

Rappit Developer Product brief



Introduction

In a crowded and competitive landscape, organizations continuously need to innovate and support new business functions. This often requires them to digitize new or changed business processes by adding to, rebuilding or **modernizing legacy enterprise applications**.

At the start of such projects, enterprises have to make a conscious choice for the most appropriate development language, methodology and platform to use. This choice will directly influence their ability to achieve a successful result-oriented outcome, within a realistic budget, and with a meaningful life-span.

While there are many platform- and technology options available, the challenge for enterprises is to choose a platform that offers the right balance between capabilities, productivity and cost, while avoiding a long-term technical or financial lock-in to a specific platform or vendor.

This Product Brief introduces the reader to Vanenburg's high productivity coding product called **Rappit Developer**. It explains how Rappit Developer provides unique advantages, while avoiding the typical pitfalls of vendor lock-in.

Modernizing Legacy Enterprise Applications

Over the past several decades, most organizations have opted to replace home-grown business software solutions with commercial-off-the-shelf (COTS) alternatives. Initially, this introduced new capabilities and competitive advantages. However, the dependence on COTS software vendors drastically reduced their ability to innovate and differentiate in a rapidly evolving business landscape. Consequently, the enterprise application market has a need for alternative options that allow organizations to take their opportunities for modernization and innovation into their own hands.

Traditional application development



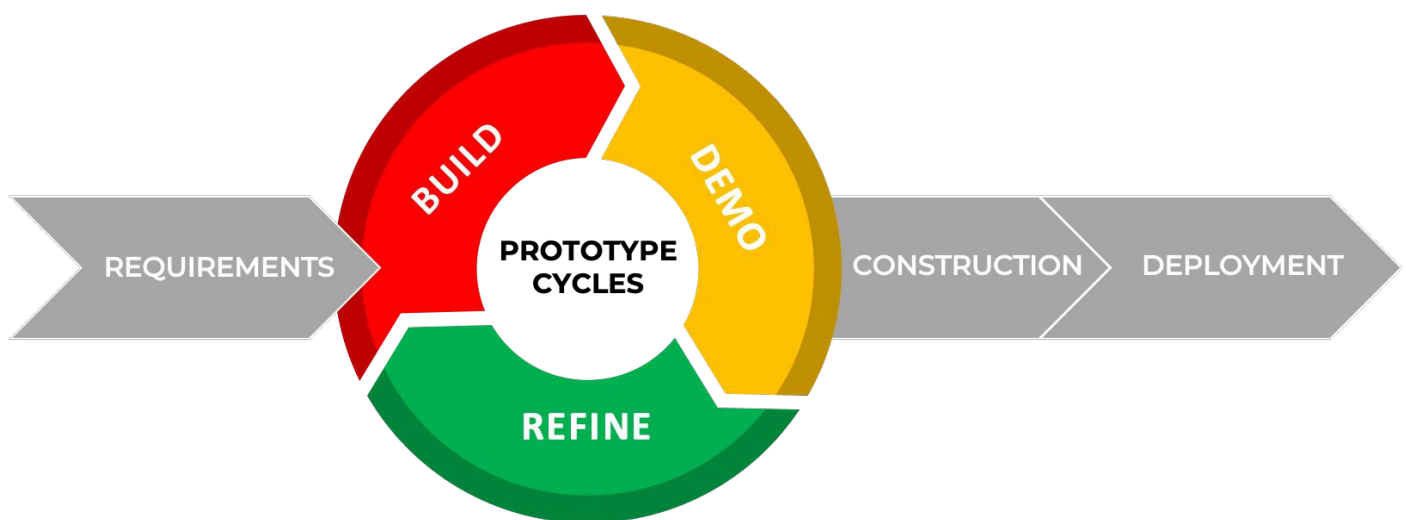
Ever since the adoption of early mainframe systems, followed by the transition to workstations, client-server architectures, and the eventual evolution towards cloud-computing, application development has gone through evolutions along the way. Apart from the emergence of a wide array of 3GL development languages like C++ and Java, the development community started looking for innovative new tools to help streamline the process of writing, compiling, testing, deploying and running code for **core system development**. This led to the emergence of so-called Integrated Development Environments (IDEs) that facilitated easier code creation with helpful user-selectable plugins, error checking, compiling, staging and delivery of resulting applications.

