The Top Ten Problems in Siebel Projects ...

... and How to Avoid Them

By Duncan Scattergood, Customer Systems plc

Organisations often spend huge sums on buying and implementing Siebel, and many see a fantastic return on their investment - increasing sales, improving customer service and reducing internal costs. However, others struggle and often become disillusioned with the whole process after years of effort. Why do Siebel projects go off track, and how can organisations ensure that theirs will be a success?

Siebel is an immensely powerful product with an extremely wide set of functional capabilities. However, it is the way in which the product is implemented that determines whether the anticipated benefits are delivered.

A project can fail for a variety of reasons. During our ten years working with Siebel we have been called on to rescue many projects that have gone off track, and time and time again the cause can be traced to a basic set of problems.

By bringing together the ten reasons why a Siebel project is most likely to fail, this article examines the factors that separate success from failure. We explore potential pitfalls and how these can be avoided, and consider how to bring problematic projects back on track.

The Top 10

1 Unrealistic short term expectations

One reason why a Siebel project may be perceived to fail is that there is an unrealistic expectation of how quickly the benefits will be seen.

Merely installing Siebel and making it available to users will not transform the effectiveness of an organisation. Instead, realising the full capabilities of the application takes a lot of hard graft from technical and business oriented teams. Time and effort are essential, and not all the benefits will be delivered at once.

For example, if a key benefit is more accurate forecasting of sales opportunities, that benefit will not be realised until the organisation's opportunities have been actively managed through the new process and there is consistency in the data.

To avoid setting unrealistic expectations, care must be taken to understand at the outset:

- The actual IT costs of the project
- The anticipated benefits and when they are expected to be delivered
- The prerequisite process steps, both technical and business, to obtaining the benefits
- The necessary commitment by the business of time and resources to the project

If adequate consideration is given to these points when developing a project plan, it is possible to celebrate milestones along the way to the ultimate goal, rather than becoming disillusioned by how much work has been done and yet how far there still seems to be to go.

> "Time and effort are essential, and not all the benefits will be delivered at once."

2 Insufficient Siebel knowledge in design team

Experts in the relevant business processes, or experienced business analysts who have worked on other applications, can play vital roles in the analysis and design team, but they must be complemented by real Siebel experts. A lack of Siebel expertise is likely to give rise to the following scenarios:

- Processes are designed based on how non-Siebel applications work, leading to many screens being required for something which could be achieved with one or two button clicks. This leads to significant and often unnecessary re-engineering of Siebel.
- Particular ways of working are imposed on the application, when Siebel already supports the exact requirement in a slightly different way. Recreating functionality that exists in the core product is unnecessary and expensive, both in terms of initial development, and of ongoing maintenance, enhancements and upgrades.
- The system fails to meet all the requirements because it is perceived that certain things cannot be achieved in Siebel when in fact, with appropriate skill, they are fairly small configuration tasks.

The importance of engaging real Siebel experts at the early stages of a project, with both functional and technical knowledge about what is and is not achievable – and how best to achieve it – cannot be overstated.

3 Separate technical team

In a traditional IT project, a business analyst speaks to users in order to gather requirements, before documenting them and passing them to a technical designer. The designer translates them into a technical design which is then implemented by a programmer before the system is tested and deployed. This approach results in a large gap between the users with the original business problem, and the technical team charged with solving it. In a more extreme version, the development work may even be carried out off-shore.

This model may still have a place in some situations but it is not appropriate in a Siebel project. Siebel is different because a skilled configuration expert can bend the core application to meet a particular requirement very quickly. This presents an opportunity to run projects in a way that greatly increases the probability of the delivered system meeting users' needs, while lowering project costs by reducing the project team size.

> "Siebel is different because a skilled configuration expert can bend the core application to meet a particular requirement very quickly."

The ideal model for a Siebel project is a small, expert team where each individual is responsible for all activities relating to a particular functional area, from requirements gathering, through design, development and testing, to deployment. In larger projects, there may well be people who specialise in design, build or test, or people who are involved in only one area, but the core team should remain throughout the whole project.

