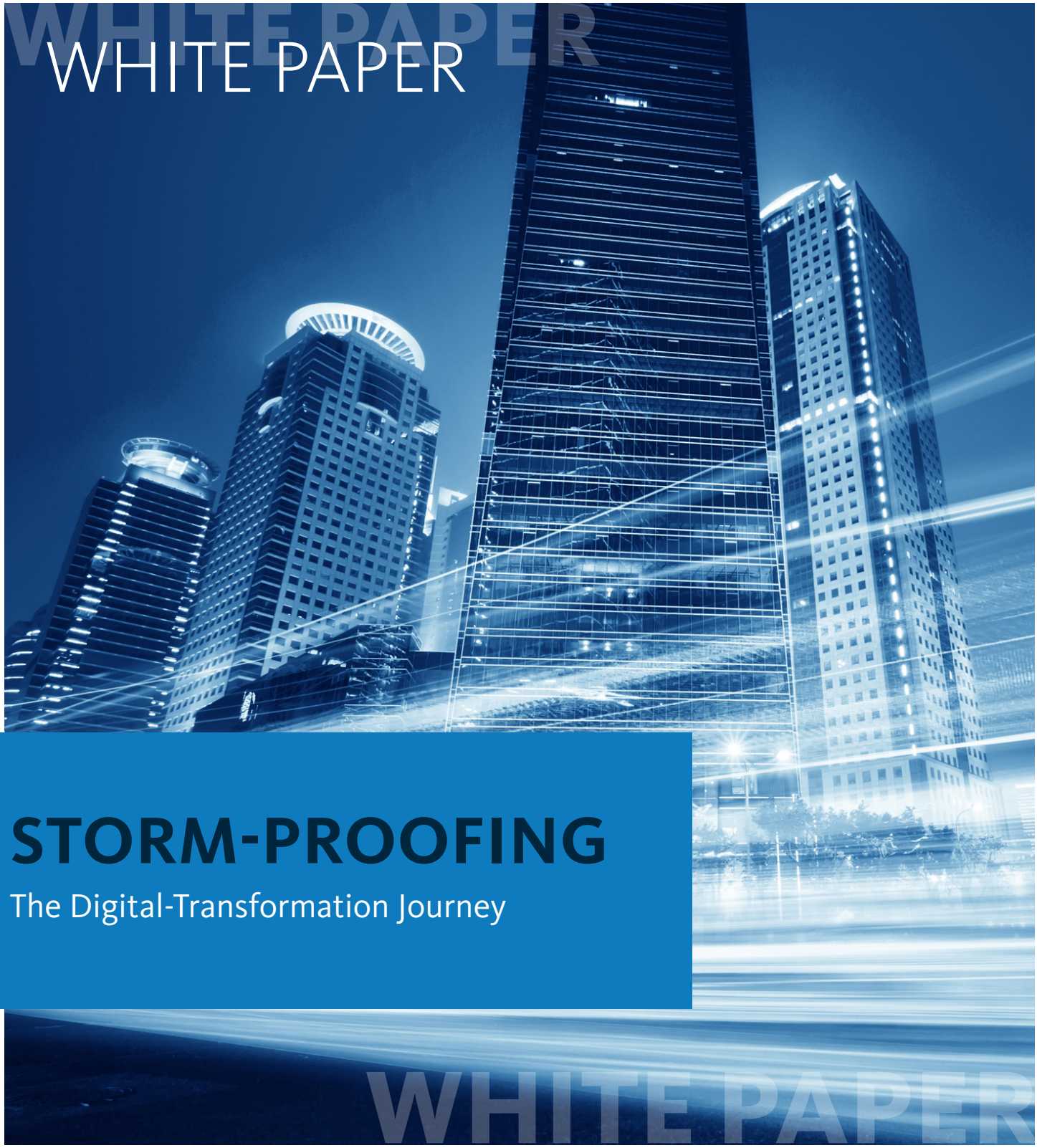
WHITE PAPER



# STORM-PROOFING

## The Digital-Transformation Journey

WHITE PAPER

AGILEPOINT

The Industry's First Future Proof Digital Business Platform

www.agilepoint.com | info@agilepoint.com | +1.650.968.6789

# Table of Contents

# The Two Technological Ages

"Digital Transformation" began with the first punch-card computers and has been going on ever since. Given this fact, many—even within the IT industry—may be fuzzy on digital-transformation mania, which is sweeping the corporate world. To really understand one of the most important technology trends of the decade, one needs to take a quick look at the history of digital technologies, which can be divided into two periods.