Core System Development

Your organization may offer a uniquely competitive product or service that is not currently supported by commercial-off-the-shelf software. You may decide to develop your own line-of-business solution, or customize an existing system. Either way, you want to fully own the resulting code and intellectual property. Your choice of a development platform should focus on capabilities and productivity, without having to worry about runtime license fees. If you create it, you should own it.

Rapid Application Development

The mass adoption of IDEs and optional plugin utilities illustrates that development teams are deeply interested in new and innovative ways to improve and streamline their productivity. One particularly important example is the emergence of a “Rapid Application Development” approach, that started with the ideas of Barry Boehm and further developed into the rapid application development methodology by James Martin during the 1980s. The premise of RAD is to **speed up software delivery** by adopting an iterative, accelerated development process built around the construction of prototypes, with direct involvement from the end user community represented by one or more consultants.



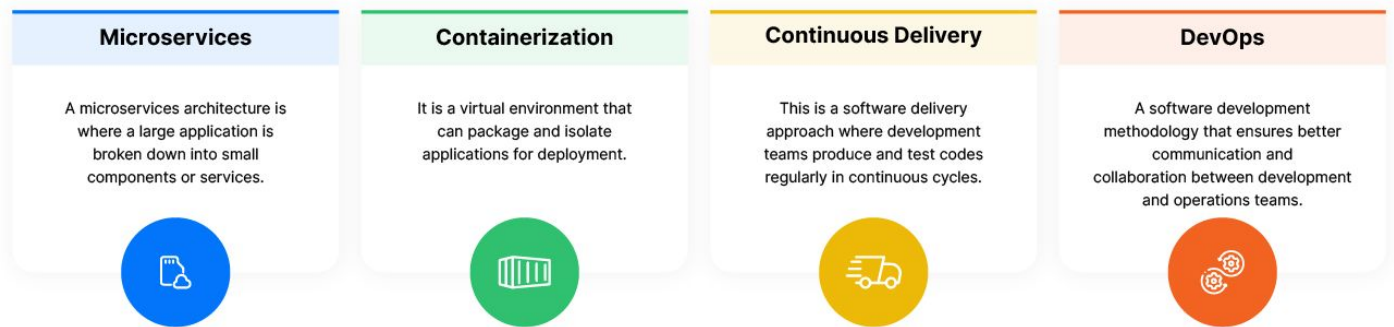
Besides using RAD for accelerated development of entirely new applications, another particularly valuable use case is **extending ERP applications or other systems of record**. Current ERP users know exactly what missing functionalities would greatly improve their productivity, speed-to-market and customer satisfaction. Their direct involvement in requirements gathering and prototyping helps ensure that the end product delivers maximum return on investment.

Extending ERP applications or other Systems of Record

ERP vendors are keen to migrate their customers from legacy, perpetually licensed products to subscription-based alternatives with new and predictable recurring revenues. To force their customers' hand, vendors concentrate their product roadmap on the subscription-based ERP replacements. The alternative for customers is to take matters in their own hand, and extend their ERP or other Systems of Record with innovative new features in external (but integrated) layers.

Moving development to the cloud

In the past 10-15 years we have seen a major evolution in moving enterprise applications and processes to the cloud. For enterprise software vendors, this meant new and recurring revenue, from both existing and new customers. This movement spurred the necessity for entirely new, **cloud-native application development** approaches and toolsets.



One of the relatively new approaches in cloud-native application development has been the emergence of low-code/no-code development tools. For organizations, this provided a way to quickly fill gaps in their application portfolio, given the ease of development. With intuitive graphical user interfaces and task wizards, the ease of low-code/no-code development allowed non-trained 'citizen developers' to quickly address a specific gap or need in the enterprise application portfolio, without putting a strain on an often overloaded development team. But low-code/no-code comes with disadvantages as well:

- Without proper governance, organizations run the risk of allowing **Shadow-IT** to emerge, with an uncontrolled wild-growth of point applications that all contribute to ever-growing runtime subscription costs.
- Low-code/no-code applications tend to have **limited functionality** for tackling rich user interfaces and complex business processes or algorithms.
- A **major disadvantage** of virtually all available low code/no-code platforms is the fact that organizations not only pay user licenses for using the platform during the development cycle, but also have to pay runtime license fees for end users of the published application.

