

**TURBINE
KREUZBERG**

***RISKY BUSINESS:
HOW TO SUCCEED
IN BUILDING
DIGITAL
PLATFORMS***

INTRODUCTION

WELCOME TO THE LAND OF RISK AND HIGH REWARD.

Building a digital platform, like any large and complex tech project, can be a perilous journey. From the client to the agency developing it and business consultants somewhere in the middle – everyone has something to lose.

Seemingly out of nowhere, implementing certain features becomes more complex than originally thought. Existing system landscapes reveal unforeseen hurdles to integration. Agency partners can turn out to be in over their heads, having sold a promise they cannot keep. These things have happened, all to the detriment of a project's planned scope and budget. Building a project on shaky foundations will lead to a faulty outcome – if anything comes out at all.

Still, there is hope. Even if many digital projects are doomed to fail, it does not have to end this way. Projects fail along three dimensions: time, budget and expectations. Development may take longer than planned, costs may get out of hand and the final outcome may not be as expected – most often, all three occur simultaneously. In general, what it means for a platform project to succeed is simple: the outcome generates real business value for the client, thereby proving the client's market concept is viable, while staying within a predefined budget.

Often, it is not technological complexity that leads a project to fail – though, to be clear, technology plays an immense role. The main risk factors lie in miscommunication, insufficient preparation, and inadequate controlling. Let's dive into the main problem areas and explain how to avoid them, in order to build platforms that succeed.

THE DEVIL IS IN THE DETAILS.

Any given platform project begins with defining what actually needs to get done: engineering requirements. What features should the platform exhibit in order to fulfill its purpose and have the intended business impact? This question shapes everything from which technologies need to be implemented, what expertise is required of developers, development times as well as costs. It is critical – and really hard to answer definitively.

Most client companies find their answer in breaking down a granular list of features and describing their functionality in high detail. This, in itself, is not exactly a bad thing; it's good to know what you want. What makes this practice dangerous is what it suggests, namely that a project is only successful when the outcome is exactly the way it was originally described in a detailed, finely tuned concept. This is a problem. A resulting platform has to fit the business model and match the market approach, not encapsulate every feature idea. Doing the latter precludes the ability to verify that all the things one originally dreamt up actually add real value. In practice, though, only while implementing them does one see whether certain features are valuable – or just too expensive for the value they are meant to add.

»What we have found is that the deciding factor isn't how good or fast developers are – they should be, no question. Still, it's actually more important to focus and set the right priorities«, says Matthias Gronwald, Technical Director.

In fact, the requirements phase is all about focus. When wishful thinking meets reality, what is doable and what is sensible? In the beginning, often everything has the highest priority. The job of an agency is to first communicate what is realistic within a given time frame and budget. Then, it's critical to assign what needs to get done first and what can wait. »This is where experience is really important. Translating requirements into technological implementation is the easy part. The hard part is convincing clients that features they really want are unrealistic or less critical than they think«, says Matthias Gronwald.

If feature lists are so terrible, why are they so hard to get rid of? The answer lies in one word: security – or rather, the perception of security. Having everyone agree on a list of features to be implemented suggests that this is indeed what they will end up with, and it minimizes the perceived risk of not getting what was agreed upon.

»Frequently, detailed feature specs become part of the contract between client and agency, as a means of securing one's position in case things don't go as planned. But this security is completely fictional. It doesn't just automatically make changing conditions or new information over the course of the project go away«

Christopher Möhle, CEO

»The idea is to define the minimum variant the client can already live with. Then, you define several steps or iterations up to the maximum variant possible – always keeping in mind that you have to meet the deadline. That's what we all want«, Gronwald adds.

Instead of focusing too much on features, companies should be very clear in their overall vision and goals, and have an adaptable plan for all the rest. Software projects are agile projects. The way to go in platform development is to define thirty to forty core requirements and flesh them out, but in a high-level, rough way, not by breaking them down into six hundred sub-features. A good project is not just built on sound strategy – it's also about deploying the right tactical maneuvers. Having high-level requirements opens the path towards tactical decision making, because they can be continuously reinterpreted in light of new facts, figures and developments.

AGENCY SELECTION

SIZE ISN'T EVERYTHING. COST IS EVEN LESS.