4 Insufficient user involvement

Ultimately, a Siebel implementation is a business project that must deliver business benefits. The project is for end users and their managers, so it is critical that they are fully engaged throughout the project to ensure that the final system delivers what they need.

Often, other responsibilities mean that users have only limited time to give to the project, but if they are only prepared to commit very small quantities of time – or even no time at all – then our experience is that a project will not be as successful as it should be.

Typically, the most important stages for significant user involvement are:

- In the design phase, including workshops and prototyping to verify that the user interface will support what they need to do. At this stage, it is also important to run through each of the processes that users will perform, to check that nothing is missing.
- In the user acceptance test phase, to check that what was finally built really does do the job.
- In the deployment phase, to assist in rolling the system and associated process changes out into the wider user community.

In addition, continual (if less frequent) involvement throughout the project, including build, is advisable. This involvement really does help users to have a strong sense of ownership of the project, as well as allowing frequent checks that the project is on track, avoiding surprises during user acceptance testing.

When planning user involvement, a key consideration is who to involve. This varies from organisation to organisation, and also depends on the task to be performed. Of all the user-centric tasks, design workshops are normally the most critical to a project's success. Senior management often do not have enough time to do the activity justice and may not actually understand the finer detail of the processes that users will perform on a day to day basis. On the other hand, running workshops with low level users can result in the collection of information that reflects the details of the current system rather than the changes that senior management are trying to implement. Therefore, the best candidates for these types of activities are often found at team leader or middle management level.

5 Poorly thought-out integration

Many Siebel implementations involve complex integration with other applications. Even if each individual application that supports a business process is well designed and robust, if the interfaces between those applications are flawed, then the resulting overall system will not be solid.

Again, expert Siebel knowledge is essential in this scenario. If the team handling the integration has little Siebel knowledge, the following often occur:

- Inappropriate split of tasks between applications: for example, something that could be easily achieved in Siebel is customised into another application because no one knew better.
- Unresolved inconsistencies between data models: for example, a contact can have many addresses in Siebel, but only one is supported in a particular legacy application.
- Incorrect interface type selection: the wrong decision is taken over which integration technology (EIM, eAI, Web Services etc) to use. Integration routines may also be poorly developed, since this is primarily a Siebel configuration job requiring expert Siebel skills.

Even when Siebel expertise is engaged in the design and build of integration, organisations often choose to subdivide the Siebel team along the line between configuration and integration. This is a mistake because decisions made at the design stage about how the user interface should work can affect the format, style and frequency of integration – and choices made in integration (often forced by legacy applications) can affect how information should be presented to users. Therefore, the two aspects are by no means independent, and as a minimum, the team building the core configuration should be expert in the issues surrounding integration and vice-versa.

Integration must be considered in the context of the overall application design, and not dealt with as an afterthought. Of course, individuals with high levels of expertise in the other applications concerned should also be involved, and collaboration is the key to successful integration projects.

6 The difference between sales and service projects

A Siebel application is usually intended to be used by one of the following groups:

- Users selling high value products to a relatively small number of customers.
- Users dealing with low value, high volume sales and/or service type transactions.

A particularly common mistake resulting from a lack of Siebel expertise is the failure to recognise the significant difference between these two types of project.

Low value, high volume sales and service environments are typically very transactional with emphasis on users gathering set pieces of information in a set sequence, and the style of user interface built needs to reflect this.

By contrast, the nature of transactions in a high value, low volume scenario means that processes are typically quite unstructured, with skeletal information about new opportunities being fleshed out throughout the sales process. It is therefore difficult to predict the sequence in which users will navigate through Siebel and enter information, and again the user interface must reflect this.

Furthermore, high value, low volume sales users can effectively boycott the application if they do not like it – organisations do not fire their top performing salespeople simply because they will not use Siebel.

Additionally in this situation, increased management control is often a key stated objective of the Siebel project. However, if this is the sole objective of the project then end users may feel that the system's only impact on their day-to-day activities will be a negative one. Consequently, user adoption will be poor and the project will fail.