Cloud-native Application Development

Organizations have started to embrace the greenfield aspect of cloud-native enterprise applications - along with the switch of capital investments to operational costs. Consequently, application development tools will need to support the development of cloud-native applications, while still allowing for on-prem or hybrid deployments. This principle also applies to the development tools themselves: cloud-native development tools with subscription licensing provide predictable operational costs, with the prospect of continued innovation.

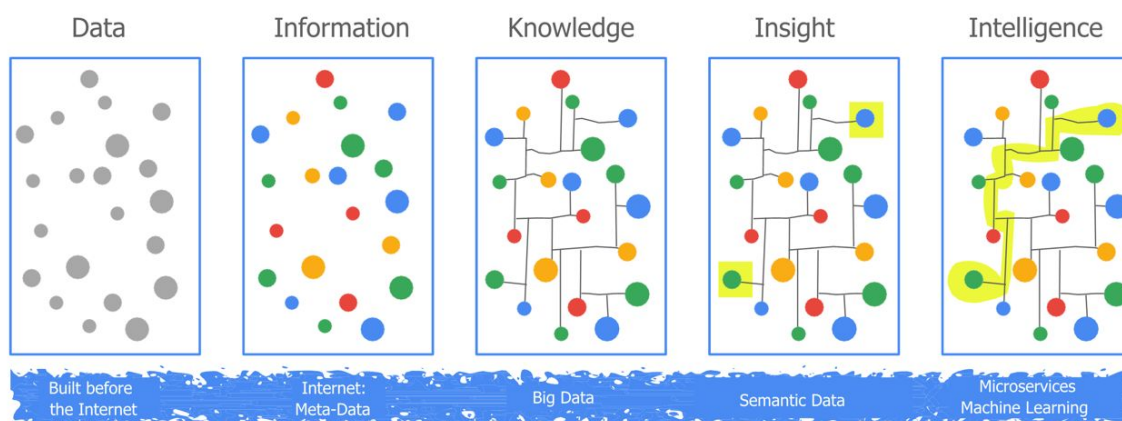
Moving data to the cloud

Besides innovations in cloud-based development tools and processes, the past two decades have seen a gradual but steady migration from on-premise data storage to the cloud. For enterprises, this provides the opportunity of **modernizing legacy data** with cloud-based tools for which on-premise alternatives tend to be too costly and complex.

Modernizing Legacy Data

Legacy data does not mean useless data. But it can be difficult and costly to access data in legacy databases. With a sporadic or more frequent need to access legacy data, it may make sense to transfer that data from legacy silos to a modern cloud database. This provides the inherent opportunity to deploy modern analytics tools on these data sets. If only sporadic access is needed, cloud databases can be configured to spin up on-demand to minimize runtime costs.

A logical progression and additional benefit from moving data to the cloud is that it provides new opportunities to **turn data into intelligence**. Traditionally, extracting intelligence from data required specialized - and costly - databases, software tools and hardware to handle the computational requirements. By moving data to the cloud, enterprises get access to cloud-based services for big data, machine learning and artificial intelligence. This enables organizations to turn disorganized data into intelligent information to fuel their business.