## 1. The Information Age

During the Information Age, new information technologies emerged at a pace that enabled organizations to stay abreast.

The general market for each of these technologies emerged and matured along predictable, technology lifecycle paths. And while many of these technologies overlapped, still, organizations en masse were able to incorporate them on a global scale and at a reasonable pace.

## 2. The Digital Age

In contrast to the Information Age, the Digital Age is marked by greatly accelerated innovation. Wave after wave of new technologies are now hitting organizations—the Internet of Things, Big Data, AI, Machine Learning, Block Chain, and on and on—all converging at once.

This escalating pace is illustrated in the chart below:

This dramatic increase in the pace at which innovations hit the market is the catalyst of the digital-transformation phenomenon, which, to many, is the incorporation of new digital technologies with existing IT systems and infrastructure in an effort to remain competitive. But if this is the nature and goal of Digital Transformation, then those who attempt it have cause for concern.

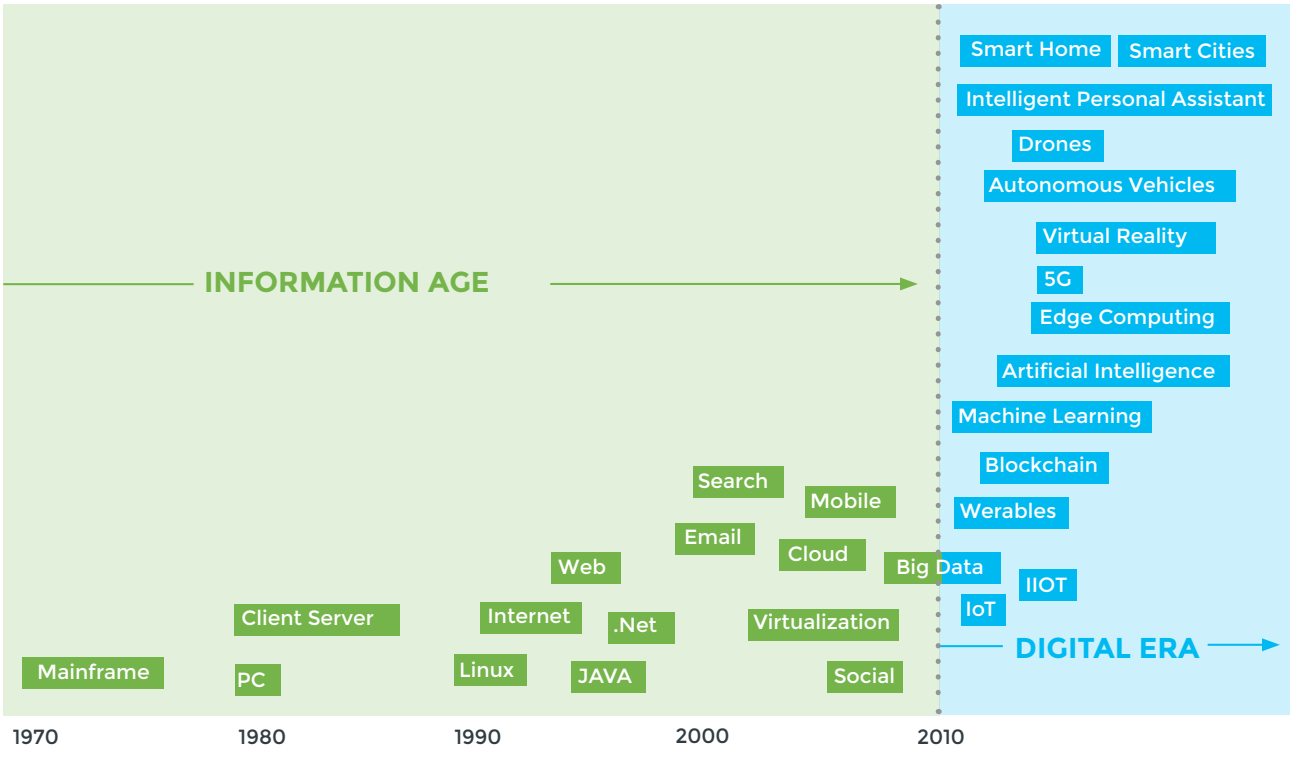



**Figure 1.** Timeline view of key technologies

## 3. The Digital Storm

The simultaneous convergence of disruptive innovations constitutes a sort of digital storm, and, for many organizations, it's causing some level of organizational chaos. Technical debt, accrued over years of band-aid coding is a big part of the prevailing weather system. The gradual adoption of new cloud-based services is another. Just getting it all to work together dominates IT resources for most companies. And exacerbating modernization and integration is the deluge of new technologies that needs to be accounted for in any organization's IT roadmap.