It's simple in theory: anyone devising a platform project needs capable hands to actually implement it. This means finding the right service provider, most likely a digital agency capable of handling the job. Often, digital consultants can help in the search, as they know which agency brings what to the table.

But what exactly is the right agency? Generally speaking, it should minimize factors that could tank a project and maximize ones that make it fly: technological expertise, loads of experience, teams with the right people, willingness to challenge assumptions, and so on. Usually, though, the decision comes down to just two factors – size and cost.

Hiring a large agency promises reliability. After all, the chances are pretty slim that an agency with a big name will allow a project to go under. This reliability, though, comes at a price. Does that peace of mind justify markedly higher costs? There is no right answer – only the reality that projects succeed and fail across the board, from large corporations to smaller companies, from big, global agencies to a few dozen people in a factory building.

Too often, clients optimize for project costs, trying to fit as many requirements into the overall scope within a fixed budget. »What we see over and over is confirmation bias at work. Companies keep searching for an agency until they find one that will tell them exactly what they want to hear: their set budget matches their ideal scope. Regardless of eleven other agencies having told them it's not that simple«, says Christopher Möhle. It's smart to be skeptical of this, because it is a clear sign of an agency under pressure overpromising to close a client. It leads to less than stellar outcome quality and surprises along the way. As the saying goes: buy cheap, buy twice.

»We know that development efforts are measurable. You can objectively determine how many days it takes to develop a given feature. And if the initial cost of that is too high, we have to adjust the scope. The alternative – failing projects – are in no one's interest. Cost is always a matter of negotiation. But it's good to keep in mind that any calculation is based on assumptions; in a way, it is all fiction it is all fiction«, Möhle says.

External digital consultants should help lay the groundwork for a flexible, agile project by shaping the client's expectations early on, as well as focusing on project experience and comparable references when selecting an agency for implementation. A general rule of thumb is: test how agencies behave in conflicts and discussion, rather than going on cost, size or even sympathy alone.

TECHNOLOGY SELECTION

USING STANDARD SOFTWARE IS SMART. UNLESS CUSTOM SOFTWARE IS SMARTER.

If the most important question is what purpose a platform is supposed to serve, then a very close second is with which toolset to actually implement it. Which technologies are suited to build and run your platform? Of course, this question cannot be answered generally; it's just too specific to each and every case. Still, selecting the right technology stacks is risky because of the high costs of enterprise software licenses – and then you haven't even started implementing them yet. You want to be sure you're using the right tools for the job.

More than anything, choosing technologies requires a close reality check. Often, companies pick standard software products – Hybris or Spryker for Commerce, Salesforce as CRM, S/4Hana as ERP. The promise of using established, tried and trusted products is obvious: everyone is using them, hence the standard, they have solid ecosystems, and there is lots of experience available. This means once a technology has been decided upon, you have narrowed down your options of service providers capable of implementing it. Companies map requirements with the feature catalogues of software products, make a decision and then look for an agency.

»Often, when we get involved in projects, we find the technology question already answered. Best case, there was some level of experience involved throughout the decision process. There are cases, though, where we are not at all convinced the software chosen is right for the project. Or it could have been solved by going another route. But licenses have already been paid, so going back is not much of an option«, says Sarah Hoidn, Senior Consultant at Turbine Kreuzberg.

Generally, it is good practice to at least get a second opinion. To avoid spending money on one tool when another may be better suited, involving the implementing agency early in the decision process can be key.

Standard software products do have their downsides. The most important lies in the extensive customization necessary to adapt them for each individual use case. Most times, this means getting rid of some modules and adapting others, which takes time and costs money – this, on top of license fees.

»If it fits the use case, especially larger companies could do well to be somewhat more brave and develop things from the ground up instead. I would rather invest in a developer team or put a small agency team on it and do it myself over two years, mapping my requirements as I go along«

Matthias Gronwald

If done well and according to established standards of development, custom development can lower follow-up costs. In addition, it enables building up your own tech expertise. After all, a platform is more than a new sales channel. Companies that build and run platforms must understand technology at their core.

INTEGRATION

IMPLICIT INFORMATION KILLS PROJECTS.

Systems integration is likely the most resource-intensive part of implementing any platform project. Very rarely does one get to build something from scratch on the metaphorical »green field«. Today's organizations have existing technological infrastructures and running systems that need to interface with the new platform. And many tend to utilize legacy software that needs to be connected.