> "The key is to encourage user adoption by offering genuine benefits for the sales force."

The key is to encourage user adoption by offering genuine benefits for the sales force, such as the provision of up to date competitor analysis, or automatic generation of presentations, proposals and quotations.

An application that is well designed for one type of scenario rarely suits the other. Misunderstanding the user types and consequently not designing the user interface correctly is an expensive mistake to rectify later, not only in terms of build cost but also in terms of lost user confidence when an inappropriate application is deployed.

7 Insufficient knowledge in development team

Configuring Siebel quickly and efficiently is not an impossible challenge but it is a specialist job. Whereas with many programming languages an expert might work two or even three times as fast as an average programmer, with Siebel development the factor is much bigger, maybe as large as 20.

A development team with an inadequate skill set risks delivering any of the following:

- An application that does not meet users' requirements because the people performing the configuration do not know what is possible or how to implement it.
- An application that was expensive to develop because it contains more configuration than necessary to achieve a given task, in particular extensive coding to implement something that could have been handled far more easily in Siebel Tools if the developer had had the relevant knowledge.
- Following on from the previous point, an application that is difficult and costly to support, enhance and upgrade.
- Unacceptable user response times due to inefficient configuration or poor user interface design.

As with many of the problems outlined in this article, the solution is to ensure an appropriate level of Siebel knowledge and skill is engaged at this stage of the project.

In the ideal project there would be no separation between the design team and the build team, and in the best projects the design team will simply be absorbed into the core of the build team.

8 Insufficient or inappropriate testing

It is not necessary to spend an excessive amount of time checking the standard features of the packaged Siebel application, which can usually be assumed to work correctly. Pre-deployment testing must instead focus on the specific configuration that has been carried out to meet the organisation's particular requirements. Ensuring that the application supports the business processes as intended, and that processes run correctly across multiple applications, is also vital.

The trick is to use the limited time period available for testing in the most effective way possible, concentrating on the areas of highest risk and/or those with the most impact if there is a problem. This usually means focusing on the following:

> "Ensuring that the application supports the business processes as intended is vital."

- Do the processes, as defined in the workshops, work as anticipated? If a good job has been done of involving users through the design and development process, this should not normally be a major area of concern.
- Are any exceptions to those processes supported?
- Do integrations work between applications?
- Is all the master data, e.g. list of values, correct? Do the initial data loads import all the data correctly?
- Will the application work correctly in the production environment? For example, the development environment might sit on a single LAN, but multiple firewalls might be involved in the production environment that require opening of ports and other infrastructure changes.
- Do any specialist modules, e.g. Siebel Remote, work as expected?
- Will the application work with production volumes? As it can be very difficult to replicate production usage scenarios, careful thought ought to be given to whether this style of testing is necessary and practical and, if so, what scenarios will be simulated.

Test plans are also advisable to ensure a degree of repeatability, although these should be as brief as possible to keep the emphasis on high quality testing, not on the maintenance of test plans.

9 Poor user adoption

It is not uncommon to come across Siebel implementations that are, in the main, technically sound and well aligned to user requirements, but are poorly adopted because the users do not understand how to take advantage of the facilities offered.

This lack of knowledge can and does occur at multiple levels:

- Not knowing how to perform basic navigation, search and data entry: for example, taking multiple actions to achieve something that can actually be done with a couple of button clicks.
- Not understanding how the application relates to the organisation's business process: for example, not knowing when to enter an opportunity in the system, or what stages it should go through. This can result in one user not completing all of the necessary steps in the system to allow the next user involved in the process to do their job.
- Incomplete or inappropriate data administration: for example, if an administrator fails to correctly set up the organisational structure within the application, a user could find that they are not presented with the subset of data they need to see in order to do their job.

> "...a key requirement is to extract data from the application and put it in a form that managers at all levels can use to make decisions."

