mxa consulting

12 Reasons Why Projects Fail

Peter Grant, Public Sector Research

February 2026

Synopsis

Over the last 30 years, there have been dozens of examples where important business / IT projects have failed. These failures have many forms: the project may have been abandoned, significantly overrun budgets or delivered suboptimal outcomes. This history of failure (even if your agency hasn't had any issues with projects) creates doubt in the minds of executives – doubt that means agencies may negatively assess the achievability of important initiatives. If executives think an initiative may fail, they may choose not to go ahead. This has an obvious flow-on effect for innovation and transformation initiatives. How can organisations confidently embark on these big changes if there is uncertainty about their ability to do the job? This all sounds a bit negative… But it doesn't have to be. The good news At Public Sector Research we believe there are 12 major causes of business / IT project failure. These account for almost every major project failure in the last 30 years. And the good news is, each of these is easy to identify and even easier to fix.

1.1 Key Takeaways

Business IT projects don't have a great track record. Three decades of reviews and audits have failed to identify the real causes of project failure – failures keep happening. We have always assumed projects fail because of some weakness in the project team. This is wrong. Projects fail because organisations are not ready to support them appropriately. Any organisation can become great at project delivery. They just need to decide. The 12 causes of project failure below are simple to address.

1.2 Overconfidence

Everything looks easier at the start. The start of a major project can be heavily influenced by the Dunning and Kruger effect. Dunning and Kruger (1996) argue 'the less people understand about an issue the more confident they will be about managing that issue.' At the start of a major project, business executives will not be able to identify all the issues and risks associated with the project. Believing everything is far easier than it really is, executives allow inadequate controls and suboptimal strategies to drive the project. Unfounded optimism at the start of the project leads to the establishment of poor practices that have a lifelong impact on the health of the project.

1.2.1 Inadequate risk planning

Any substantial project will have hundreds of potential risks and issues. Identifying these at the start of the project and developing plans to mitigate them is fundamental to success. If people believe the project is simple, they tend to either skip this important step or treat it as 'compliance' documentation. The giveaway of this behaviour is a very short list of project risks in the risk register.

1.2.2 Lack of executive involvement

Executives are busy and don't want to bother with things that look easy. Why would they get involved with an easy project? Lack of executive involvement (especially in transformation initiatives where big calls need to be made during the life of the project) is a key reason projects go off course.

1.2.3 Choosing the wrong solution

Choosing the wrong solution will doom a project from day one – every minute of work effort in the project will be aimed at the wrong outcome. This issue is far too common. The cause is a less than adequate effort in the business case generally caused by underestimation of the project. People think, 'why would we invest in an expensive and complex business case for an easy project?'

1.2.4 The wrong project strategy

Choosing the wrong project strategy will absolutely guarantee project failure. This paper touches on project strategy (especially implementation strategy) in a subsequent section. I am sure a lot of readers are underestimating this issue right now. Please don't. Overconfidence is probably the number one cause of project failure. It's subtle but 100% effective in insuring projects do not work well.

1.3 Clarity of Vision

Sometimes executives assume everyone knows about their project's goals. Over time, folklore takes over and there are dozens of project goals floating around. Maintaining a focus on the actual goals needs ongoing proactive attention. An experienced executive once explained it this way: *'If you haven't told people at least 50 times what you are trying to achieve then you are failing in your job as an executive and communicator.'* Fail to reinforce the project's goals and in no time the project team, stakeholders and vendors will all have a different view on how they add value. This is deadly – they will make hundreds of micro decisions every day all targeting the wrong outcomes. The symptoms of this issue are easily identified.

1.3.1 Meaningless project names

Often, we see project names like 'IT REFRESH' or 'IT REDEVELOPMENT'. Meaningless IT-based project names like these indicate low levels of understanding of the project's actual goals. If someone is struggling to name a project, they probably don't understand (and maybe don't believe in) what the project is trying to achieve. The business case including the options analysis must be completed before there is a project at all. Constant debates over priorities Every project has some tension over project priorities. Which new business function will get the most features? What functional area will be implemented first? These are normal priority discussions within any complex project. When the objectives of a project are unclear, these priority discussions become more radical. The many differences in perceived goals driving a complex web of priority demands.

1.3.2 Unclear (or no) implementation strategy

Clear, generally agreed goals and priorities need to be defined before an implementation strategy can be developed. The implementation strategy MUST be identified, agreed and funded at the start of the project. Unplanned and unfunded implementation strategies almost guarantee delays and budget overruns.