System integration bears risk, because problems often only become apparent once you are looking under the hood. The implementing party has little to no influence over the existing IT landscape, but implementation is really influenced by it. »One of our main tasks is to convey to the client their biggest problem areas, which can massively influence their wishlist and project timetable. And we have to show that we are on their side, but we cannot solve all these problems just by ourselves«, says Matthias Gronwald.

»In pitches, clients often ask us about various technologies we should be able to handle, because they're afraid we have any gaps. Sure, there are things we've never done before. But technology is always a tool and any developer worth their salt will learn something new when needed. Projects don't fail because of that.«

From a technological standpoint, they can fail because developers don't know things they really should have known. The main issue is how knowledge is institutionalized on the client side. When it comes to API integration, such as ERP software, it's all about internal organization and processes, and whether knowledge is made explicit. In software projects, oftentimes there are only one or two people on the client side who know how something has been built and why – if at all.

An example: a company has customer data and wants to use it in a new commerce platform. The data itself seems well-structured, developers can build an API for the platform to be able to read it, the API is tested and everything appears to run smoothly.

After a while, though, strange data begins showing up in the frontend – data that includes internal employees. After investigating the issue, it turns out that a few years prior, someone had entered employee information without adjusting the data structure, and until now, it has worked without issue. While this might have been an okay fix at the time, the problem is that existing software was used for an entirely different purpose, this was not made transparent and the team developing the new platform was not told.

So, the important thing to do on the agency side is to ask the right questions and get access to datasets as early as possible. And, it goes almost without saying, extensive testing should be a key part of implementation. »Experience has taught us to bring this up very early with technical contacts on the client side. It is important to insist on seeing complete datasets as early as possible. If not, then we don't really see larger issues until implementation – and then often only during integration testing, when they become apparent in the frontend«, Gronwald says.

On the client side, information has to be as explicit as possible. This does not mean simply having more documentation, as all documentation is out of date to a certain degree. Information is best made explicit within the software itself, meaning: if you have a new requirement for an existing system, develop the software accordingly, don't bend it to your use case. Don't just add the employee data to the customer data, just because it's easier to do.

This also speaks to a larger point. »Always make sure that existing infrastructure is as well-prepared as possible before starting to implement a new project«, Matthias Gronwald advises. Data should be clean, the quality of data structure should be high, and APIs defined explicitly.

CONTROLLING

CONTROLLING ENSURES MORE THAN A CLEAN PROJECT. IT KEEPS YOU FLEXIBLE.

No project can ever be successful without a high-quality controlling setup. Software projects are risky; risks need to be managed.

Put simply, most controlling looks like this: knowing how much total budget is available, keeping track of expenditures, and figuring out the difference, in order to allocate correctly for remaining efforts. Controlling is done periodically, such as monthly, bi-weekly, or around project milestones. Account managers share and discuss the results with project stakeholders to negotiate the way forward. Sounds easy enough.

In truth, this does not meet the mark: when it comes to controlling, the agile approach is over. While in the planning stage you should remain flexible and focus on high-level requirements, you should be much more detailed when controlling the actual development process. It is critical to know how much time has been spent not just on each task, but also each subtask and sub-subtask. How far along are they in development and how much more time is required to complete them? Time, after all, is money, and money is finite.

In addition, controlling is a continuous exercise conducted throughout the project, not just periodically. Agencies should be able to give a highly detailed, granular status update every day – if not for every ticket, then at least at the epic level. Continuous micro-controlling should never be neglected for risk of losing oversight and failure.

This keeps a project adaptable and allows for shifting resources. »One should see red lights as soon as they start blinking, not just at the end of the month, or in the worst possible case, when a project is almost completed. Put simply, you have to know at every point how many days of development are left for each task and reallocate accordingly. Things that were in no way obvious before and aren't part of any contract will suddenly become unavoidable. And they still have to get done«, explains Christopher Möhle.

This means getting developer input to make residual estimates. Time and time again, developers have to reassess how much future effort will be required. This variable changes constantly, because the longer developers work on a project, the more they learn, the better they become at anticipating how long something will take, and the faster they get.