One of the main causes of this is poor training. The training provided may simply be insufficient, or it may be inadequate in that it only covers the mechanics rather than the procedures specific to the organisation. Training that is not delivered very close to the point of system deployment, or training that does not reflect a person's prior technical experience, will also not yield the best results. For example, it would not be appropriate to use the same training material for a field sales person who has previously hardly used a computer, and a technically literate power user. However, both of these users still have training needs.

Surprisingly often, projects get as far as the end user training and deployment stage before it is discovered that the detailed procedures for using the system have not been fully agreed and documented. This lack of clarity is another reason why users do not use the system that has been developed. The normal cause of this is insufficient user involvement throughout the project and it is often one manifestation of more deep seated problems in an implementation.

To avoid delivering a system which users cannot use to its full potential, it is essential to thoroughly plan the way in which the application will be used, and to provide adequate training to users across the organisation.

10 Inadequate reporting

When designing and configuring a Siebel application, it is easy to focus entirely on the entry of data, and how that data should be transferred to other applications by integration.

However, in most implementations, a key requirement is to extract data from the application and put it in a form that managers at all levels can use to make decisions. In other words, the application must have the capability to report on and analyse information held in the database.

Oracle provides a number of reporting and analysis tools including Siebel Reports (Actuate), Oracle BI Publisher and Oracle Business Intelligence Enterprise Edition (Siebel Analytics). The most suitable tool depends on the situation, and it is often appropriate to meet multiple requirements with multiple technologies.

It is essential to consider reporting at the start of a project, to ensure that the information captured suits the reporting requirements, and to ensure that reporting and analysis are delivered either in parallel with, or very shortly after, the main Siebel releases. If any of these steps are omitted, an otherwise excellent Siebel implementation can elicit a reaction of "so what?" from the business community.

The common theme of product knowledge

The majority of the problems outlined here are underpinned by a common theme. If the project team does not understand Siebel well enough from a functional and/or technical perspective then it is very unlikely that the system will be well aligned to users' needs, of reasonable technical quality and cheap to create.

Product knowledge cannot be measured in years of experience. It is quite possible for an individual to have worked with Siebel for several years without actually having learnt a great deal about the product.
In fact, true product knowledge encompasses three key things:

- A thorough understanding of what the core product does and how that can be applied to particular business situations.
- A deep comprehension of the principles behind how the product is built. It is not possible to hold the entirety of such a vast product inside one person's head, but it is perfectly possible to have a firm enough understanding of the underlying technologies to allow any new functional area to be learnt very quickly.
- An understanding of how to gently bend the core product to any given business situation and, in particular, which of the multitude of available configuration techniques is most applicable in which situation.

Projects may have effective team members who have either a strong functional or technical emphasis in their skill set, but a truly worthwhile team member will have knowledge about both dimensions.

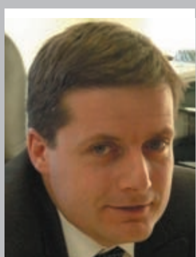
In our experience, the most successful Siebel projects have a small team of experts who can both speak to users about requirements and effectively configure Siebel to meet those requirements. The value of real product knowledge (as opposed to certificates or years of poor quality experience) cannot be overestimated.

In conclusion

A Siebel project has the potential to increase an organisation's sales, improve customer service and reduce internal costs. However, without adequate effort, planning and expertise, the project can run into a number of problems, and ultimately fail to deliver on expectations.

To achieve success, it is essential that your project team is equipped with the skills and resources, both internal and external, that they need to maximise the benefits of this powerful software suite.

About the Author



Duncan Scattergood is Operations Director and co-founder of Customer Systems plc, where he is responsible for all sales, marketing, project delivery and recruitment functions. Customer Systems is wholly focused on Siebel, Siebel Reports and Siebel Analytics and has accomplished hundreds of successful implementations. Based in the UK, they have delivered services in 29 countries across 4 continents. 19% of the customers listed in Siebel's last annual report have used Customer Systems.

Duncan can be contacted at dscattergood@customersystems.com