Want to check if your project team know the goals? Walk around the floor and ask everyone about the project's goals. Don't be surprised if you get a whole bunch of different answers.

1.4 Weak Options Analysis in the Business Case

The business case and options analysis are often completed by the project team. This is a recipe for disaster. Project teams have a different motivation than an in-house architecture team – the people who should be responsible for the business case. One is about just getting it done on time within budget (project team) while the other (architecture team) is about getting the business and IT investments across the whole organisation to work effectively together. Can you see the conflict of interest here? Get the options analysis wrong and subsequently everything you do, every dollar you spend and every promise you make will be a waste of time. Your project will be fodder for yet another audit report that will tell you your project team failed. No, they didn't – the organisation failed! Develop business cases using the in-house architecture team. This team know how all your business and IT systems work together. They are the ONLY people who are positioned to: 1. Ensure an integrated solution that will work with other systems. 2. Direct the project team to common infrastructure in use by other systems in the organisation – saving time and money while reducing future costs and complexity. 3. Determine and cost implementation strategy options.

1.5 The Sponsor Didn't Know What They Wanted

This issue is related to the business case but a little more subtle. Here the sponsor 'thinks' they know what they want. They build a business case around these requirements and proceed with the project. Then as the project progresses, the sponsor changes their mind, not once but hundreds of times. Generally, the sponsor has no idea of the impact of these changes. Some are simple while others impact the foundational architecture of the product or services being built. In IT, just as in the built environment, it is very difficult (often impossible) to retrofit a foundation. Often the best approach is to start again. Unfortunately, few projects do. And almost all these projects fail. The Sampoong Department store collapse provides a tragic example. *The building's plans originally called for a residential apartment with four floors. However, during construction, the blueprints were changed to instead create a large department store. This involved cutting away a number of support columns to install escalators and the addition of a fifth floor. On June 29, 1995, the building collapsed, killing 502 people and injuring 937. This is a tragic example of the risks involved in making changes on the fly without a deep understanding of the consequences.* As business system projects progress, stakeholders and the project team learn more and more about the requirements. This increased knowledge drives demand for changes to the original specifications. Each of these changes needs to go through a design authority approval process to confirm: ▪ whether changes should be implemented given the work already completed; and, ▪ if proposed changes drive implications for areas outside the project. Some changes can be implemented easily – it is up to the project team to only consider the impact on time and cost. Others may be far more dangerous as, like The Sampoong Department store example, they run contrary to the foundations already in place. In the absence of a design authority, stakeholders and the project team will have no mechanism to (thoroughly and independently) assess the impact of a proposed change. We must stress many of the potentially deadly proposed changes often look benign on the surface. Why don't organisations have a design authority function? Well, the authority is staffed by the architecture team and many organisations don't have an architecture team. They leave these decisions to the project team. The project team is not in a position to make good calls on architectural issues. In addition, the project team are motivated to finish the job – the wrong motivation to underpin change assessments. It is best not to start a major project when there is a lack of clarity about the outcome. It is recommended executives build a deeper understanding of the objectives before they start the business case. Running one or more small, short-term prototyping exercises is a good first step to gaining understanding. This approach often leads to a reduction in change requests later in the project – a very beneficial outcome. Side Note: The value of a proof of concept is KNOWLEDGE about the requirements. It should never form the basis of the new system. It is an experiment and by definition is full of mistakes. Throw it away and start the project from scratch once the proof of concept is complete. Why don't people conduct a proof of concept – especially if they are not confident, they understand the requirements? Well, the answer is simple. Just go back and read Point 1 above. At the start of a project, people think they know the requirements! Overconfidence is a silent killer in projects.