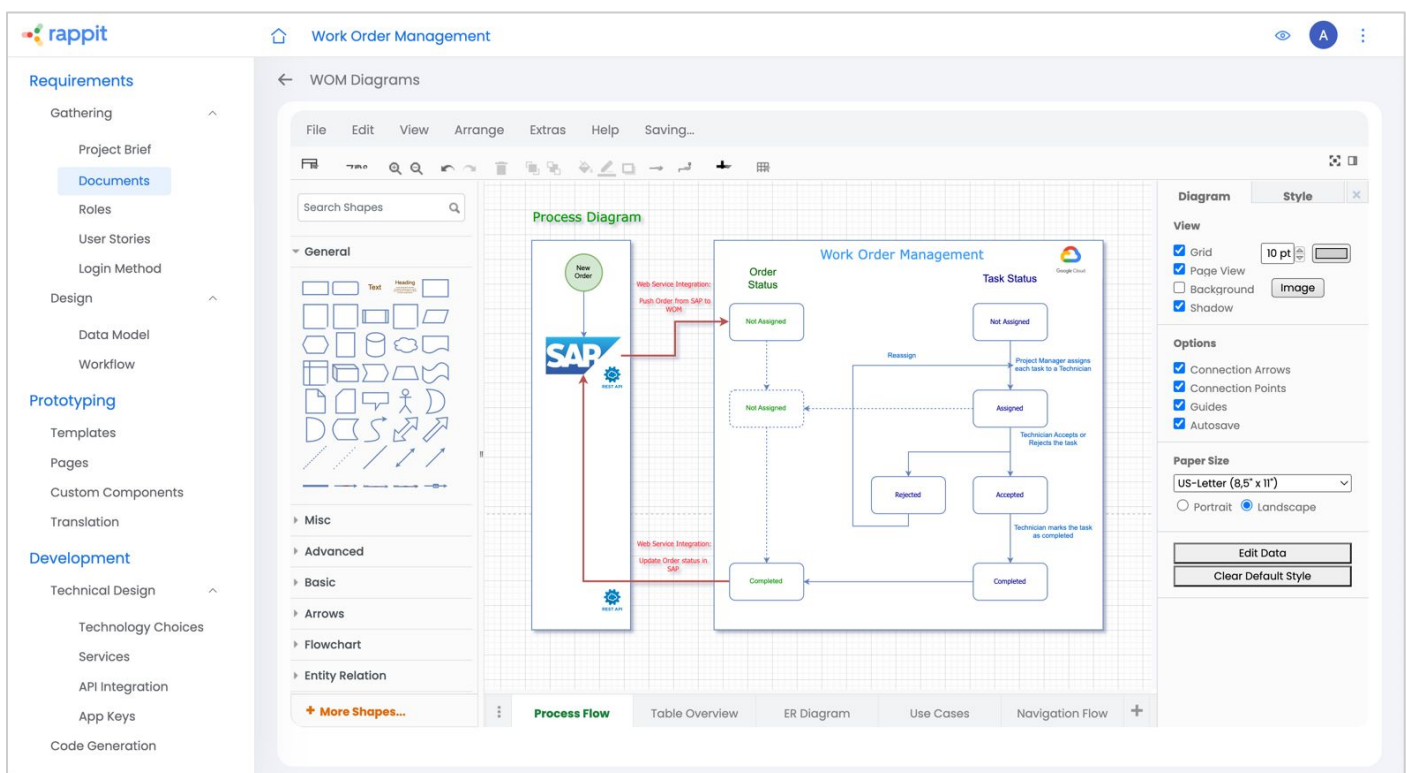
Turn Data into Intelligence

Enterprise applications generate a lot of data. Standard reporting tools typically don't go beyond factual summaries. But your data may contain a wealth of hidden information that could be used to improve and optimize the business, and potentially uncover cost savings or new revenue opportunities. It takes experience, skill and the right tools to extract intelligence from data, such as machine learning, artificial intelligence and real-time analytics.

Vanenburg Rappit Developer

Vanenburg has its roots in early Enterprise Application development - starting in the late 80s with the first workstation based predecessor of a full-blown ERP system. This not only gives Vanenburg deep experience with enterprise application development, but also cemented our understanding of what organizations need to support their business processes with effective, value added software. At Vanenburg, we recognize that full-blown enterprise software upgrades or replacements are not only costly and complex, but often unnecessary. A far more cost effective approach - with a much **faster return on investment** - is to selectively update, replace or add specific business software functions with external, but fully integrated add-on components.

