HICX

SUPPLIER MANAGEMENT PLATFORM

The low-code alternative to in-house development

Contents

Part 1

Requirements of a Supplier Management Platform

The main focus of this paper is on how you evaluate the build vs buy options, but it will be helpful first to consider the main requirements for any such solution. These are broken down into the functional areas, within which specific use cases must be developed, and then the underlying technology components and considerations that are required regardless of the solution type.

Functional areas:

Data consolidation
Defining a 'golden record' for supplier data, followed by the heavy lifting to normalise and merge all existing supplier data, from multiple ERP and other applications, into a single merged repository. This includes being able to manage global and local data – that which is consistent centrally and changes very rarely, and that which is specific to individual relationships between the supplier and a particular entity within the buyer company.

Supplier on-boarding
A standardized process for anyone in the company to add a new supplier quickly and easily, while maintaining risk, compliance and process controls. 100% of suppliers should be covered by this process, which in large organisations with complex supplier relationships is both highly challenging and almost impossible to predict in advance so flexibility is key.

Part 1

Supplier lifecycle
Workflow and data management to support other events in the supplier lifecycle, such as new products, extending the supplier to work with new buying entities, managing compliance, performance assessments and off-boarding suppliers.

Risk and compliance management
Managing risk and ensuring supplier compliance with applicable legislation – anything from GDPR to conflict minerals and labour regulations – is integral to enterprise supplier management. At its core this means collecting the right information during on-boarding and managing the lifecycle of certification documents for many thousands of suppliers, each with different combinations of compliance requirements. Automation of these cycles is key.

Supplier performance management
The ability to run performance assessments for hundreds or thousands of suppliers combining subjective assessments completed by staff with data-based performance insights through integration with ERP, supply chain and other relevant applications.

Ecosystem management
Taking a holistic view of the entire supplier base and being able to treat this as a business asset and source of competitive differentiation is at the heart of supplier ecosystem management. Increasingly, Global 5,000 businesses are being evaluated as the sum of their suppliers, so being able to analyse, report, communicate, collaborate and make decisions at an ecosystem level is now a strategic imperative.

Technology considerations:

In addition to the above functional areas, any supplier management solution must also include the following underlying technology components:

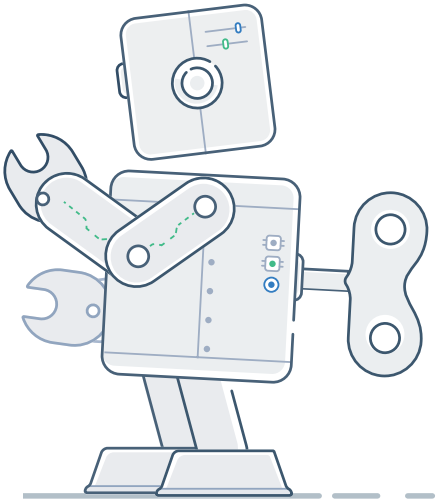
Supplier Portal
A single entry-point for suppliers for all interactions with the buying organisation, and the primary focus during on-boarding, lifecycle stages and for the exchange of information. Portal technology is mature but enterprise class solutions must elegantly handle different profiles for the business user, suppliers and approvers. They must accommodate changes to the underlying data layer and must be configurable for any supplier relationship use case, ideally by non-technical business users.

Master Data Management
A single version of the truth for every supplier, accommodating 'global' data attributes that change rarely or not at all, such as company name, DUNs number, bank account; and local data attributes which vary based on the supplier / buying unit relationship, product category, locations and more – for example being able to handle different payment terms with the same supplier and different payment or remittance destinations. It must also incorporate data governance and data management to ensure ongoing data quality and integrity, tasks ill-suited to transaction-oriented applications such as ERP or S2P.

Integrations
Supplier data is used by potentially dozens of different application types, and sometimes hundreds of different instances – particularly ERP systems in large complex businesses – and at the core of master data management is the ability to integrate with these applications and orchestrate the changes and ownership of data across the complex web of systems.

Workflow
Given the number of different processes involving suppliers – from initial on-boarding to sourcing events, purchase transactions, compliance and risk interaction, collaboration and more – a workflow capability that is highly flexible, but which can be configured to supplier-related processes quickly without having to reinvent the wheel, is essential.