For many, the urge to just hunker down and wait out the storm is compelling—modernize the old stuff; integrate it with the new, cloud stuff; and worry about all the bleeding-edge stuff later. But there are several critical problems with this strategy.

### A. The New Normal

Unlike actual storms, which always end, the digital storm may not. In fact, it's hard to imagine a scenario whereby the pace of digital innovation will do anything but accelerate. In other words, storm conditions are the new normal.

### B. The Customer Factor

Customers have been widely exposed to consumer apps that require virtually no training, are augmented by AI, plugged into the IoT, and work seamlessly across devices and channels.

Unfortunately, customer expectations ratchet in one direction only, and that's up. Failing to meet today's customer expectations—as well as keep up with evolving ones—could easily result in mass migration to vendors that provide a better overall user experience.

### C. Digital Darwinism

Ray Wang, founder of Constellation Research and author of the book "Disrupting Digital Business: Create an Authentic Experience in the Peer-to-Peer Economy" (2015), sums up the wait-out-the-storm **strategy**:

. . . the impact is significant and now quantifiable with 52 percent of the Fortune 500 gone since 2000 and the average age of the S&P 500 company in 1960 down from 60 years to a little more than 12 projected in 2020. That is a 500 percent compression that has changed the market landscape forever in almost every industry. Basically, Digital Darwinism is unkind to those who wait. In fact, you will be out of business if you **wait**.

### "Customer Behavior in the Digital Storm"

According to Entrepreneur, 80% of companies believe that they deliver superior customer service, yet only 8% of the customers agree.

The effect of this alarming statistic is made clear in a Customer Experience Impact report by Harris Interactive, which states that "89% of US adults who have ever stopped doing business with an organization due to a poor customer experience" migrated to a competitor.

McKinsey's research augments these studies, stating that 70% of the buying experience is based on how a customer feels that they are treated.

## 4. The Digital-Transformation Conundrum

Multiple studies show that garden-variety change initiatives fail at a rate hovering around **70%**. But the challenges presented by the digital storm are anything but ordinary. Consequently, industry analysts predict the failure rate for digital-transformation will be much higher. Forbes, for example, puts the number at a staggering **84%**.

The question, then, that all business and technical decision makers must ask is this:

> *"With legacy modernization, cloud-service adoption, and the deluge of new technologies that must be accounted for all hitting at once, how do we effectively transform while the storm is constantly re-arranging the IT topography?"*

On one hand, if you begin a transformation initiative, now, and the digital landscape undergoes radical change before you're finished, your investment could be wasted. On the other hand, if you wait until there's more visibility, you'll become a digital-storm statistic, as Wang predicts.

## 5. The Need for Future-Proof Applications

In concrete terms, digital transformation, for most organizations, will involve radically more software development, including the following:

- Dozens to hundreds of back-office applications that enable cross-functional collaboration across departments
- Hundreds to thousands of apps that address every touch point for every customer at every phase of their journey, from the sales cycle through long-term maintenance, and which interact seamlessly with back-office systems
- Thousands of automated processes that streamline operations, fortify governance, and institutionalize customer experience

The challenge is developing all this new software in such a way that it won't become a legacy burden, needing to be re–architected with each new innovation or change. In short, for digital transformation to be successful, all of this new software must be impervious to storm conditions; in essence, it must be Future-Proof.

# Critical Characteristics of Future-Proof Apps

The need to build new applications fast has catalyzed the emergence of Low-Code Development Platforms (LCDPs), which can speed development as much as ten times. Likewise, LCDPs across the board use a drag-drop-and-configure approach to application development, making development much easier, given that non-programmers can build functional applications.

But in the context of the digital storm, fast and easy isn't good enough. Applications need a range of characteristics that will enable them to continue working when inevitable change occurs.

Following is a list of critical characteristics that make applications and the platforms that produce them largely "future proof."

## 1. Architecture

Historically, enterprises have re-architected existing applications over and over. Common examples include movement from desktop-based applications to browser-based, client/server applications; change from a monolithic architecture to a layered architecture; and, more recently, migration to containers and a micro-services architecture. This practice of re-architecting applications is costly, time-consuming, and, in itself, is a contributing factor to the high failure rate of transformation initiatives. Future-proof applications must have an architecture which can adapt to changes of all types and can be extended easily, without the need to recode, enabling enterprise to better focus on business.