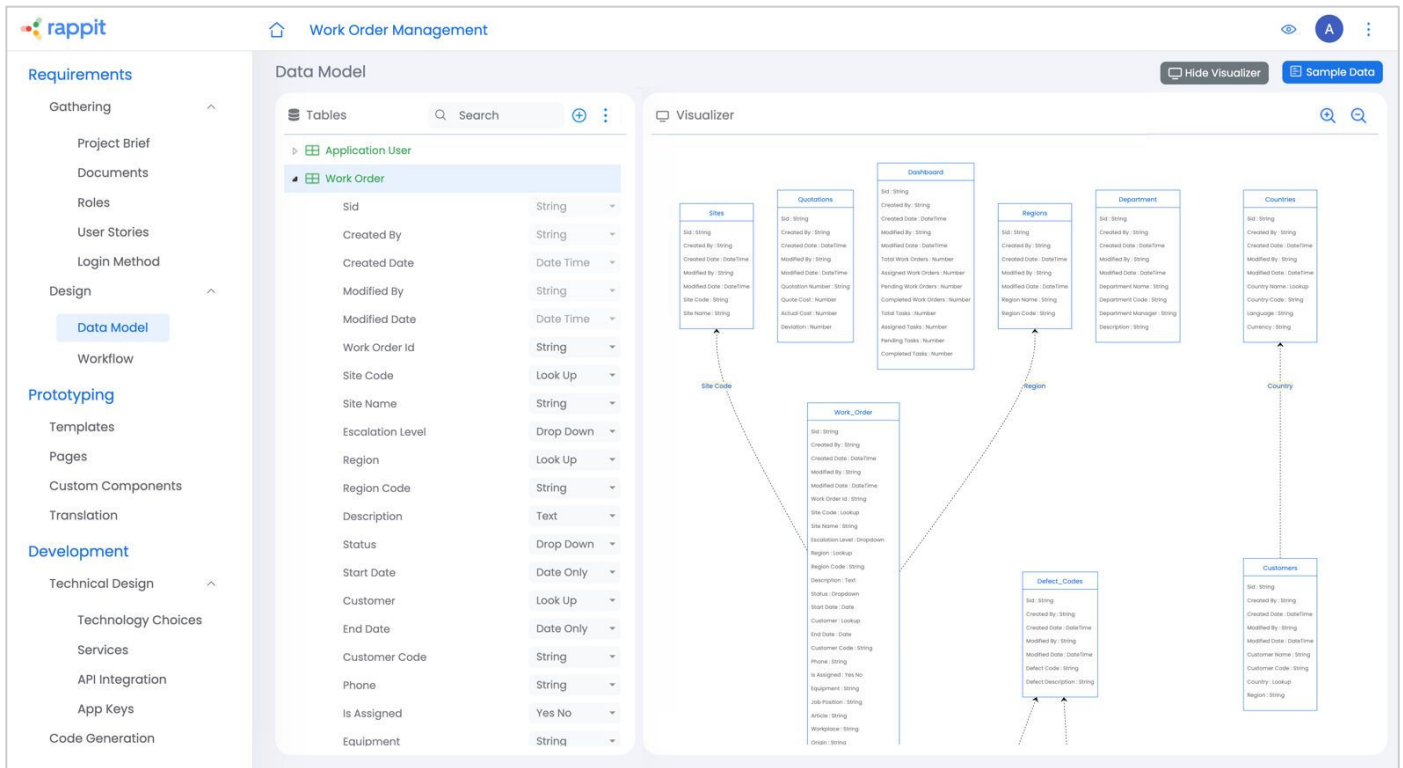
Vanenburg's Rappit Developer can be used to develop cloud-based add-ons to existing solutions, as well as entirely new enterprise applications. Rappit Developer adopts the ease-of-use features of low-code/no-code tools during the prototyping phase, but combines that with the full power of native Java-based software development by experienced developers.



Rappit Developer Studio - Requirements Gathering

Rappit Developer guides its users through the entire process of requirements gathering and documentation, to workflow and data model design, UI generation, CI/CD from test through production and end user testing. For typical enterprise application extensions with their own data model, user interface designs, workflows and integrations, a consultant role can **generate up to 30-80% of the application code**, which is stored in a Gitlab repository. Experienced developers subsequently finish the application by adding specific functional logic using their IDE of choice.

Rappit Developer is hosted in Google Cloud, and has no local installation requirements. Consequently, it does not burden the customer's IT organization with maintenance. Applications created with Rappit Developer can be deployed anywhere as Java or Web archive files (JAR/WAR), or packaged as a Docker image for deployment in container engines such as Google Kubernetes Engine (GKE).