1.6 Common Infrastructure Left to the Project Team

Capabilities such as middleware and identity management are considered common infrastructure. All business/IT projects should expect common infrastructure to be on the ground and ready to use. If a project finds an unexpected and unplanned need to build 'common' infrastructure, then it is in big trouble. There are no good outcomes from this point for the project or the wider organisation. Let us look at why a business/IT project team should not have responsibility for common infrastructure and then discuss how this should be managed. Why a business/IT project team should never build common infrastructure: 1. The project team won't have the right skills to select, build and implement common infrastructure – selecting middleware or identity management products for example is a job for expert architects. If left to a project team (chosen to implement a business system), it is likely they will waste time and money implementing a substandard piece of infrastructure in the process. 2. The project team won't have the right motivation – at best they will build something that suits the needs of their project at the expense of every subsequent project that looks to use that infrastructure. 3. It is much harder to build a shared function than it is to build a discrete function for a single project. It will take more time – time that won't be in the estimates. The response from most project teams will be to implement only what they need – undermining the concept of common infrastructure. 4. It is unlikely the funding for a difficult common infrastructure implementation will have been incorporated into the project's budget. In an effort to keep the budget under control (something they are held accountable for) the project team may rush the choice and implementation of common infrastructure leading to a suboptimal product for them and later initiatives to use. 5. It is unlikely there has been a business case for common infrastructure gaps that are discovered by an inflight project. An important decision such as common infrastructure should never be made by a project team – there needs to be a formal business case process. In summary, leaving a common infrastructure implementation to a project team running a discrete business IT project will: ▪ blow the project's budget and timeline; and, ▪ lead to a guaranteed sub-optimal outcome for the wider organisation. If your inflight project is faced with the unexpected task of selecting and/or implementing common infrastructure, STOP IMMEDIATELY, complete a full business case for the common infrastructure required and then get the job done in a different project with a different team, skilled and measured (with KPIs) to work with common infrastructure. What should happen: Common infrastructure must be anticipated and set up in advance of ANY business-oriented IT project. And yes, that might sound like an IT project that has no linkable business benefits. And it is just that. Sometimes it is necessary to invest in capability in advance of need. If you think about it, we build roads, bridges, hospitals and schools in advance of need all the time. Obviously, there are circumstances where proactive investments like this are just common sense.

1.7 Essential Support Structures are Missing

Every major project needs support from in-house areas with people well versed in the nature of the business, local policies, the extent of the applications portfolio, the mix of concurrent initiatives, business priorities, culture, existing skill levels, etc. This support comes from the following areas: a. Finance b. Procurement c. Change Management d. Program Management e. Enterprise Architecture. Unfortunately, project teams often cannot count on quality support from these areas. The following is a very brief analysis of the issues.

1.7.1 Finance

Finance teams can run into trouble when they fail to understand the implications of new IT business models like software as a service. Often internal standards with fixed positions on capital and operational expenditure do not align well with the move by the IT industry to a services-based model with minimal capital and higher levels of operating expenditure.

1.7.2 Procurement

From a procurement point of view, this new services approach opens the door to prototyping, proof of concept and sandboxing. In some agencies, procurement policies do not support this approach.

1.7.3 Change and Program Management

Too many agencies see change / projects as a rare and one-off event. The reality is there will be a constant stream of projects for decades to come. This incorrect perception that projects are RARE one-off events has resulted in many key roles across change and program management being filled by temporary short-term resources. There is a strong argument for organisations to have longterm people (can be contractors) in these areas – people who understand the business intimately and who have built good relationships with key stakeholders. When called upon to aid new projects, these longterm resources will be able to provide urgent, timely deeply contextual advice and support. Many organisations about to embark on a major project make a last-minute decision to engage contractors to fill many of these key change and program management roles. These newly appointed contractors are not ready to make considered decisions in the timeframes needed by a fast-moving project.

1.7.4 Enterprise Architecture

Underinvestment in the enterprise architecture team is probably the most impactful issue facing projects. The following provides some examples of how in-house architecture teams should support a project. The items flagged with a * happen before the project even starts – before there is a project manager and before there is a project team! ▪ Identify the business needs and develop business model options* ▪ Conduct technical options analysis - assess the market for technical options* ▪ Ensure there is support for identity management* ▪ Identify integration options and provide an integration strategy* ▪ Plan deployments – architects should know about the wider applications portfolio and where all the data is located* ▪ Develop a decommissioning strategy for systems being replaced* ▪ Recommend the security architecture* ▪ Consult with other system owners on impacts to their business operations* ▪ Provide advice on change requests through a design authority service ▪ Help document risks and add these to the risk register for the project and the operational risk register for the organisation ▪ Provide support to the project steering committee ▪ Review and update the information architecture and digital strategy Do you have a team of permanent in-house professionals supporting projects in finance, procurement, change management, program management and enterprise architecture? Are your internal financial and procurement standards up to date? For most organisations, the answer to all these questions is NO. And yet these same organisations expect transformational projects to run smoothly!