## 2. Manageability

Most enterprises run operations through web-based business platforms, such as SharePoint, O365, Dynamics CRM, Salesforce, SAP, NetSuite, and so on. To interact with such platforms without continuous breakdowns, a future-proof platform must separate the business platform, application, and data onto separate layers. With this approach, an organization can manage and access each layer separately. This layering approach facilitates easy migration, given that switching from one business platform to another will not impact applications, which exist on their own layer. Additionally, enterprises can make the same application available on multiple business platforms in parallel.

## 3. Deployment

Enterprises struggle when it comes to migrating from one environment to another. Consequently, future-proof platforms must be capable of deployment virtually anywhere, including on premises, in private clouds, in public clouds, and in hybrid environments. A future-proof platform should enable porting from one environment to another in minutes or hours, rather than the current benchmark of months.

## 4. Embeddability

Future-proof applications must be embeddable in virtually any business-application platform, such as SharePoint 2010/2013/2016, SharePoint Online, Dynamics CRM, SAP, Salesforce, NetSuite, etc. Furthermore, the same user experience should be available regardless of where an application is surfaced, thereby eliminating expensive workforce retraining.

## 5. Devices and Browsers

A wide range of web browsers (Chrome/Safari/Internet Explorer/etc.), operating systems (iOS, Android, Windows 10/8/8.1/7/Mac OS/Linux) and devices (tablets/mobiles/laptops/desktops/wearables) can be used in almost limitless combinations. Building individual apps for each conceivable combination would, of course, be impossible. Consequently, future-proof apps must automatically adapt to any combination.

## 6. Mobile Platforms

Enterprises face multiple challenges in rolling out native mobile apps. For example, enterprises are forced to setup development teams for different platforms (Android, iOS and Windows Mobile), and procure build servers, mobile devices, and other infrastructure components.

Teams need to manage app submissions to stores, configure apps for enterprise users, and continuously rollout upgrades to users. A future-proof platform must simplify these challenges, enabling the rapid development of mobile apps that will run on iOS and Android devices as well as any new mobile platforms that may enter the market.

## 7. Upgrades to Business Application Platforms

Enterprises often create custom applications on commercially available business platforms, such as SharePoint Online, Dynamics CRM, SAP, Salesforce, NetSuite, and so on. Such business platforms are typically upgraded many times a year. (Cloud services may be updated as often as weekly, in some circumstances.) Future-proof apps must be largely impervious to updates, a realistic outcome given the layered approach described in "Manageability," above.

## 8. Downtime

Many business users resist and even fear updates and upgrades, which can often require additional training and unpredictable downtimes. These downtimes might last for hours and, in some cases, days. Future proofing should eliminate update-related downtime, both at the application level and at the application platform level.

## 9. Scalability

Application-platform scalability has to align with business growth. Consequently, a future-proof platform must enable application usage to scale up or down, given the unpredictable nature of many businesses. Application parameters that need to scale include the following:

- **Process Size:** Process-based applications can be extremely complex, involving thousands of steps and dozens of sub-processes. A Future-Proof platform should be capable of building and deploying applications of virtually any size and complexity.
- **Number of Apps:** A digital-transformation initiative could push the size of an organization's application portfolio to tens of thousands. A future-proof platform should be able to handle thousands of applications and millions of running application instances simultaneously.
- **Application Bandwidth:** As an organization grows, application usage could conceivably go up exponentially. Future-proof apps must be capable of scaling from hundred users to millions, without being re-architected.
- **Scalability at Platform level:** By design, the platform itself must be scalable to support growth in the number of users, data, events, and processes.

## 10. New Technologies

Given the nature of the digital storm, it's a virtual certainty that organizations will want to incorporate new technologies—AI, Block Chain, etc.—into existing applications. A future-proof platform must enable continuous incorporation of new and upcoming technologies.

## 11. Reliability

Mission-critical processes enable core business operations. An example of a mission-critical process might be remittance of funds to a foreign account. Such critically important processes must work 100% of the time, never deviating from the expected outcome. Circumstances that might disrupt mission-critical processes include system reboots, platform upgrades, and power outages. A future-proof platform should virtually eliminate the possibility that mission-critical processes would be disrupted for any reason.

## 12. Open Standards

With open standards , there are two factors to consider: application design and application execution.

- **Application Design:** A future-proof platform will not force users to learn a proprietary scripting language.
- **Application Execution:** During execution, a platform would communicate with other systems. A future-proof platform must utilize open standards, such as REST or SOAP.

The more of these characteristics a platform incorporates, the more it will enable extensive, sustainable digital-transformation in today's digital-storm environment.

