

WHITE PAPER: LOW CODE

Reduce the Pains of Mobility. Think Low Code.

4 low-code best practices for a modern user experience and low-cost development.



Use a low-code approach to give your hardworking SAP solutions new life and create awesome new desktop and mobile apps by making the user experience, well... worth experiencing.

There's a host of business drivers pushing at the development lifecycle right now. Industries are being disrupted left and right – by big companies like Uber and Amazon and rising startups like Domo, Wework and Uptake, among others - all of them incredibly agile and able to quickly create lasting change. These disruptors are also forcing the pace of business and digital transformation to rapidly accelerate. Many businesses struggle to keep pace.



BUSINESS TODAY NEED TO BE AGILE AND CONSTANTLY INNOVATING TO STAY RELEVANT.

All the while users grow more and more savvy, increasingly demanding rich and flexible consumer-grade experiences at work. For SAP users that means new mobile and offline capabilities that work in tandem with still-in-demand desktop solutions.

Succeeding in this wildly new environment requires new strategies and capabilities beyond the traditional ABAP development skillset, ones that help reduce project timelines and costs while delivering the streamlined functionality that makes users happy.

To stay relevant, seasoned ABAP developers need to navigate this shift, and fast. And as mobile and cloud developers enter the SAP world, they too need to bridge a skills gap in order to build the kind of robust SAP tools users need. Finding and priming developers able to succeed within this new reality is a tall order for most enterprises.

Now, with SAP UI5 and Fiori as the standard for role-based, consumer-grade user experience delivery across business lines, tasks, and devices, developers must incorporate another language and framework into their SAP development process.

While Fiori empowers developers to build the kind of experiences users are looking for, the learning curve needed to be able to create Fiori-base code from scratch can be daunting for teams already overwhelmed by ever-changing business demands. And it today's increasingly self-service environment, non-developers are now being takes to own the development of business-critical apps, ushering in the need for simple development tools.

There's good news: a low-code approach can make the difference between development projects that wildly succeed...and businesses that fail.

This white paper examines the benefits of adopting a low-code approach and offers 4 best practices for moving forward.

HIGH CODE, NO CODE, LOW CODE:

What's the difference?

HIGH CODE: *Always building from scratch*

With a traditional high-code approach, it's as if you're building a modern building from scratch, every single time. There's little thought put to code reuse, and existing code isn't reused because it would be considered too risky to do so. It feels like a full-fledged renovation because it is. And it's costly.

The reality is that it's not smart to use cumbersome methods to update to new design that users want and need – especially mobile. Nor does it make sense to ignore the newer tools that make it simple to navigate the transition to UI5 and Fiori.

If you're considering building new apps from scratch using Fiori, you'll need to ensure a high element of future reuse in order to create tomorrow's low-code library. Otherwise, you'll continue to build apps from scratch, never gaining speed in the development lifecycle.

Still, this type of development process could make sense for your organization if you already have strong client-side SAP development standards that provide reliable oversight from development to design to delivery. A good central documentation repository and strong governance and QA practice are also needed to ensure what's delivered stands up to the design requirements and that what's designed fits well in context with the overall system application repository.

NO CODE: *Beautification at the presentation layer*

There are a few tools that are touted as no-code options to building a better SAP user experience. Chances are they include a limited set of app functionality, and limited customizability. Since this type of tool typically operates as a presentation layer on top of SAP GUI or WDA applications, it's also likely to be limited to desktop or online capabilities. That can limit teams needing mobile and offline functionality.

Other non-SAP no-code options rely on the available APIs and require the integration scenario to be an allowed pattern by the enterprise infrastructure guidelines. For example, if the application relies on an RFC connection, a non-SAP system may not be allowed to connect to the corporate enterprise environment.

However, since they typically let you create apps without needing to create any business logic or APIs, a no-code option may be useful if an organization is looking to push app development for simple applications to the user base.

LOW-CODE: *Preserve the best of your existing infrastructure, build a strong foundation for digital transformation*

Low-code development platforms provide simple drag-and-drop (DND) development capabilities and the functionality to help you overcome existing design flaws of older systems, so you can say goodbye to clunky workflows, ugly screens and illogical tasks.

While DND tools aren't new, what's nice about a strong low-code tool is that it will enable you to preserve the best of your existing infrastructure, while modernizing and optimizing your SAP processes and interfaces and enabling you to take advantage of all of the benefits of Fiori UX and ABAP. That includes delivering a unified user interface and faster app response times.

These features not only shorten the learning curve for developers, they make use of existing team skill sets while greatly reducing development time and maintenance costs overall. That can give you the kind of scale and speed needed to address the rapidly increasing pace of innovation happening across industries. Another benefit? More dispersed development capabilities, meaning that you can offload the simplified development to newer teams and free your ABAP resources to solve heavy-duty back-end and larger enterprise challenges.

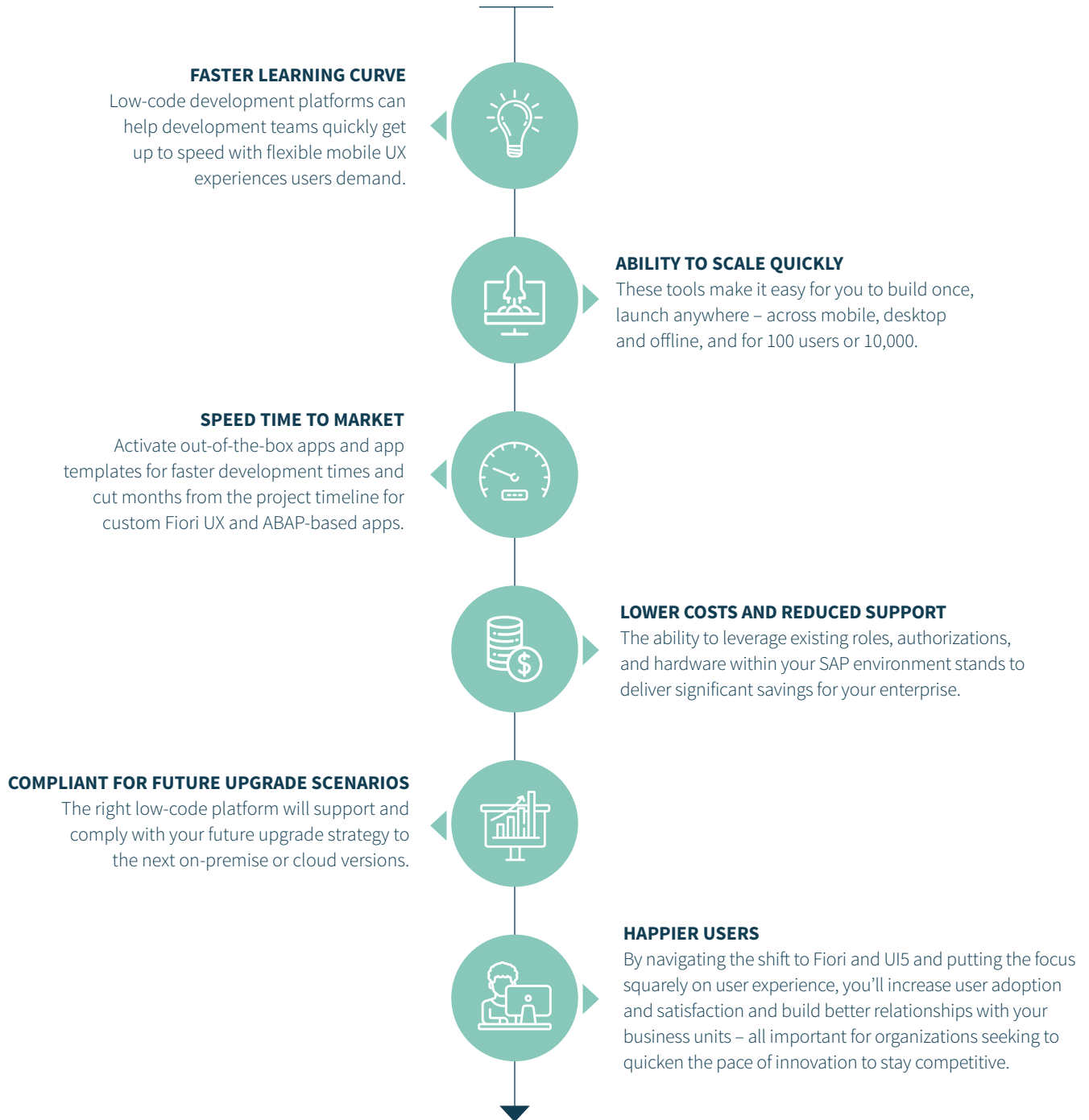
Just as importantly, they help you reduce the time and cost to design, build, deploy, and maintain SAPUI5 applications—on new and legacy SAP systems across desktop, mobile, and offline environments.

Combined these benefits create an ideal environment for building a strong foundation to navigate the coming digital transformation.

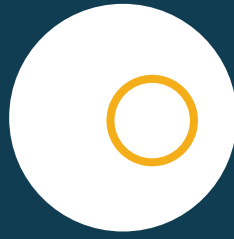


HIGH CODE, NO CODE, LOW CODE:

Why does it matter?



Neptune Software can help organizations **reduce development and software maintenance costs by up to 80%, and reduce total cost of ownership by 65%**, compared to other solutions.



THE 04

Best Practices for Moving Forward with a Low-Code Strategy





01

Build a Strong Foundation

Expand your application design skills to a broader spectrum of resource background and skills

Low-code development platforms allow the inclusion of resources that would not previously have been considered part of the delivery capability beyond representation of the requirements and user acceptance testing.

The emergence of the “Design Thinking” approach and user-friendly prototyping tools means that a lot can be designed and iterated before development actually needs to start.

Understanding and making use of this broader pool of resources should be encouraged in order to maximise the effectiveness of the discovery and design phases.

02

Prioritize Well

When prioritizing app development, start with the user first. Introduce convenience and simplification to the broader user community to create interest and demand that promotes a new faith in the capability of the application delivery function.

Next, look for quick wins that make the user’s workday easier and simpler. Tackle simple approval workflow or apps that display data only and can be enhanced with “search and display” functionality.

Additionally, existing bespoke WDA applications written using the MVC design pattern represent an opportunity to replace the WDA View components with the Neptune UI. These bespoke applications were written as functionality gaps considered mandatory to the function of the business process and therefore represent a high worth to the business that could be made to look better, simplified and potentially mobilised.

Then, address additional needs and solve complex issues that require an application based on their criticality to the business. Finally, apps that are nice to have but don’t make business easier or drive significant process change should stay on the low priority list.

DO THE MATH WITH SAP’S UX VALUE CALCULATOR

Get an estimate of the potential cost savings if you adopt the latest UX innovations with SAP’s UX value calculator.

How to prioritize? Gartner recommends basing “the importance of any app on two factors — effort and impact.”

Further, in “The Key Fundamentals Required to Scale Mobile App Development,” Gartner recommends “To scale up mobile app development initiatives, application leaders need to:

- Prioritize their app development by understanding the needs of business stakeholders; don’t sacrifice app quality and positive ROI when increasing app delivery speed.
- Adopt a bimodal IT approach by creating an agile API layer. This will optimize mobile integration and simplify the process of connecting mobile apps to many different types of data sources.
- Encourage adoption of rapid mobile app development (RMAD) tools across the organization by selecting a small subset that corresponds to organizational needs to deliver more apps.
- Implement a product management approach to mobile app strategies and application life cycle management by treating their apps as products and not projects.”

THE COST OF BAD UX

A recent report from CareerFoundry’s UXSchool estimates the cost of poor UX to be an alarming \$1.4 trillion in lost sales.





03

Build for Today, Build for Tomorrow

Make today's high-code efforts provide tomorrows low and no-code opportunities. That means making available key components including server-side application data object level re-useable objects, such as BAPIs, public class, and the like.

Relational, reusable

While SAP provides a number of re-useable objects as standard, each organization requires specific functionality and data to be available for their unique processes; therefore, applications should be written in a relational (to the data object concerned) and re-useable approach that is agnostic to its eventual consumption, whether that's by a UI application, a workflow, or a system-to-system interface.

Separate your concerns

Key to creating organised re-useable objects is the design principle of Separation of Concerns (SoC) for creating logical units of executable code that represent distinct features of functionality that exist with as little overlap as possible.

For an example of an existing SAP repository of how to organise data object level relational and re-useable object, look at the Business Object Repository (BoR) which was primarily created for possible SAP's first low-code solution, SAP Business Workflow.

On the client-side, just in the same way that snippets are provided as re-useable/template Javascript code, any organization-defined functionality should also be considered for addition as a re-useable common UI component and as part of your organization's library/repository.

04

Standardize It, Document It

Regardless of the coding strategy your organization employs, strong documentation and sound development standards and guidelines will help define how agile your organization will be in this fast-paced new business environment.

Properly written code is an important asset that can be leveraged as and when required, creating great efficiency in the development process. One example of this is code previously written in re-useable objects for workflow or Web Dynpro ABAP (WDA) applications that can be reused for UI low-code application development.

Documentation

Documentation serves as the artefact that helps the team and the business understand how something was built, as well as how it can to be maintained or extended. It also serves as the repository for possible re-useable objects.

Development standards

As standards, their purpose is to articulate a common set of principles and procedures for naming, development, and documentation of custom solutions.

As guidelines, their purpose is to provide recommendations for proven programming methods and as a way of sharing the knowledge of experienced developers with others. The goal is to maximize the quality, value and maintainability of each custom solution. Thus, it is essential that each person performing programming activities familiarizes themselves with, and adheres to the standards.

Bottom line

Good documentation and development standards are proven to prevent fragmentation and paralysis of a system. Without them, your systems are in danger of evolving into monolithic, inflexible and unmaintainable environments that no one wants to work in, and that keep your business from moving forward.



LOW CODE IN ACTION

GLOBAL CONSUMER GOODS

Global consumer goods company says goodbye to high-code approach

One of the world's largest consumer goods companies reduced the amount of time it takes to manage its high-volume supply chain process by 40 percent and expects to save 1.12 million dollars in paper costs across 16 sites by implementing a low-code approach with Neptune Software. With Neptune, the team successfully reconciled its large SAP instances into one platform, accelerate the pace of app delivery across the org, and made users happy.

[READ THEIR STORY](#)

STATKRAFT

Provides intuitive, easy-to-develop app for any device

Renewable energy provider Statkraft put users first by providing an intuitive, self-explanatory user interface that required little hands-on training and seamlessly offered mobile, desktop and offline capabilities. Not only did Statkraft simplify operations and maintenance processes while supporting an advanced level of SAP functionality for users, the company found a development platform that's makes it all easy for developers to manage and maintain. Now users love their "mini ERP system for their pocket."

[READ THEIR STORY](#)

BIOTEST

New mobile UX for sales cuts development time in half.

"What I like about Neptune Software is the clean structure on both the front- and back-end. All methods and attributes are bundled in a single ABAP class and are easy to find, making coding easy. With Neptune we were able to develop capabilities in half the time initially estimated, which was really great. And now that we've navigated the simple learning curve, we expect building additional apps will take even less time as we move forward."

[READ THEIR STORY](#)



With Neptune Software, we can deliver highly productive and back-end intensive apps in a third of the time it took with our previous options.

DARREN BAMBRICK, SENIOR DEVELOPER AT GLANBIA



NEPTUNE SOFTWARE:

The building partner you need

Neptune Software's low-code, Rapid Application Development Platform is the only solution for SAP Fiori® UX to directly leverage SAP's own ABAP source code. Our solution brings powerful SAP ABAP® and Fiori® capabilities to life today, so you can deliver the fast, secure, and "beyond mobile" UX applications your users need, when they need them, to keep business moving forward...and make users happy in the process.

With Neptune Software, you're able to maximize the use of your existing SAP set-up and minimize the need for integration, so you can deliver SAP Fiori applications quickly, securely and cost-effectively across web and mobile. We bring you a cost-efficient way to make the most of your SAP investment...and gain a competitive edge.

Neptune User Experience Platform runs on any NetWeaver 7.x platform and/or HANA and is SAP Certified. With it you can tackle the entire application lifecycle—across design, build, manage, and run. Plus, you can take advantage of over 100 ready-to-run Neptune Fiori apps that can be automatically installed and updated for fast implementation and customization.

Since 2011, more than a million users are taking advantage of the ability to conduct business anytime, anywhere with mobile and web apps for SAP built on Neptune Software.

To learn more about how you can take advantage of our powerful low-code development platform to create a modern SAP foundation, visit our website at www.invature.com



About Neptune Software

Neptune Software is a company dedicated to helping organizations accelerate their enterprise application development projects and the adoption of SAP's Fiori UX Design—all to increase employee satisfaction, productivity and business efficiencies. Neptune Software and its SAP Certified Neptune User Experience Platform provides the fastest, most cost-effective and secure way to develop and make any SAP application functionality seamlessly available across Mobile, Desktop and Offline—bringing powerful SAP ABAP and Fiori UX capabilities to life today with its low-code, Rapid Application Development Platform (RAD)—so you can solve strategic business and mobile challenges, when and where you need it.

Invature
Beyond your ERP

www.invature.com info@invature.com