Communication with stakeholders is an important part of this, keeping them informed as transparently as possible. This requires effort and resources, the right people doing it, as well as a system everyone can agree upon.

»When problems arise in a project depends on two things: experience and quality of controlling. Experience lets you know from the start when something is too complex and costly for the budget and to estimate what would actually be required«, Möhle says.

»Good controlling allows you to »positively disappoint« as early as possible, meaning pointing out problems before they manifest and escalating them immediately. If these things unexpectedly turn up later, you haven't done your job.«

Christopher Möhle, CEO

In short, a project that shows cracks early on is often more successful than a seemingly smooth, quiet project. Problems turning up during the implementation phase can be a sign of inexperience. If they reveal themselves at the end, controlling has fallen short.

COMMUNICATION

COMMUNICATION IS A TEAM EFFORT. AND EVERYONE HAS A ROLE TO PLAY.

Keeping a constant communication stream up and running with all the various project stakeholders throughout a project's phases is hard work. While the advice may seem pretty shopworn, never forget: communication is a team effort – it cannot just be up to one person alone for all purposes at all levels.

»What happens in quite a number of projects on the agency side is that the product owner gets overloaded with all the client-facing communication tasks«, Sarah Hoidn explains. »This is, of course, on top of managing the project. Both are extremely time consuming and complex. It's good practice to separate them into different roles. Otherwise you risk crucial things getting missed, as well as burning out qualified people.«

Which roles are present in a team is key. Besides the developer team, there are three roles: a key account manager, a product owner and an agile coach. The agile coach's job is to keep the team running at maximum capacity – always keeping tabs on how much money it is actually using up in every sprint as well forecasting everything from when people are on leave, to how many sick days a team typically accumulates, and so on. The product owner's job is to distribute the project budget among the various tasks that need to be carried out and then see them through.

A third role, a key account manager, is tasked with preparing all the information for stakeholders in order to show the project status at any given time. This often means being the bearer of bad news, sometimes prompting heated discussion – something not everyone is suited for. This should not be done by someone tasked with day-to-day project operations.

»Sometimes, you can say four months in advance that there is this one stream, data migration for example, which, as we can already see, will be twice as expensive as planned and is going to mean rebudgeting«, Christopher Möhle says. »While the basis for this is precise controlling, what results is a communication challenge: giving bad news and showing a way out.

Some will wait until the very last moment, but being transparent about issues months in advance is much more de-escalating than doing so when it's too late. And while tough, it shows everyone you're in control and trustworthy.«

Obviously, it goes without saying: clients need to be updated on the status of their projects. Still, and this is one hard-learned truth, too much of a good thing can be a bad thing. To be precise: too much transparency without the necessary context can be problematic.

When a project has been running for a while, client stakeholders will get the hang of things, even if they are less experienced in the type of project at hand. The account manager on the agency side must always keep in mind that the knowledge clients gain through continuous status updates is generally still superficial. Unless they are absolute experts, they are not deeply involved at the technical level. This is where the right context is key. Otherwise, things can get dangerous very quickly.

»People can draw bad conclusions from faulty assumptions, which leads to even worse decisions. Most of the time, however, client stakeholders see only the metaphorical tip of the iceberg. It is an experienced agency's job to know how to frame information and contextualize it. Transparency for transparency's sake is simply not enough«, Sarah Hoidn explains.

»Instead, it helps to think about what is necessary from the project's perspective. How will this information benefit the project and its outcome? Information has consequences and it's good to game them out in advance.«

If it's not already high, client engagement should be promoted throughout the project. The feedback gained from clients that actively take part in reviews is invaluable. Feedback is a continuous process – you're not working towards fireworks at the end, but approval and disapproval throughout the project to shape its outcome.

Authors

Thomas Weber, Brand Ambassador
Samuel Krist, Brand Consultant

With input from

Christopher Möhle, CEO
Sarah Hoidn, Senior Consultant
Matthias Gronwald, Technical Director

**TURBINE
KREUZBERG**

Turbine Kreuzberg GmbH is a digital agency specialized in implementing transactional business models. Working across a wide range of sectors, Turbine Kreuzberg develops tech strategies and builds digital platforms, marketplaces and individual applications.

Counting 100+ people with high technological expertise, the Turbine Kreuzberg team is based in Berlin, Leipzig and Faro (Portugal).