1.8 SaaS is the New Paradigm

In 2022, unique solutions can be made up of COMMON components (called services). It's how we put these services together that makes the overall outcome unique. We can make millions of unique systems by simply combining a different set of services and micro-services. You can virtually have anything you want. Services are a great way of building systems because they are cheap, well tested and ready to use. If you think about it, 95% of your new system has already been written – the services are out there just waiting for you to use them. The industry is using the term COMPOSABILITY when it comes to assembling services. Composability is like Lego – putting know components together to achieve an outcome. There is very little coding required. An organisation writing code for more than 10% of its system requirements is generally missing the opportunity to leverage services. So why isn't everyone working under a services approach? The answer is simple. Most organisations don't have anyone whose job it is to identify services that can be used. These organisations pay people to build (that means write code) functions from scratch. Consider which is the more achievable. Walk through a car yard and choose a car to buy (look for and buy existing services) or build a car yourself from scratch (write code)? Now just reflect on how you are putting your business system together! Project teams leveraging a composability approach with being able to deliver higher quality outcomes faster and cheaper. While this is attractive it will not be possible unless the organisation has established a capability to identify and implement services. See the Public Sector Research paper '45 Organisational Capabilities Needed to Implement SaaS'.

1.9 You Have the Wrong Skills

If your organisation has expertise in steam trains, then your people know all about coal and water. If your organisation wants to build a very fast train, their steam skills won't be of much use. While this example sounds too simple to even mention, in real life we make this mistake all the time. Many agencies looking to implement new technologies and/or transformational changes allow the direction to be dictated by IT people whose skills are a generation old. This skills gap can create an alignment problem between a project team looking to implement technology innovation and an IT team who do not understand and maybe doesn't believe in the new technology. This is an organisational skill and cultural issue that must be addressed before any transformation initiatives are attempted.

1.10 You Don't Have the Skills to Hire

the Right People

A lack of skills and knowledge of emerging technologies results in organisations being incapable of identifying the new people and skills they need.

Populating a selection pane with people from the steam paradigm when hiring people to build a very fast train simply won't work. And just because someone has the title IT manager doesn't mean they have the slightest clue on how to run a modern service-based digital initiative.

1.11 No Project Engagement Skills

Next time you are at a steering committee meeting take note of how many people have something meaningful to contribute. It is 20% - 30% if you are lucky. If you haven't trained people to engage with project work, then it is unlikely they will be very effective at contributing to the governance of a project. When failed projects are reviewed most reports argue for improved governance. However, few ever consider that the executives who attend governance meetings didn't understand how major projects are governed. Want to know how well your team can play a role in project governance? Ask them to enunciate their steering committee roles and responsibilities. Even better ask them about the purpose of a steering committee. Every member of a steering committee should go through a formal induction process. During this induction, it is essential to explain each person's role and responsibility in the governance process.

1.12 No Implementation Strategy

We often see projects get to the delivery stage and then discover the implementation will be a big expensive job that will take a long time. It's a nasty surprise for everyone!

1.12.1 So why is implementation so difficult?

Imagine you own a trucking fleet and you want to replace the gearboxes on your trucks. It's a simple decision. You take the trucks off the road and do the job. Unfortunately, business IT systems don't work that way. They can't cope with downtime of more than a few hours. In trucking terms, it's like continuing to use the truck for deliveries while you are changing the gearbox. Because implementation of business IT systems is time critical, we need to make changes in periods of minimal downtime. To address this constraint there are two industry approaches – Big Bang and Incremental implementation.

1.12.2 Big Bang versus Incremental implementation

Most modern business systems have dozens of interconnected functions. If we choose an incremental implementation approach, we will need to keep old and new functions running simultaneously. A little like only changing the first gear in the truck's gearbox while leaving the other gears in place – all while still making deliveries. On the surface, an Incremental implementation approach looks complex and expensive – and it is. So, what is the alternative? The alternative to an Incremental approach is known as a "Big Bang". Under a Big Bang all functions are replaced at once. This is often the step chosen by organisations that haven't bothered to plan their implementation. It appears to be an attractive way of minimising the complexity! Unfortunately, it ends up being far more complex. Big bang approaches rarely work well. They open the door to change overload in the business and support overload from the IT team. Because of the scale of the change, unforeseen problems become unmanageable. Imagine two towns connected by a bridge. A project is started to build a new bridge out of materials that have never been tested. The day the new bridge is opened, the old bridge is pulled down before even one car drives on the new bridge. Does that sound risky? What if the new bridge can't support traffic? How will people travel between the two towns with the old bridge pulled down? That's a Big Bang approach! How would you feel about it? It is very unlikely an experienced person would recommend a Big Bang approach. If the implementation were small enough it would also be easy to use an Incremental approach. If the implementation was bigger than a Big Bang approach would be far too risky. There just isn't a good time for a Big Bang. This leaves an Incremental implementation approach. What are the challenges of an incremental implementation approach and how are these overcome?

1.12.3 The challenges of an incremental implementation approach

▪ An incremental implementation approach often requires temporary structures (small temporary systems) to synchronise data between old to new functions. Sometimes this synchronisation must be kept in place for months. These temporary systems need to be production quality. And their development and deployment take time and cost money. ▪ Incremental implementation approaches require close cooperation between the project team, business executives and people on the business coalface. There may be short periods when the workload for business users increases and/or where short-term training in new procedures is required. A good relationship between the project team, change manager and the business areas are critical. ▪ An incremental implementation approach will involve a prioritisation decision – which functions are implemented in which order. This has an impact on the operations of the business and (surprisingly) on the budget for the project. Technically there will be an optimal implementation sequence – one that is the lowest risk and lowest cost. This is the one the project team will prefer. From a business perspective, the preferred implementation sequence may be quite different. Meeting the preferred business priorities may cost significantly more and take longer to achieve. The only right answer is a negotiated one. ▪ Embedded in all the above challenges is the need for data conversion. The data conversion process provides an opportunity to identify and repair data quality issues. Software to undertake the data conversion needs to be purchased or developed.

1.12.4 Implementation Summary

An organisation's ability to implement a new initiative can influence if the initiative even goes ahead. Implementation could blow the budget, be considered too risky or just take too long. Implementation challenges can also impact the choice of solution. A great solution that cannot be implementation is of little value. Public Sector Research argues the ONLY way to make good decisions about these important challenges is to have them well researched and this MUST be done in the business case – well and truly before stakeholders even consider if there needs to be a project at all.

1.12.5 But wait – we always seem to have implementation challenges at 5 minutes to midnight.

Project teams are often blamed for overruns and other failures that occur because no one has considered the costs, risks and time it will take to implement an initiative. This is not the project team's fault. Identifying implementation costs and challenges should have been done before there was a project team at all. Why isn't the implementation strategy considered in the business case for major projects? Unfortunately, it comes down to culture and relationships. Here are some reasons why the implementation strategy is not considered in the business case: a. Implementation planning requires intimate knowledge of the systems portfolio, integration architecture, data sources, business process and business priorities. A permanent architecture team can build this knowledge over time. Unfortunately, many organisations do not have an in-house architecture team and/or the existing team in not resourced to capture this detail - the information is just not available. b. Many business cases are outsourced to consultants. External consultants could never have sufficient inhouse knowledge to provide advice on implementation. This critical component is simply left out of their deliverables. c. There is a mistaken belief (often put forward by consultants who want to cherry-pick the easier parts of business case work) that we should look at the implementation strategy later. d. Implementation planning can be quite difficult and time-consuming – even with a quality in-house team. People argue we need more information, so we should wait until the project is well underway so we can better understand it. e. Implementation planning is a big job that can take time. People pushing to get the project moving argue to skip the implementation planning to get the business case done. As shown above, implementation can be expensive, high risk and time-consuming. If the business case hasn't recognised these impacts on both costs and timing, then the project is by definition late and over budget. And that is not the worst of it. Many projects hit the wall at this time and never deliver anything. Auditors blame the project team – when really the failure happened in the business case – long before the project was even established!

1.12.6 Summary

There we have it. Eleven reasons why projects fail – and none are the fault of the project team. Yet all of the reviews into project failure concentrate almost exclusively on the project team or the governance processes around the project. None ever look at the fundamental capability of the organisation to run projects. Ok, but I said there is a 12th reason.

1.13 Micromanagement

Sometimes a well-meaning executive will intervene in a failing project believing they can make it successful through sheer force of will. Often this belief assumes the project team are underperforming and with the right 'motivation' they will pull up their socks. Clearly, without addressing the eleven points above this approach cannot work. These micromanaged initiatives almost always end in tears. Failure will potentially undermine the career of the executive and in the process torture the project team who are already doing it tough. What must project team members think? Most must sense they have been let down by their organisation and the remedy is to have someone who knows little about projects micromanage them. We have seen projects fail for 30 years. And each time we have blamed the project team. Maybe it's about time we invested in organisational capabilities to help projects succeed. Having the right capabilities (a lot more than most organisations have) will not guarantee success. But without these in place, we are inviting, even guaranteeing failure. One final point People may ask, 'why don't project managers identify these 12 issues and raise an alarm?' I am sure there are times when they have. The problem is most of these issues are outside the remit of the project team and even the project stakeholder. These are wider organisational issues founded on long-held beliefs. It's up to executive leadership to recognise project work is both ongoing and important. With that insight they need to decide their agency will become GREAT at delivering successful projects. We haven’t got there yet!

mxa consulting

National Office
Level 3
383 George Street
Sydney NSW 2000

hello@mxa.com.au