Data Modelling
Recognizing that every organisation has different systems and will require different data types and information requirements, the ability to model data, create relationships between them and create the profile(s) of information needed is key. This can also include the ability to model corporate hierarchies and manage data at different levels of the hierarchy.



Part 2

Build or buy

The 'build vs. buy' consideration is as old as enterprise technology, and procurement functions in major enterprises have well-established methodologies to help project teams assess their options and weigh up the advantages of buying off-the-shelf packaged software, versus having a bespoke solution built, either in-house or by a 3rd party development team. For supplier management, the solution categories most often considered are outlined below, along with a summary of the pros and cons for each, based on the high-level requirements documented above.

Source-to-Pay Suites (S2P)
Typically the cornerstone in the enterprise procure-tech infrastructure, S2P suites are usually responsible for supporting the discovery, negotiation and contracting with new suppliers, as well as managing sourcing events and potentially paying suppliers. While in theory S2P Suites can manage both indirect and direct suppliers, in practice many enterprises have separate systems that are more tightly integrated with their supply chain technology for handing suppliers of direct materials or goods for resale.

In their favour, S2P solutions already have many thousands of supplier records in the system and they're also likely to be familiar to both internal users and the suppliers themselves. However, despite this, mature procurement teams recognise that S2P is fundamentally unable to support supplier relationship management at scale.

The systems are built for transactions and not data management and so have incomplete and inflexible data structures. They are poor at integrating with other applications, have poor support for global versus local data, and very simplistic workflows, supporting only the most common and least complex workflows.

Part 2

Enterprise Resources Planning (ERP)
The core of most enterprise application architectures since the 1990s, these started life as accounting systems and then expanded – often through acquisition – to a suite of business applications that reach into most parts of the business, usually driven by the Finance function.

As every supplier is at some point paid, ERP has at least some data associated with each supplier and it is likely to be integrated with the S2P solution. IT and Finance teams also often like having 'one throat to choke' in terms of vendors to manage, but beyond this, ERP has little in its favour as a supplier relationship management platform.

As with S2P, ERP is built for transactions and so sub-optimises the management of data for any other use case. ERP is notoriously inflexible, without large investment in customisation. AP and IT teams almost never allow any changes to the system by end users, and supplier access would be unthinkable, so some sort of gateway application is required to satisfy data and security requirements.

Even the most ERP suite-friendly CIOs (a dying breed) rarely view ERP as a viable master data management platform and while ERP is an integral part of the supplier management technology infrastructure, it lacks the core flexibility and data management capabilities that are essential.

Master Data Management (MDM Platforms
Most commonly adopted as part of an IT-driven project to improve data quality, management and governance, MDM platforms fall into one of two broad categories – domain specific or multi-domain. The most mature domain-specific MDM solutions are Product Information Management (PIM) and Customer MDM – these are both well-established categories which combine an MDM layer to ensure a single source of truth, with application functionality related to the product or customer record. Multi-domain MDM solutions, in contrast, are essentially data management and governance toolkits which in theory can be applied to all, and any number of data types, i.e. domains. Multi-domain MDM by its nature does not usually include application functionality related to any given domain and so is rarely considered by business-line buyers for whom domain context and application functionality are much more important than pure MDM capabilities, and herein lies the challenge for supplier relationship management. Buying a generic MDM toolkit will provide a strong data foundation but thereafter will require a significant investment, which in practical terms is much closer to the 'build' option defined below.

In summary:
The software categories most often considered for supplier management all fall short in different, but equally fundamental, ways when assessed against the criteria summarised in Part 1 of this document. While ERP and S2P solutions already operate in the supplier domain, already include part of the supplier dataset, and are already trusted by those teams that are most likely to use a supplier relationship management system, by their very nature – having been built for transactions, not data – they are unable to satisfy the foundational requirements. Indeed, there is no real debate to be had because both categories have existed for well over 20 years and both claim to address supplier data – yet there is widespread consensus that supplier data is still a major challenge for most companies. And on the other hand, MDM solutions tick all of the data management boxes, but none of the supplier domain functionality boxes, leaving project teams with the daunting task of defining the end-to-end requirements, and a long and expensive implementation and roll-out project to endure.