Rappit Developer Studio - Data Model Design

As a major differentiator compared to **low-code/no-code** or other accelerated application development tools in the market, solutions developed with Rappit are fully owned by the customer and **do not require runtime licenses**. All generated code and the resulting applications are managed by the existing IT organization and fully accessible to- and owned by the customer.

Vanenburg Rappit Developer is the ultimate tool for **modernizing enterprise applications** and **high productivity coding** of cloud-native core systems and add-ons.

Low-Code / No-Code

In the past decade, many enterprises have adopted Low-Code/No-Code platforms to address skilled developer shortages. While these platforms are certainly capable of 'filling the functional gaps', they also introduce additional organization challenges and costs. Commercial LC/NC tools not only require license fees to create an application, but also charge runtime license fees for deployed applications. In addition, LC/NC tools typically lead to 'Shadow IT', with uncontrolled deployments of point-solutions that lack life-cycle management.

Rappit Developer and Artificial Intelligence

Artificial Intelligence will play an increasingly important role in application development. Rappit Developer is ideally positioned to make full use of Generative AI capabilities as the technology matures. Currently, developers already have the ability to add custom coded functionality to Rappit Developer applications, and such code can optionally be generated with external AI tools. With GitLab bringing AI to DevOps, developers have an integrated experience in utilizing AI to generate custom code for use in Rappit Developer projects.

Below are two examples of how AI functionality can potentially be applied in future Rappit Developer releases:

AI/ML Intelligent Applications

In this scenario, applications generated with Rappit could have AI/ML capabilities embedded inside them. For example, when developing an eCommerce application with Rappit Developer, a developer would use Generative AI to automatically generate useful product descriptions. The "Product Description" field is defined as AI-generated, along with a selection of related fields and optional manual user inputs that together drive the AI/ML model. Rappit Developer would subsequently generate the application source code plus the ML code, and create and deploys the application and the ML model. This process greatly improves and accelerates the development of applications without requiring the developer to be an expert in AI/ML. In addition, it takes the burden of creating hundreds or perhaps thousands of product descriptions away from the end users.

AI-Assisted Development (AIAD)

A second use case of Generative AI with Rappit Developer could directly benefit consultants and developers. For example, if a consultant creates a new 'Expense Claim' application project, Rappit Developer could propose a sample Expense Claim application. If the user accepts the proposed app, Generative AI would be used to create an application model in Rappit Developer with fully defined tables, fields, workflows, and a User Interface definition. Upon accepting or tweaking that model, Rappit Developer subsequently generates the code, builds the app, generates a test data set, and deploys the prototype app for testing and validation.

By utilizing AI capabilities to complement Rappit Developer, experienced application developers can focus on value-added functionality and custom coding, with quicker results.

Artificial Intelligence

"AI will not replace human workers. Human workers utilizing AI will replace human workers not adapting to the latest trends in AI."

Why Vanenburg

- 40+ years of experience in the Enterprise Software market with deep expertise in ERP and integrations to back-end systems
- Dedicated Google Cloud Practice since 2014 with 150+ FTE
- Unique 'Rapid Application Development' tooling to quickly prototype and develop applications at large scale against low Total Costs of Ownership
- Fully automated cloud hosting & management service based on "Infrastructure as Code" approach to provision and manage (24 x 7) cloud resources and applications
- Strong artificial intelligence capabilities based on Google AI

Interested? Contact us at info@vanenburg.com to learn more or schedule a demo.

vanenburg

Vanenburg is a leading independent software provider. Building on a proud heritage, we are experts in developing Cloud based Enterprise Software modernization solutions, that include Custom Made Applications, Data Analytics, Artificial Intelligence and Machine Learning. Based on Rappit, our application development suite consisting of amongst others Rappit Developer and Rappit Composer, we enable our customers to create unique business value.

Vanenburg serves a customer base consisting of multinationals and medium sized organizations, mainly in the Manufacturing, Supply Chain & Logistics and Retail & Wholesale Industries. Our Rappit Suite also supports ISV's and System Integrators in building world-class applications.

For more information, please visit www.vanenburg.com

Follow us:

