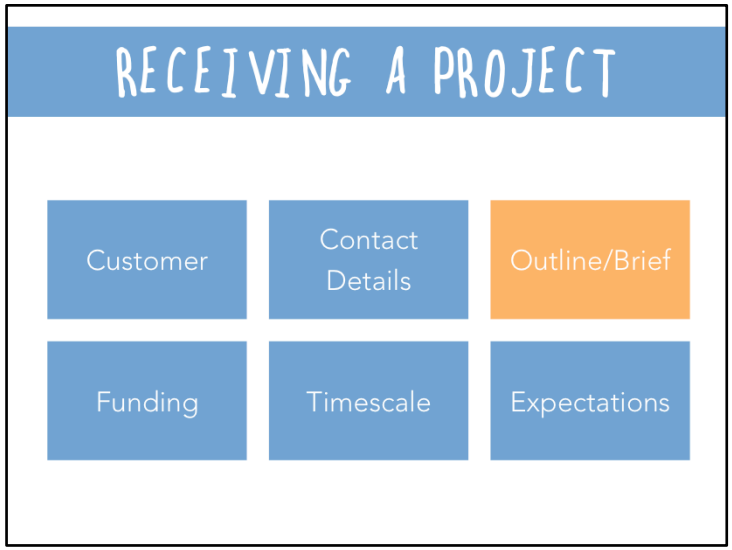
Upon receiving an initial contact there are six key pieces of information to collect.



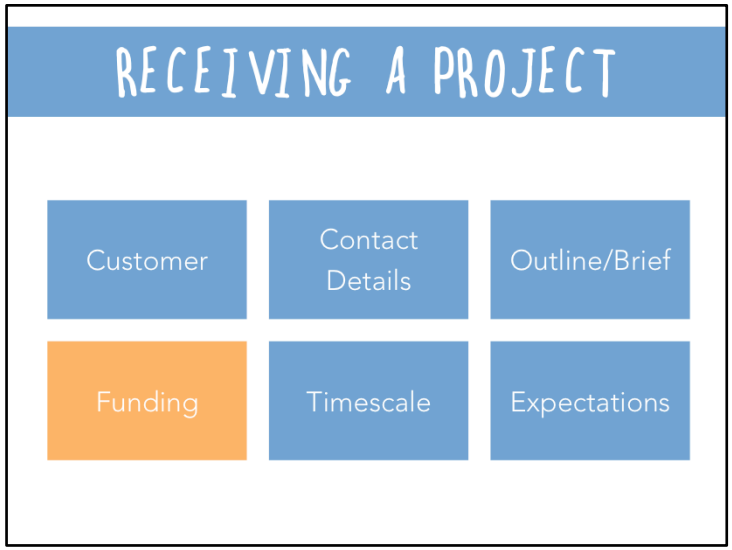
Who is the customer? Are they an individual or an organisation? Are they commercial or not-for-profit? Are they a small or large? Where are they based?



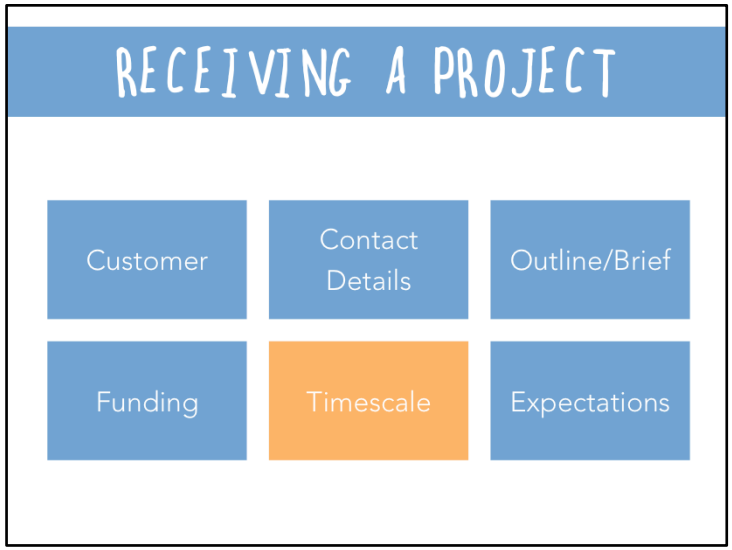
What is the best method to contact the customer? Collect email addresses, telephone numbers and other contact details as appropriate.



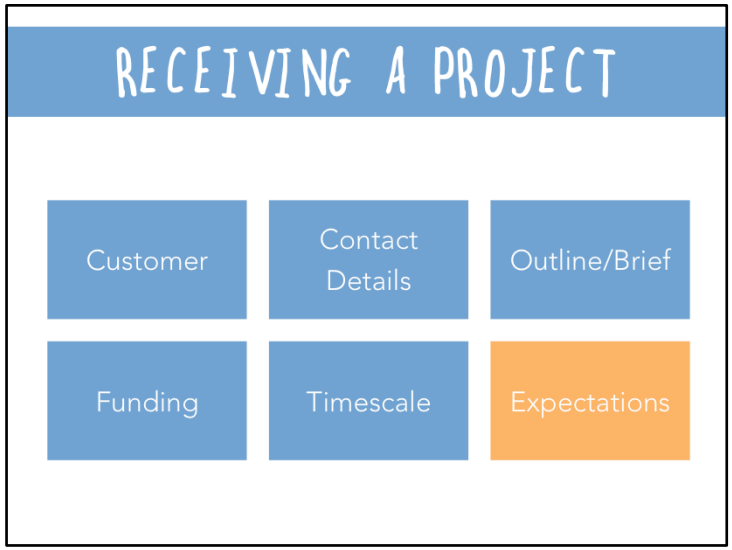
What is the project outline? Does the customer require software development or some sort of consultancy? Is it a web application, a mobile application, or something else? Is this for internal use, or for use with their own customers? Does the customer have a more detailed project brief that they could share?



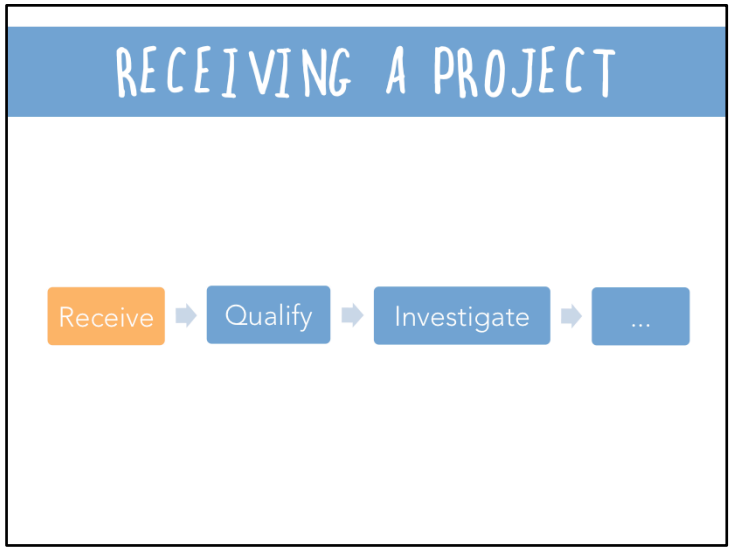
Is funding available for the project? Has this already been confirmed, or does a bid need to be submitted? Does the customer have an estimated budget that they could share?