Build Option

While the idea of a completely bespoke solution tailored to the precise requirements of the business is appealing, in recent years most large enterprises have tried to avoid this except in cases where the solution itself represents a clear competitive advantage. Instead, enterprise IT teams, working with their major systems integrator partners, have sought to at least start with packaged software and then customise this to address any gaps in functionality.

Customisation, however, brings with it its own set of drawbacks – custom code is usually not supported by the software vendor and customisations are not included in vendor regression testing which means upgrades are either simply not possible or must be managed very cautiously and only after a great deal of internal testing and reversion planning. Changes to customised code usually rely on the 3rd party or in-house team that developed it, which may result in increased costs as the 3rd parties aren't able to take advantages of economies of scale for bespoke applications; and sooner or later the customised code may simply become uneconomical to support, leaving the customer with limited options and a great deal of IP sunk into obsolete code.

Furthermore, a custom-developed solution, even if it manages to address the initial use cases satisfactorily, will not receive the same level of investment in R&D. Developments will be limited to minor

improvements based on end-user feedback and there will be none of the innovation that one can expect from software developed for sale to multiple customers.

If customisation, then, is becoming less and less appealing, developing bespoke applications from the ground-up suffers from many of the same drawbacks – expensive to maintain as teams evolve and individuals move on, and high change management costs. To these can be added:

- Slow and expensive requirements and design phases. While end-users know what they do today, they're not usually familiar with software requirements development, so the role of business analyst and product manager becomes crucial,
- Difficult roll-out and adoption phases. With the best will in the world, custom applications don't always start with the best user experience as pressure to release 'MVP' (minimum viable product) tends to trump end-user UI.
- Escalating project costs. Initial estimates of what is required almost always underestimate the complexity involved in large organisations with global requirements and a huge number of unknowns. Seemingly simple projects become more and more complex, requiring more and more investment and stakeholders are often unwilling to stop given the amounts already sunk into the project.
- Infrastructure costs for hardware, data centres, redundancy etc. Partly mitigated if developed on cloud platforms, but still need to be managed and budgeted for by the project team,
- Operations, support and roadmap. Very quickly organizations find themselves having to make this a core capability. There is a reason why Google, Microsoft, Facebook, Amazon, etc. still don't build their own ERP systems or, even if like Microsoft they do, they don't use what they build. It is very hard achieve the same level for experience and learning as someone who is doing this as their core business.
- Security challenges for custom-built software. Often only discovered over time this can create a huge amount of risk for organisations handling supplier data.

The point about the requirements and design phase warrants a little more attention, as it is one of the obvious advantages that is traded when choosing 'build vs. buy.' Even if a given software vendor has only limited knowledge of a company's sector, and next to no knowledge of their specific processes and environment, they will still have the advantage of:

- Effective requirements gathering expertise honed by serving dozens or hundreds of other businesses with at least similar processes.
- Templates – either literally a library, or de facto in the production instances of other clients – that can likely be cloned and edited.
- Time-to-value – it's easy to underestimate the value of the concentrated real-world experience that vendor implementation teams (or their partners) have from multiple projects and the impact this on time-to-value. They have already made a lot of mistakes, iterated a lot of design decisions, crowd sourced ideas and battle-tested processes and concepts with comparably large and complex environments.
- External Perspectives – it is easy for large organizations to get caught up in how they have always done things. The ability to see other perspectives, use cases and have constructive challenging is critical in driving innovation. The old saying of 'we only know what we know,' still holds true and often leads large organizations down a dead-end road which is too far out to see in the early stages.

In summary

It is perhaps not surprising that few companies in the Global 5,000 have the level of control, insight and data quality needed to take full advantage of their supplier ecosystems as summarised in the requirements presented at the beginning of this document, given the limitations of the available packaged software in the adjacent S2P, ERP and MDM categories, and given the justifiable shift away from building major enterprise applications in-house in the last few years.

There is emerging, however, a third way that leading global multi-billion dollar organizations across multiple sectors, including EDF Energy, BAE Systems, Mondelez, Baker Hughes and more are adopting; a best-of-both-worlds approach which provides the genuine flexibility of a 'build' approach, with most of the advantages of buying a packaged software solution, and it is based on the rapid evolution of low-code development tools in the last few years.

Part 3

Low code

In many ways low-code development is not new, but rather the latest stage in the evolution of the way in which software applications are built. In simple terms, low-code can be viewed as a series of building blocks – the Lego analogy is often used – that can be assembled to create fully-functioning applications, without the need to write code.

This continues a trend which, since the early 1980s, has sought to make development faster and more efficient by distributing some of the work to end-users by providing intuitive interfaces, and which was accelerated massively with the advent of cloud platforms which deal with much of the operational burden of software development – hosting, redundancy and back-up, security, access management etc.

In its paper, "Low-code Development Technologies Guide" Gartner includes a 'Pyramid of Applications' which shows the extent to which low-code has worked its way up different classes of application, from simpler and less mission-critical 'workgroup class' and 'departmental class', low-code is now actively being adopted for the second highest, 'enterprise class' of application, with only 'Extreme-scale' above it. In other words, low-code as a methodology is now deemed suitable for highly complex, mission-critical enterprise applications.

Low-code for supplier management
Different companies manage their suppliers, and govern their data in vastly different ways based on industry and organisational history. There is no right or wrong, as it is completely driven by context. Decisions around how much is centralised and how much is localised have dramatic implications at the foundations of any supplier data management approach. How the organisation and the surrounding functions that touch suppliers are organized (Quality, Health and Safety, Sustainability, Treasury, Finance, Marketing, Sourcing, Procurement, etc.) will all drive how those foundations are laid. Those foundations are ultimately what enable your end state vision around digitalisation, simplification, and automation of supplier management related business processes.

Part 3

"

The days of IT delivering all applications for the enterprise are gone. The present and future depend on holistic and collaborative delivery of digital products by joint business and IT delivery teams, and on the elimination of separate enterprise IT and 'shadow IT.' Low-code development is a pivotal enabler for this.

Gartner
Low-code Development Technologies Evaluation Guide

If the above advantages are applicable to more or less any business application, in the context of supplier management, the keyword is 'flexibility.' In organizations with tens or even hundreds of thousands of suppliers, serving multiple business units, no off-the-shelf solution can possibly account for all variations, so the ability to configure to accommodate any scenario is the secret sauce of low-code.

By flexibility, we mean:

The data model
As discussed above, at the core of supplier management is the master data record or single version of truth for each supplier, and while there are obvious attributes that are always needed – supplier name, location, bank details etc. – in practice no supplier record is the same for every buying organization. Variety is driven by the different types of supplier (indirect, direct, government, strategic, tactical), the industry sector, the ERP(s) used, the S2P systems and the unique micro-processes used in different businesses. Assuming that 'the supplier record' is fixed and can be defined up-front to cover all future scenarios is absurd, so it is essential that (ideally) business users can add to the supplier data model as new scenarios arise.

Workflow
The term covers a multitude of capabilities but examples in supplier management include the on-boarding approvals process; managing workflow associated with new supplier types – is there a different set of approvers or a more (or less) rigorous approval process specifically for this supplier type; new legislation – for example associated with management of end-of-life of manufacturing waste could require changes to the off-boarding process. As with the data model, low-code, with its graphical interface and building block approach enables non-technical staff to assemble individual tasks and rules into an end-to-end process, without the need to go through a complete change request cycle with a 3rd party vendor or the IT department.

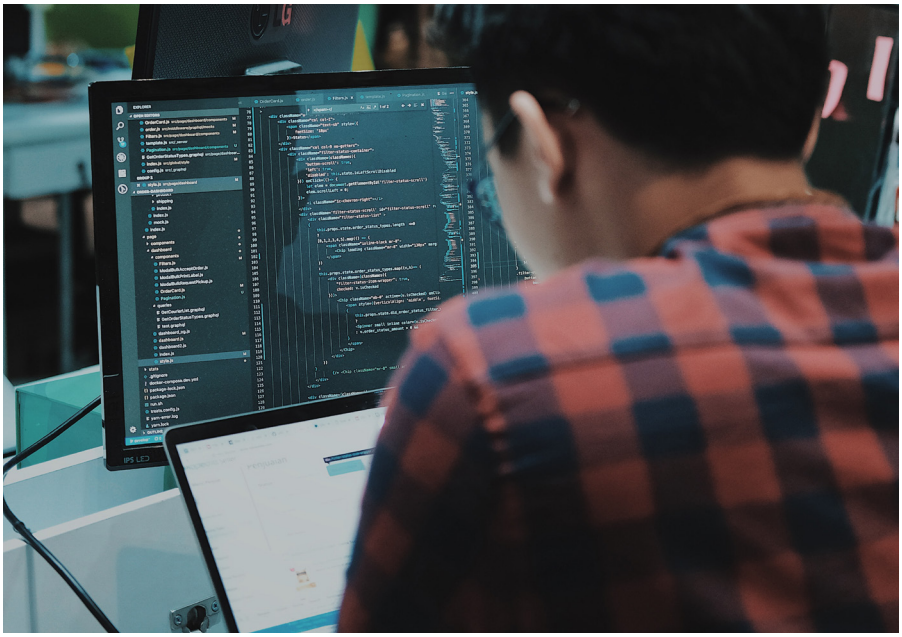
Forms
The most visible part of the solution, and the one most likely to impact adoption. Changes to the underlying data model or workflow, or simply changes needed to improve usability will often require modifications to the forms used by requestors, suppliers and approvers. These forms may well be in active use by potentially thousands of users, so without a low-code approach making even a small change can require a lengthy request, implementation and test process, all performed by a team other than the end-user. With low-code, buying organizations make changes to forms on an almost daily basis with no negative impact on the system.

Automation and alerts
Supplier-related events that trigger alerts – such as a new sanctions match, or being alerted that a business is now single-sourced in a given category – are essential in businesses with high supplier volumes and, as with workflow and data model, are subject to frequent modification which, ideally, shouldn't require a change request into IT. Using low-code building blocks and 'IFTTT' (If This Then That) style interface enables end-users to configure highly bespoke alerts as needed.

Validations and integrations
As legislation is introduced or changed, supplier compliance is impacted, and such change is only increasing. Even in moderately regulated sectors, staying compliant with financial, labour and data privacy laws across the entire supplier ecosystem is a significant undertaking and integrations to third party sources of compliance data – such as D&B, EcoVadis, Maplecroft – automate much of this. Low-code enables users to configure process flows to include call-outs to these third parties at the appropriate point in the process and / or when triggered by an event.

Stakeholder surveys
Within performance management, businesses in certain sectors run high volume stakeholder-based performance assessment programs which require complex hierarchies in order to filter and organize assessments based on multiple factors, including supplier type, product, location, BU, project and more. Low-code enables business users to configure new surveys quickly by assembling the building blocks – forms, scoring model, assessment workflow, KPIs and reporting – into any combination needed.

Part 3



Domain is king

While this paper argues that low-code is the best option for large organizations wishing to address the challenges of supplier management, it's also worth noting that not all low-code platforms are the same. As with other software categories which can loosely be classed as 'tooling' – databases, reporting tools, MDM (Master Data Management), for example – when selecting a low-code solution, there is a generic versus domain-specific question to consider.

For IT departments assembling a technology stack in order to support requirements from multiple departments in the business , emphasis is likely to be placed on areas such as how wide the talent pool is, documentation, community support, usability, interoperability, platform and security considerations. This is similar to software companies themselves selecting a technology stack like 'LAMP' – how easy will it be to hire developers, will the resulting app work on the platforms our customers use, will it enable integration to other technology environments etc.

We would argue the above are 'generic' scenarios, and the same principles can apply when selecting a low-code platform. If an IT team wishes to encourage a degree of 'citizen development' and / or to use low-code principles to speed up app development, then it might select a generic low-code platform.

This, however, is almost completing the circle back to the 'build' option which is largely discredited above. Yes, generic low-code platforms will reduce the operational burden of development – hosting, security, release management, regression testing etc. – and yes it should in theory enable non-technical end-users to make substantial changes using intuitive UI and a building block approach, but in practice, none of the domain-specific benefits apply.

As discussed earlier, the requirements and design phase is crucial and a generic low-code platform provides no short-cuts or head-starts.

Using a generic low-code platform, domain-specific integrations, which in the world of supplier management means 'punch-outs' to 3rd party data providers for supplier risk and compliance information, do not exist by default so those 'building blocks' need first to be built. Similarly, the data model must be defined from the ground up – or more likely based on the vendor master in the dominant ERP system, which in turn sub-optimises the data model for all other applications. Workflows, including supplier on-boarding and approvals processes do not start with a library of templates based on other customer implementations and so must all be defined from scratch.

Just as in the case of domain-specific master data management, therefore, a low-code platform that has supplier management domain context represents a genuine third way that combines the benefits of speed and existing IP with the flexibility of a bespoke solution.

Part 4

Business case considerations

The key business case therefore comes down to a comparison of build versus configure using low-code in order to achieve the flexibility of build, but without the associated cost and resource.

There are five key considerations to evaluate in this regard:

- Scope to be covered within allotted budget and time.
- Development costs, including ongoing maintenance and upgrades. Cloud-based solutions will likely be more cost effective over the life of the project, and will certainly be easier and more efficient to upgrade.
- Speed to deployment, with a focus on how long it would take to build versus configure a low-code.
- Expertise required (cost of mistakes and unknowns). For example, the extent to which end-users can make changes themselves versus having to engage a member of the IT or 3rd party development team will have a significant impact on costs (see the section on 'flexibility' above for the types of changes, in particular with regards to domain specific changes and updates).
- Running costs. If a solution is cloud versus hosted, or on premise, the operational cost model will vary; as it will if the solution is bought based on domain-specific low-code platform versus developing a solution in-house form scratch.
- Training. The cost of end-user (business user requestors', suppliers', approvers') training might be assumed to be a wash. After all, any solution must be 'easy-to-use,' but this isn't true in practice. Close attention must be paid to training resources and materials, learning curve, adoption and roll-out, especially if a home-grown solution is to be developed.

Part 4

The differences in cost and resources relating to the above can therefore be summarised as follows:

	Scope cover	Development costs	Typical time to deploy	Client domain expertise required	Running costs	Training costs
Build	Very Low	~$5-10 mil.	3-5 years	High	High: 10-20x for maintenance, hosting, integration, service desks, technical support	High
Low-code	Medium	~$2 mil.	12 months	Medium	Negligible, 1-4 FTEs for admin and support	Negligible
Domain specific Low-code	High	~$500k-1 mil.	5 months	Low	Negligible	Negligible

Conclusion: The best of build and buy

The success of most multi-national multi-billion dollar organisations is to a significant degree dependent on how they manage and engage with their supplier base. It is no longer realistic to assume suppliers will always want to work with you, will tolerate continued price pressure, or will accept deeply inefficient or painful processes when doing business with you. Nor is it realistic to develop digital transformation plans that in any way involve suppliers if companies have poor quality and incomplete supplier data, and are unable to derive insights from that data.

Some of the largest companies in the world recognise this and are investing in comprehensive supplier management capabilities that will extend across the entire supplier lifecycle. However, the inability of the 'usual suspects' vendor categories (ERP, S2P and MDM) to address these challenges with off-the-shelf solutions is driving many of these businesses to re-consider a 'build' strategy, despite the well-understood disadvantages.

A third way, however, is increasingly being included in the mix and is reaching a level of market maturity as the early adopters move into steady state and are realising significant benefits. Supplier management platforms, built on domain-specific MDM and incorporating a low-code development model combine the convenience of packaged software with the flexibility of a bespoke solution.

HICX is a low-code, software-as-a-service platform, that enables seamless digital supplier management. We enable businesses to find, maintain, and re-use trusted supplier data and information across their enterprise, across any spreadsheet, app or system. We ensure our customers achieve their goals and business outcomes with minimal IT input and expense using a platform based on an agile, high configuration building block approach and many years of experience in enterprise data management, combined with proven adoption and implementation methodologies.