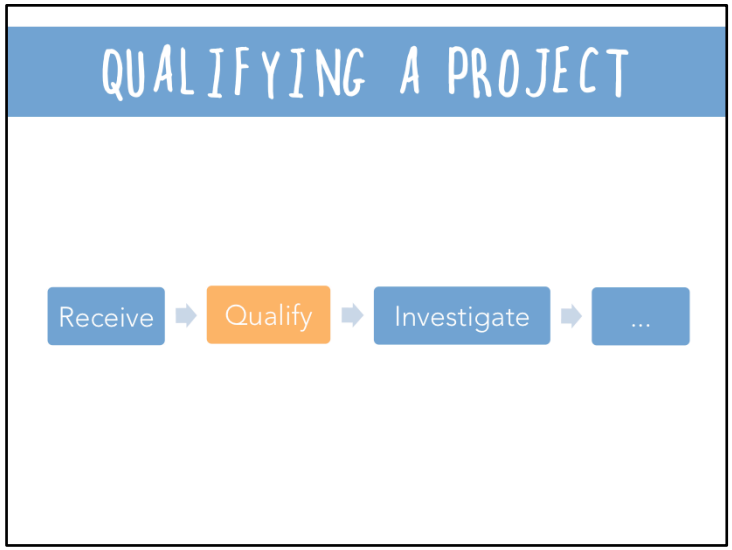
What is the timescale for the project? Is this fixed or flexible? Is there an event that defines the deadline, for example the start/end of a financial year or similar?



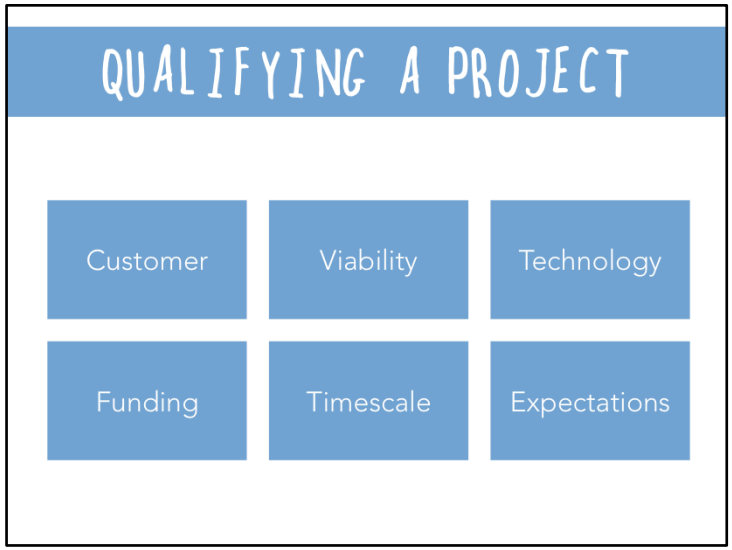
What expectations does the customer currently have? Describe the next steps to the customer – we will share the information provided with other members of the team, and after discussion we will contact the customer to let them know if we can help, and to arrange an investigatory meeting if appropriate. Provide an approximate date by which we will contact them. Is this satisfactory for the customer? Do they have any other specific requirements in order to proceed, for example is there a formal tendering process to follow?



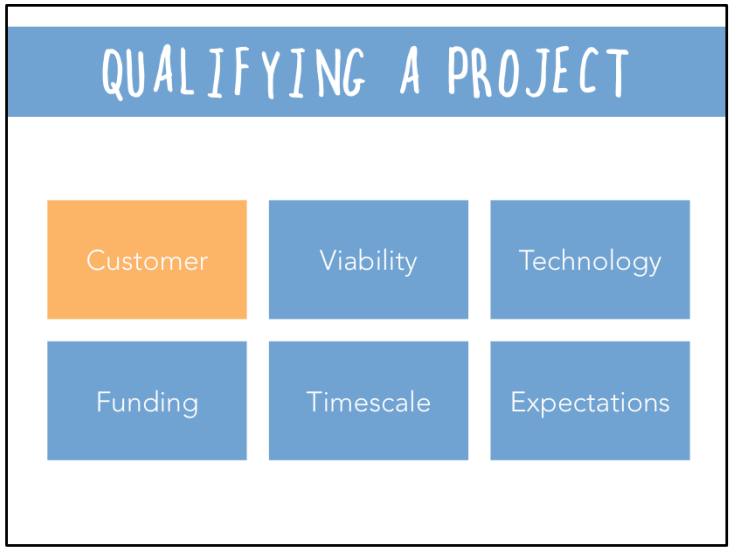
For the purposes of the Genesys modules, the staff at epiGenesys will typically have already received projects on your behalf during the summer, and collected the necessary information. However, it is useful to be familiar with this stage of the process should your customer or their colleagues approach you about additional projects.



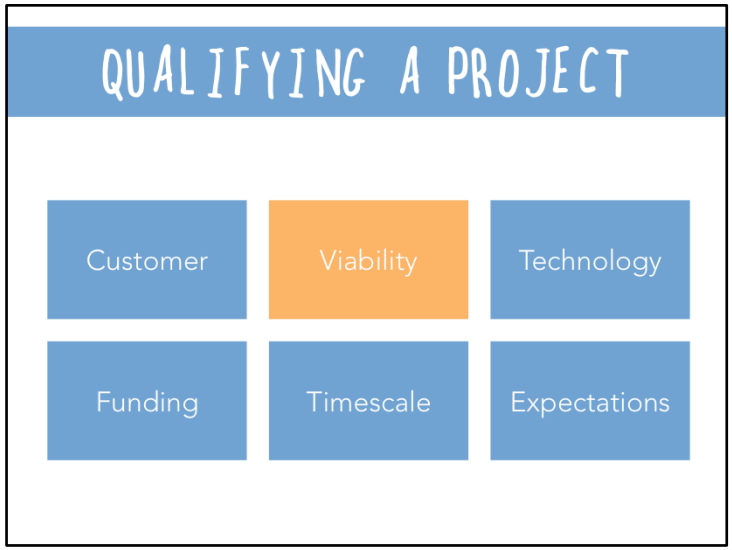
The next task is to qualify the project, by which we mean, to determine whether the project is suitable for us to work on.



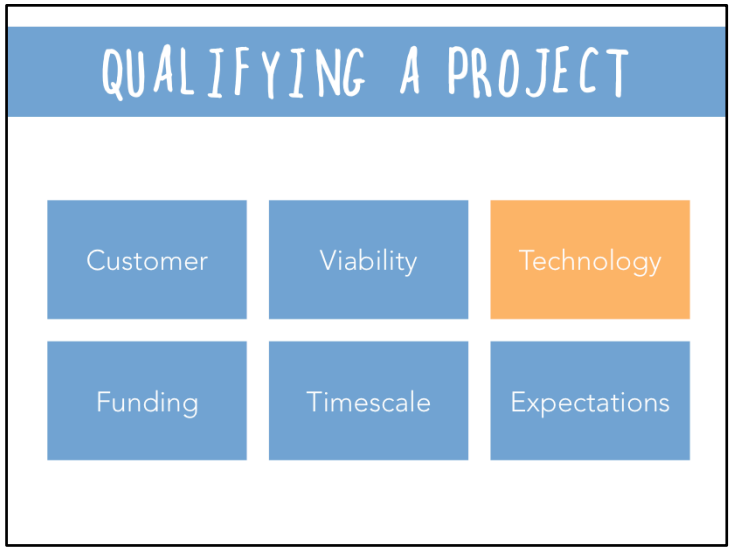
When qualifying a project there are six key issues to consider. These are of course closely related to the information collected when receiving the project.



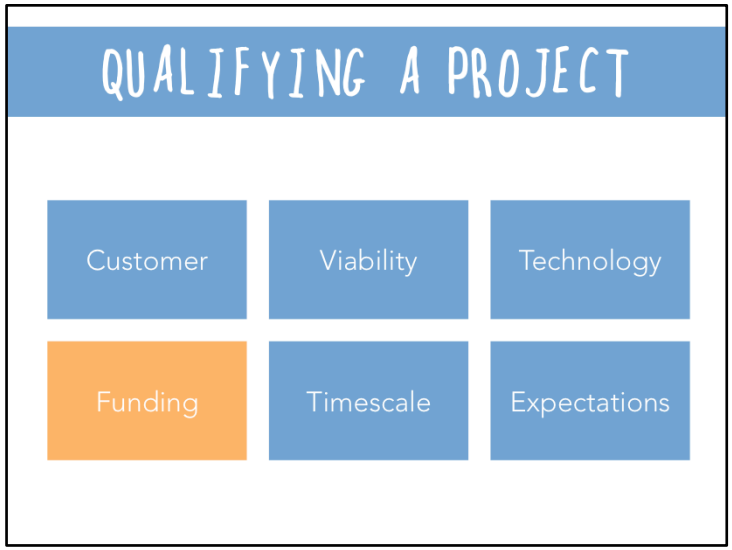
Do we think that we will be able to have an effective working relationship with the customer during the project? Does the customer have enough time to work with us to discuss requirements and test demonstrations? Is the customer based in a location that will make regular face-to-face meetings possible, or will we need to rely heavily on remote communication tools?



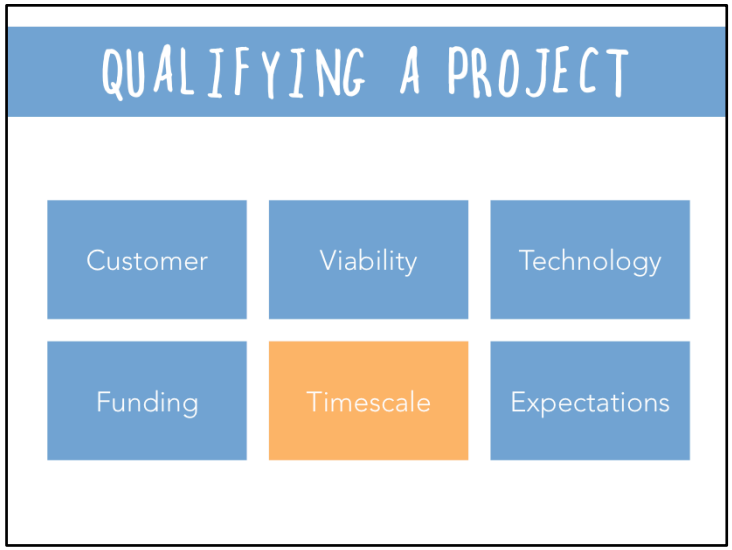
Is the project viable? Does the customer have a reasonable understanding of what they need? Are we able to provide the service the customer needs? Is the project excessively challenging such that it constitutes an unacceptable level of risk?



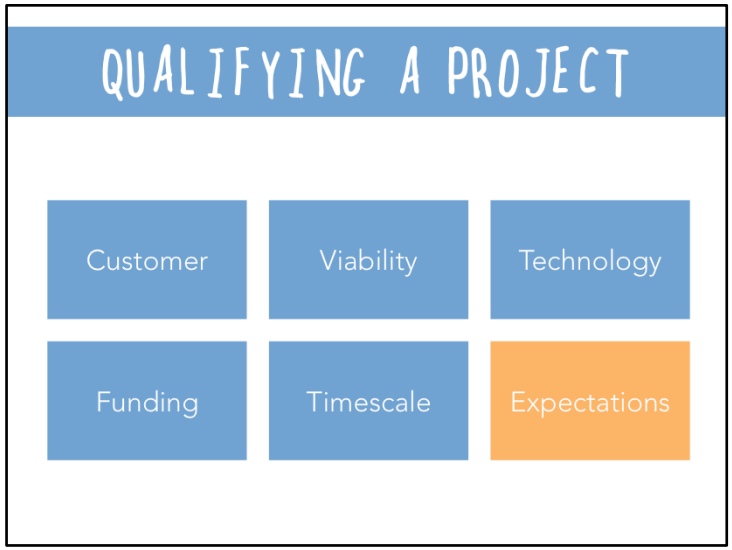
Is the customer satisfied with our preferred technology choices? Are we able to work with any technology required by the customer?



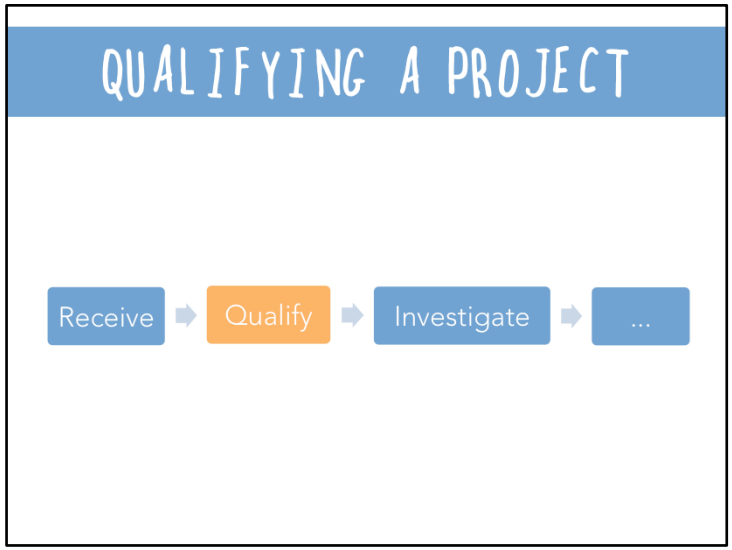
Are we happy with the funding arrangements for the project? Is the budget sufficient to satisfy the requirements of the project? Is there excessive risk that the budget will be insufficient?



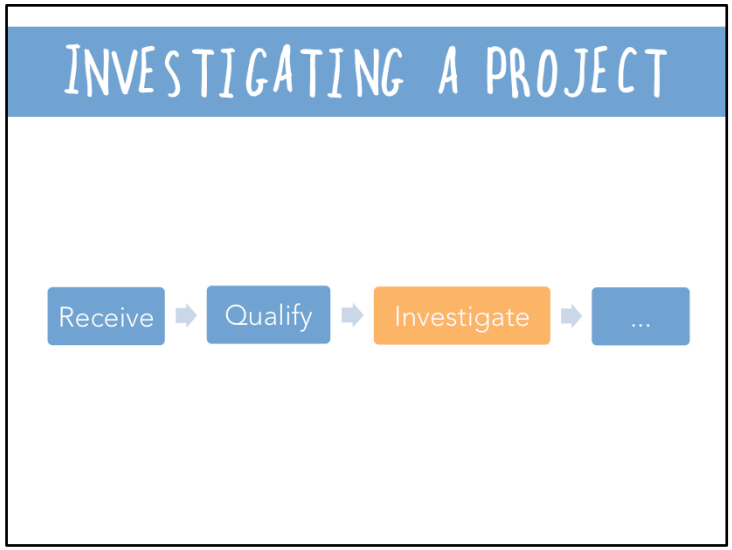
Are we happy with the timescale for the project? Will the timescale conflict with other projects or commitments in our schedule? Is the timescale sufficient to satisfy the requirements of the project? Is there excessive risk that the timescale will be insufficient?



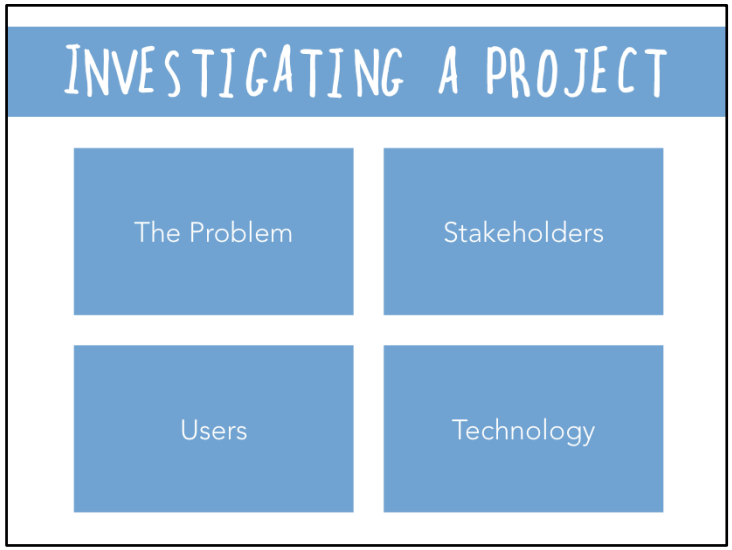
Does the customer appear satisfied that we can meet their expectations? Are we happy that the expectations of the customer are reasonable and can be satisfied? Are there any other risks we should consider?



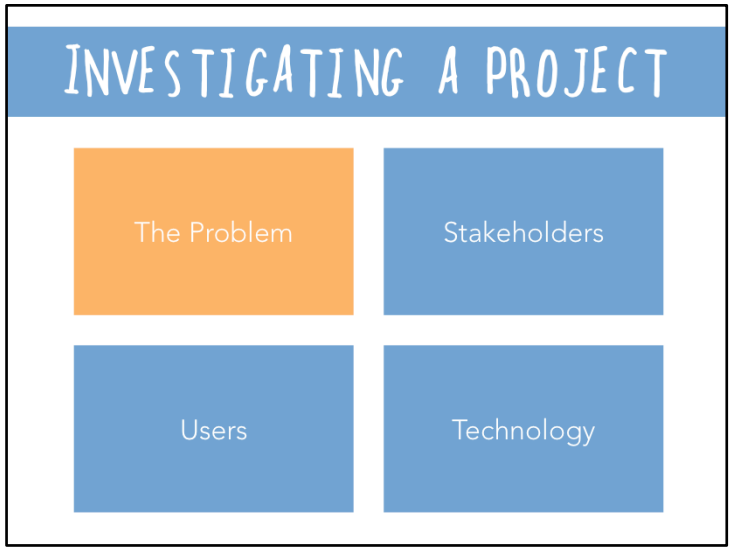
Again, for the purposes of the Genesys modules, the staff at epiGenesys will typically have already qualified projects on your behalf during the summer, and determined that they are suitable for you to work on. However, it is useful to be familiar with this stage of the process so that you can consider whether a particular project is suitable for your team to work on. For example, is there a technology required that you are not familiar with, and that you would not be able to learn in the required timescale?



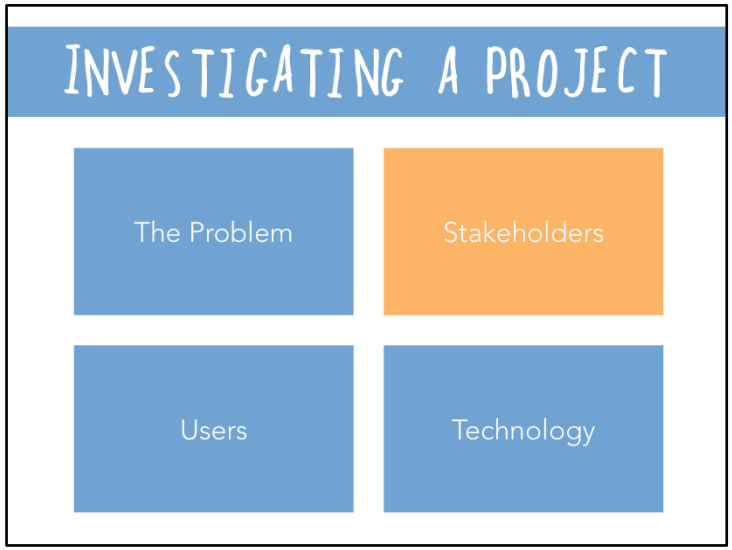
The next task is to investigate the project. This is the means by which we collect sufficient information to enable us to define the requirements of a software system. Ideally the customer will have already provided a reasonably detailed project brief, although in some cases only a short outline will be available. In either case you need to begin by exploring the available information with your Genesys team.



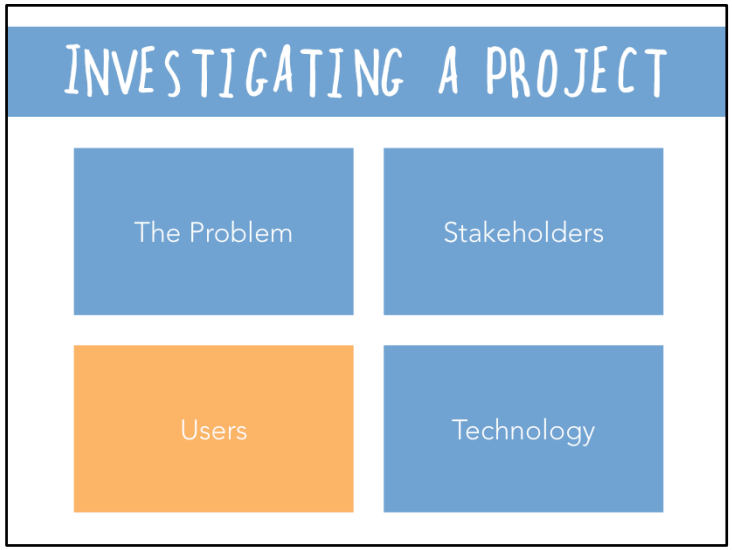
There are four key issues to consider at the start of an investigation. Discussing these with your team will help you to identify what you do not yet know about the project. You can then use this knowledge to prepare for further discussion with your customer.



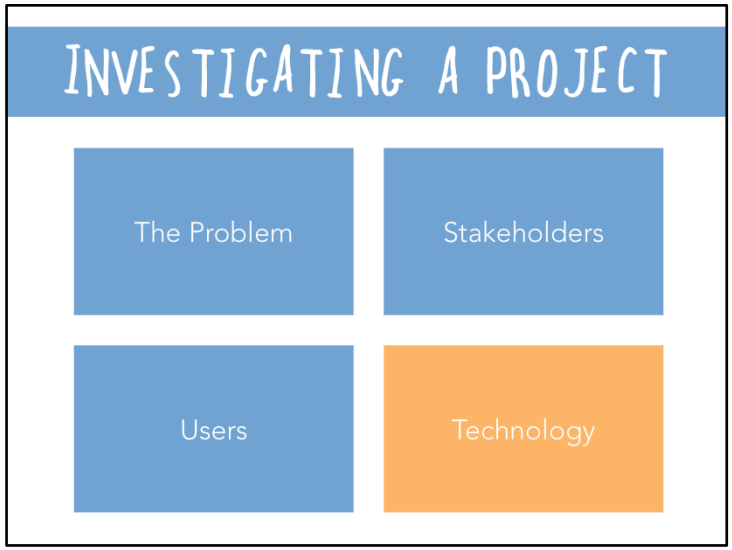
What is the problem that you are required to solve during the project? The problem might be described as the thing that your customer wants to be able to do that they cannot already do. Many projects involve replacing an existing process, which may currently be completed manually or may already make use of one or more software systems. The problem in such projects is rarely that the existing process needs to be replaced (i.e. we need you to make software to automate this). It is more likely that the problem is the existing process results in many errors which create additional work for staff (i.e. we need you to make software that has lots of built-in validation to prevent errors); or perhaps the problem is the existing process is slow and causes delays for customers (i.e. we need you to make software that allows us to spend less time entering data and more time processing orders). A clear definition of the problem that you are required to solve will help guide your decisions throughout the project as you consider what software functionality might be needed, how important it is compared to other functionality, and how it should operate. Without such a definition it is easy to create a software system that simply replicates the existing process and fails to improve it, or even makes it worse. It is also important to note that you may be required to solve several related problems during the project.



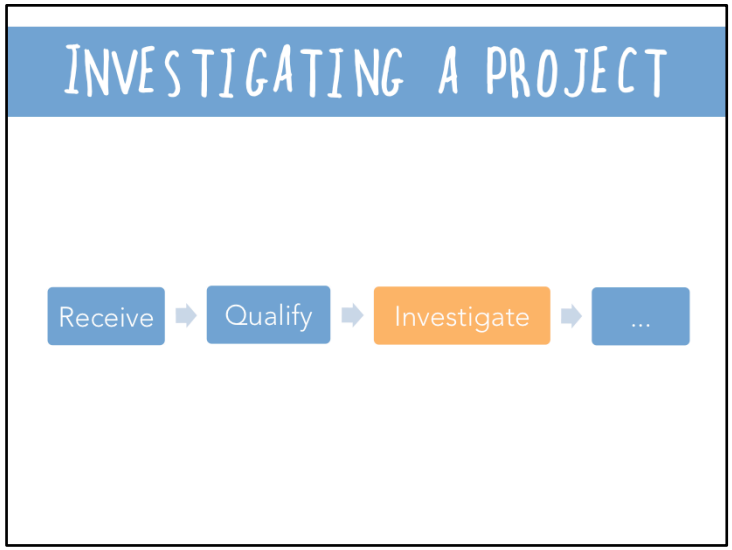
Who are the project stakeholders and what do they need? These may be at any level of the customer organisation, and may not necessarily be the people you are in direct contact with. For example, there may be senior managers who have an interest in how any software system you implement affects the performance of their organisation, or junior staff whose working practices will be affected by the software system. There may also be external stakeholders who need to be considered. For example, a separate organisation that has a dependency on the information that will be provided by your software system. You need to identify all stakeholders so they can be properly consulted during the investigation process, whether directly or via your customer.



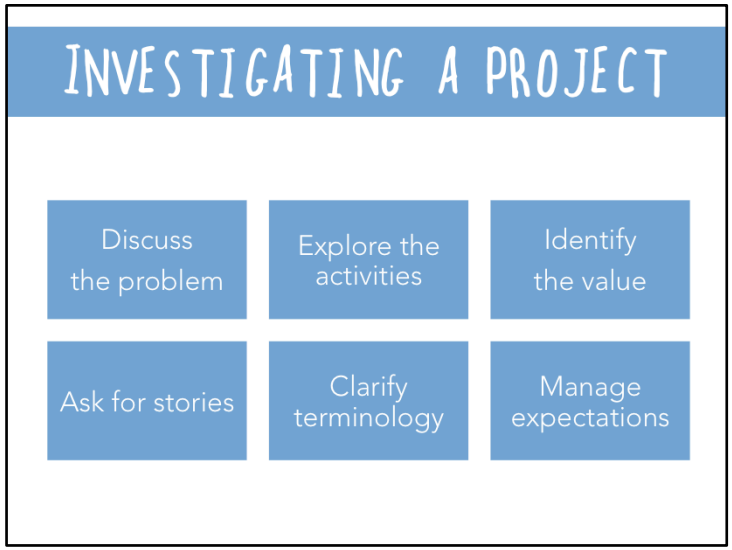
Who will use the software system to be implemented during the project and what do they need? Again, this may not necessarily be only the people you are in direct contact with. The system may also be used by others in the organisation, or by the customers of your customer. Identifying all users at the earliest opportunity will help in ensuring that you have not failed to consider vital aspects of the system. Ideally you should consult directly with all types of user to ensure you have captured their requirements. You should also consider how you might design the system to work in a way that is suitable for those users, and whether it will be possible to test the system with real users to confirm that the design meets their needs.



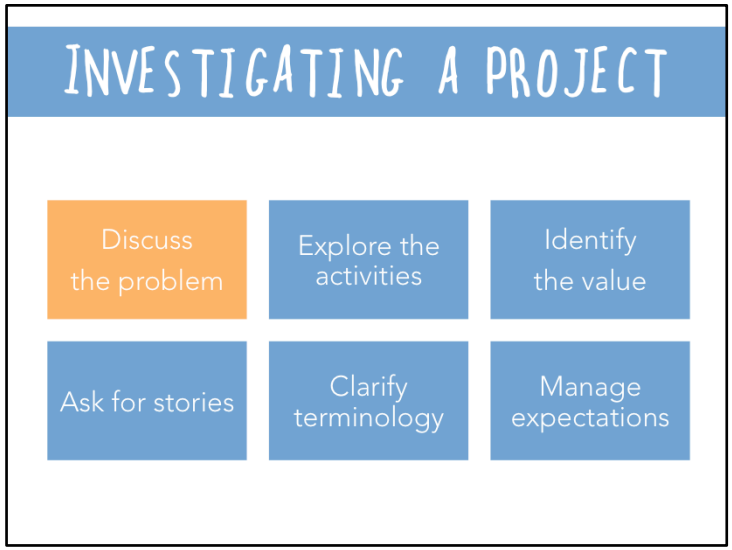
Does the project involve particular technology constraints? For example, does a software system need to interact with other systems, or meet particular standards to be used within an organisation. At this stage there may be no need to consider in great detail exactly how these constraints will be satisfied, but it is important to check whether they might extend the time needed to implement other aspects of the software system. You should also identify any technical contacts who will be able to provide you with further information, particularly so that you can contact them early and find out if there are any constraints on their availability during your project. If you are working on a project for a department in the University of Sheffield you will typically find that the staff at epiGenesys are the most suitable technical contact; you should not contact CiCS directly without first seeking their advice.



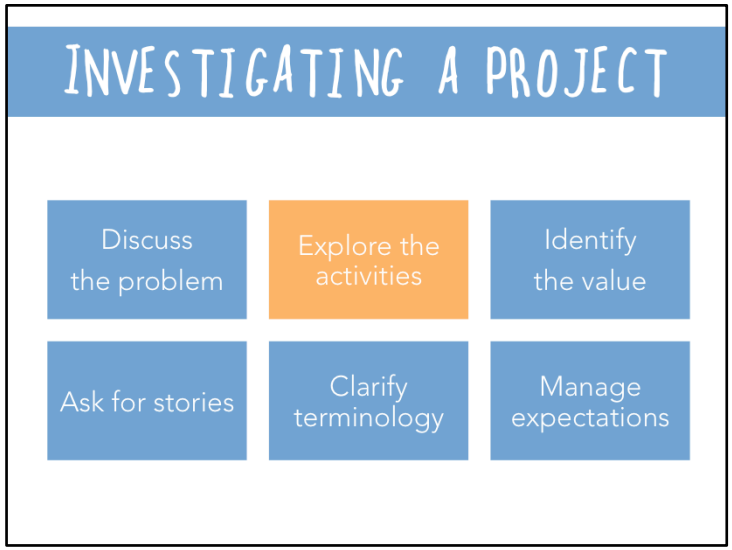
The next step in investigating the project is further discussion with the customer. We strongly recommend that this is done by holding a face-to-face meeting. Where this is not possible you should consider video calls before telephone calls or emails. Body language and facial expressions can be very helpful indicators when you are obtaining information from a customer.



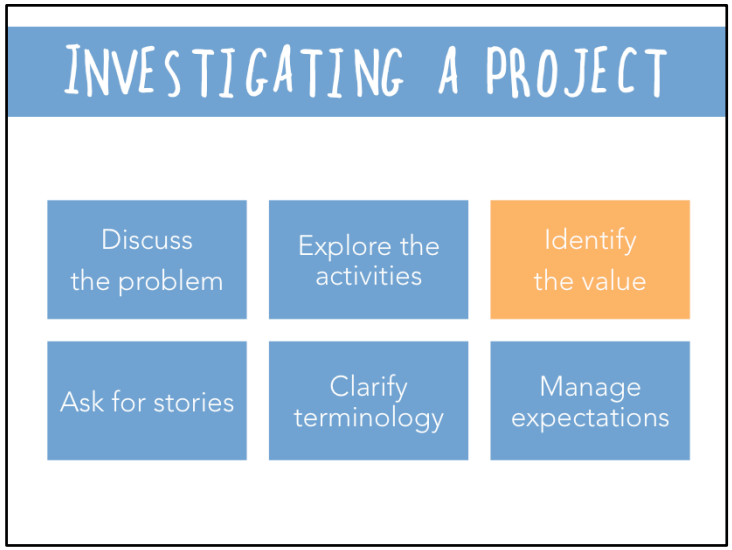
Here are six key things to do that can help you to obtain information from your customer.



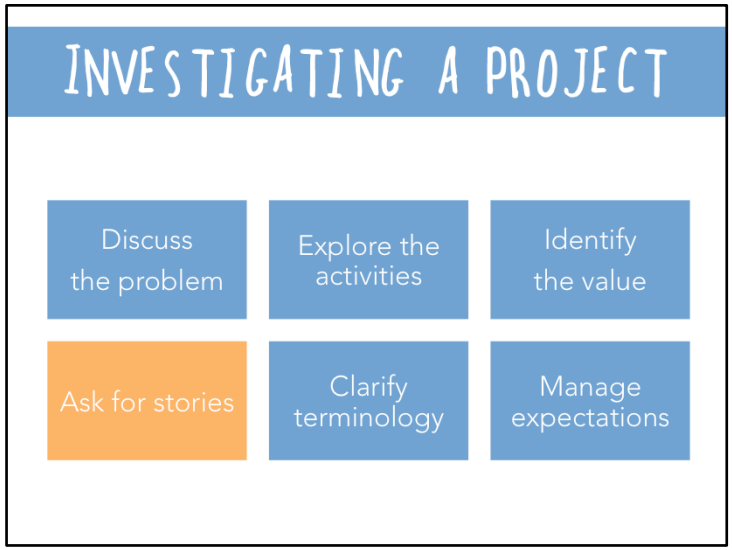
Make sure you ask the customer about the problem, or problems, they need you to solve. What is it that they need to be able to do that they cannot already do? How are they dealing with the problem currently? Are they or their colleagues currently using workarounds to get things done?



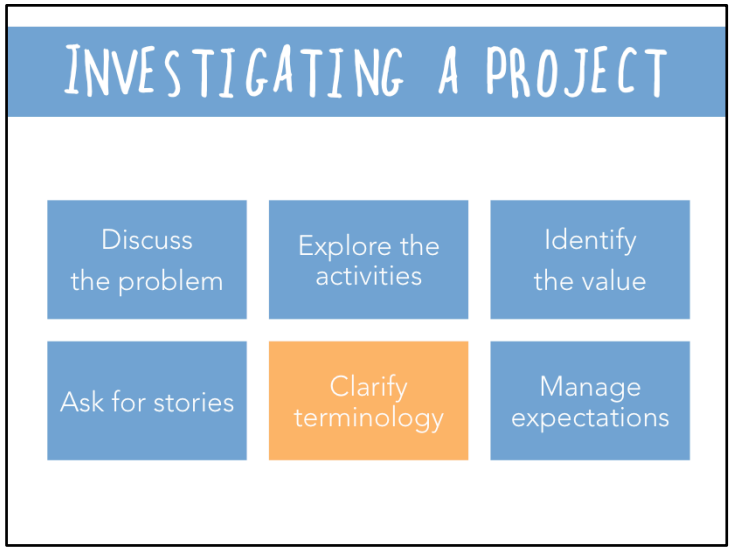
Find out what any software system you might implement has to do, by exploring the customer's activities. Where does a software system fit into a wider process for the customer? What are the key activities in the process that need to be supported by software? How do these activities work – what are their inputs, what processing needs to happen, what are their outputs? Who completes these activities and what are they trying to achieve? If there is an existing process, is there anything about it that the people involved particularly like or dislike? The phrase 'can you tell me more about…' can be useful to lead your customer into an open discussion about a particular activity. For example, 'can you tell me more about placing an order'.



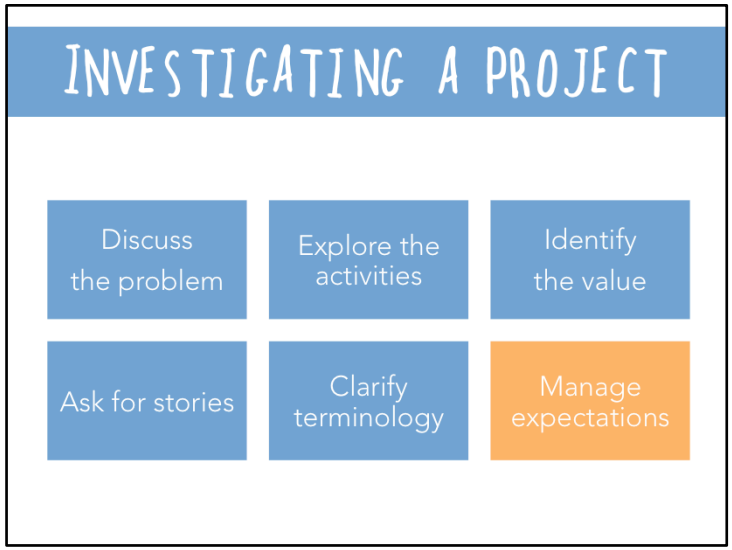
Find out why activities need to be done, by identifying the value they bring to the customer. Do they produce outputs that are valuable to the customer? Is there a legal requirement for certain activities? Are there any activities that only support other activities and might not be necessary in a revised process?



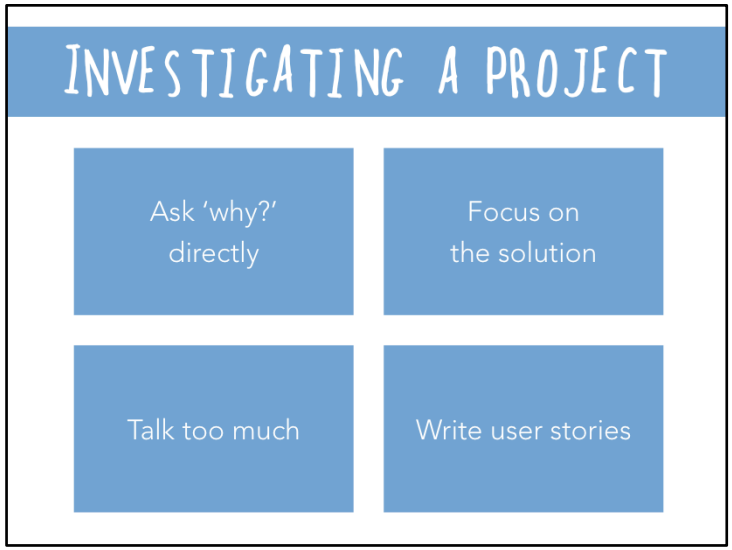
If you are replacing an existing process, ask the customer to tell you stories about things they have to do. Particularly look for stories about times when the process did not work correctly, as this will help you identify discrepancies between the ideal process they are likely to have described to you, and how things actually work in practice. The phrase ‘tell me about the last time…’ can be useful to help direct the customer to recall useful scenarios. For example, ‘tell me about the last time you had to correct a mistake in an order’.



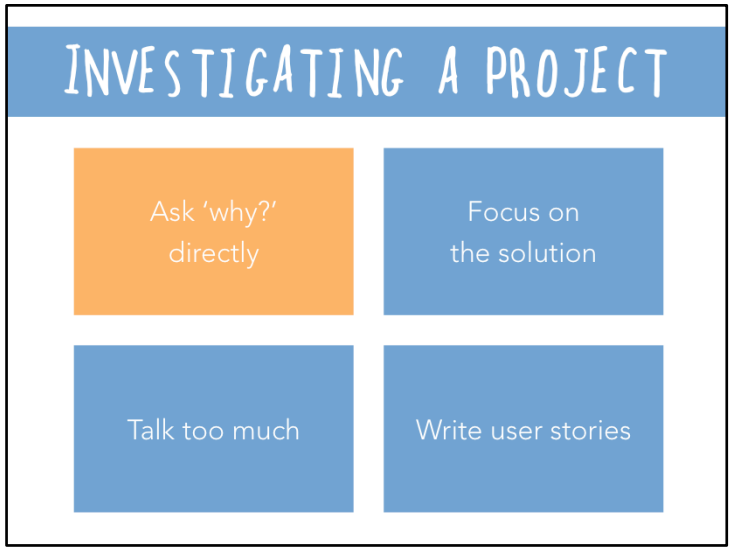
Do not assume the customer uses the same terminology as you, and be sure that you understand any specialist terms by asking for clarification. The phrase ‘can I check what you mean by…’ can be useful to invite the customer to explain further. For example, ‘can I check what you mean by “an order” – is it a request for one item or several items, is it different to a pre-order, is there anything else interesting about an order?’.



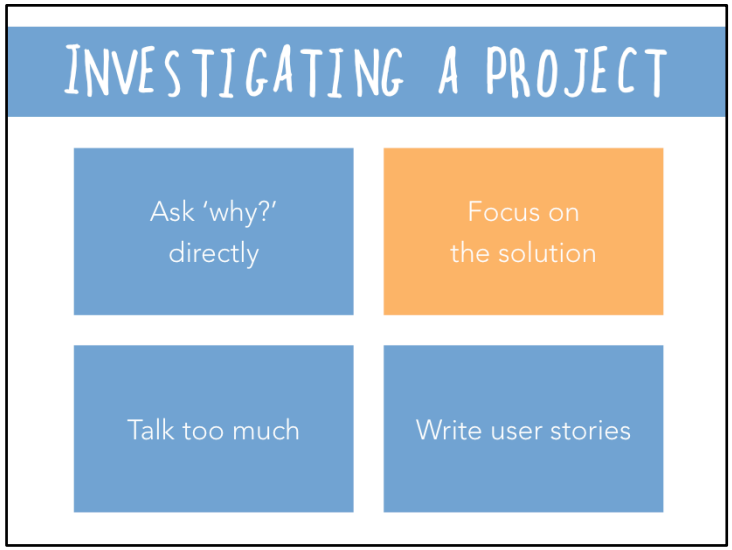
Throughout customer meetings be careful to manage expectations. Make clear that you are exploring every aspect of the project so that you understand how a software system would need to operate, but that the scope of the project will be discussed later. If the customer is concerned ask if there are any key aspects they consider essential to the success of the project that they would like to share with you. Do not feel pressured to provide an estimate of the size or cost of the project at this stage, and if prompted explain that you cannot yet provide useful information but will do so as soon as you are able. If the customer is concerned ask if there is a particular timescale or budget constraint that they would like you to consider.



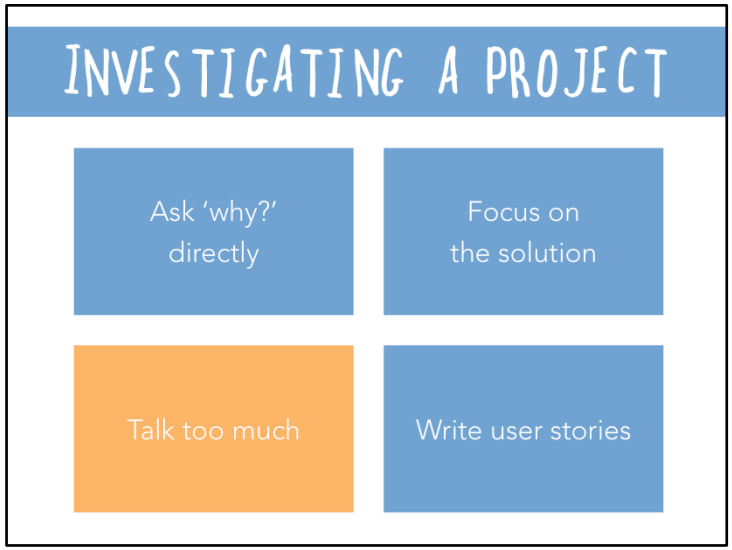
Here are four key things to avoid that can make it difficult to obtain information from your customer.



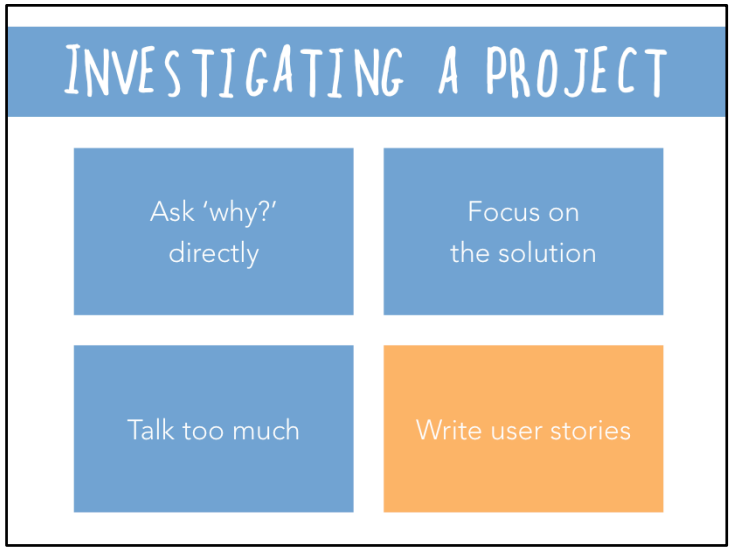
Try to avoid directly asking your customer 'why?'. This question is often difficult to answer; your customer may never have needed to explain the reason for a part of their process before, but will likely feel that they are expected to know this information. Simply asking 'why?' may cause your customer to become defensive in their responses. Instead try to ask more carefully directed questions, for example, 'what goal does this activity help you to achieve?', or 'who in your organisation would consider this activity to be important?'.



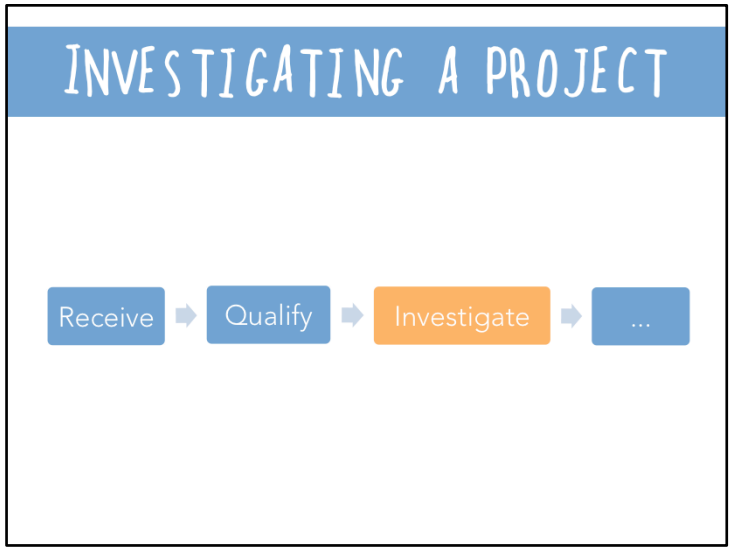
At this stage you should avoid discussing the solution you might provide. Doing so can encourage you and your customer to make false assumptions about what is required. If you suggest features that your solution could provide your customer will likely agree that they are useful, even if they are unnecessary, simply to be polite. Instead keep your focus on the problem, activities and value involved in the customer's process; if these are things that already exist they are much easier to discuss analytically than an invented feature. At this stage you do not need to sell your solution, but rather your intention to ensure you fully understand what the customer needs.



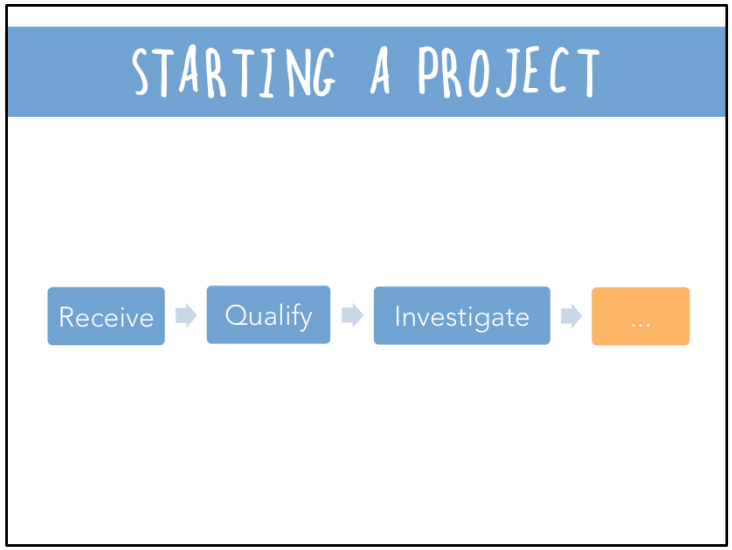
Avoid the temptation to talk too much. Instead your goal in a discussion with the customer should be to direct the conversation, but to spend most of your time listening. It is helpful to occasionally pause, even if the silence seems uncomfortable, to provide everyone with an opportunity to reflect on what has been said, and to give your customer an opportunity to volunteer additional information that they might think is useful.



It is generally not helpful to write user stories during a customer meeting. Although it may seem more productive to capture information in this format during the discussion, it can be an unnecessary distraction that causes you to miss key insights. Instead we recommend that you take notes using pen and paper, which also allows you the freedom to quickly draw rough diagrams and link key points. The user stories can then be written with your team after the meeting, which also gives you an opportunity to discuss your thoughts in more depth whilst doing so.



The investigation stage is typically where you will commence work on a project with your Genesys team. The staff at epiGenesys will of course be available should you need advice.



Having completed the investigation of your project you will then need to use this information to write user stories and validate them with your customer, estimate the work needed to implement the necessary software system and consider how to price this, work with the customer to prioritise user stories, and finally prepare a formal project proposal. If the proposal is accepted a contract may need to be prepared before software development commences. All of these activities will be covered in subsequent videos.

Here is some recommended reading aimed at people who are interviewing customers during the process of creating a startup; many of the techniques described are also useful when discussing the requirements of a software development project with a customer.