# AgilePoint NX—The Industry's First Future-Proof Digital-Transformation Platform

The depth and breadth of digital-transformation requirements demand both the power of an enterprise-class BPMs (for a small number of "deep," highly complex processes that can take months to build) and the speed and agility of a low-code platform (for a large number of lightweight apps that address countless needs within an organization and which automate every touch point in the customer journey and sales process.).

Platforms that include features of both BPMs and Low Code are capable of what Forrester Research **calls** Digital Process Automation (DPA), which is the extension of your automated processes to customers, suppliers, and partners. DPA empowers you to act proactively, respond quickly, and generally provide an improved customer experience.

AgilePoint NX is a powerful DPA that includes virtually all of the future-proof characteristics described above.

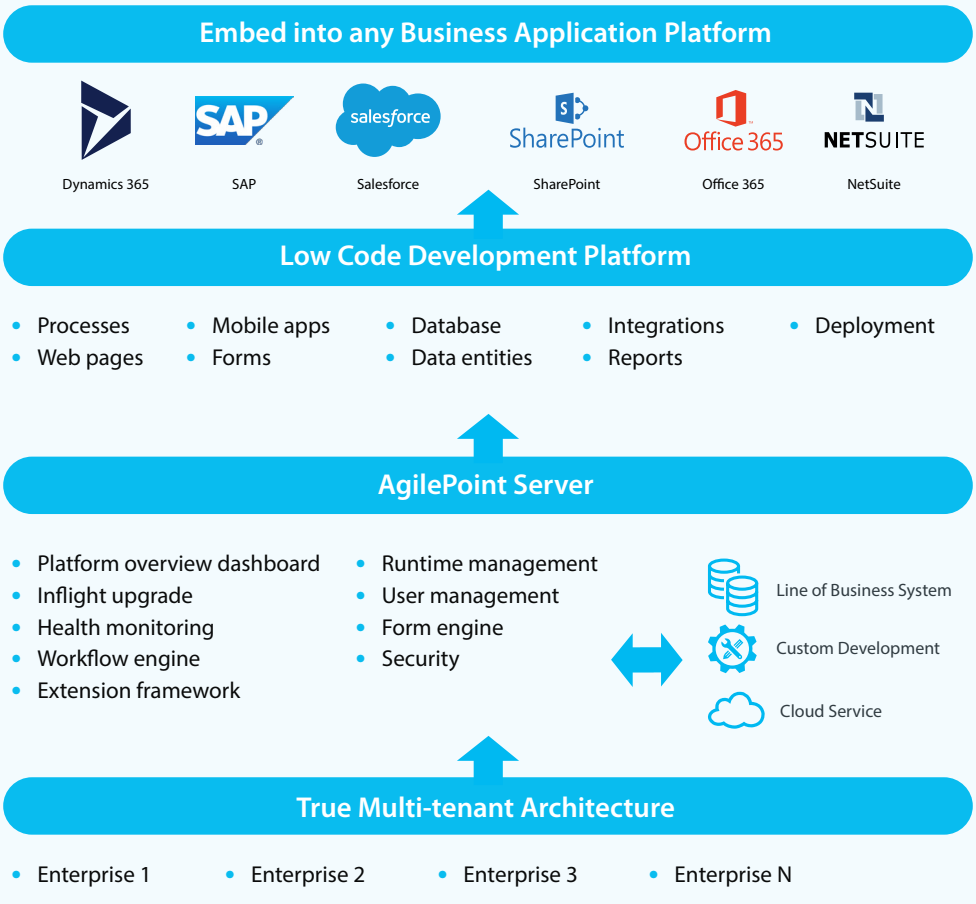



**Figure 2.** AgilePoint NX – The world's first future proof digital transformation platform

# 1. Process Engine

The core component of both a BPM suite and a Low-Code process platform is the process engine, and the power and sophistication of the engine are closely related to the overall capability of the platform itself. AgilePoint NX's process engine is a key aspect of its future-proof architecture and has several defining characteristics:

## A. Scalability

AgilePoint NX utilizes a stateless process engine, meaning an entire process is not loaded into memory during execution. Rather, only the active process tasks are loaded. This design ensures optimal memory usage. This stateless characteristic dramatically limits the amount of system memory required.

Furthermore, the AgilePoint NX engine supports automated processes of almost any size, a necessity with back-office applications that can easily include many hundreds of tasks. Likewise, the AgilePoint engine does not restrict the number of automated processes organizations can build, nor does it limit the number of versions of any particular app. This aspect of scalability is critical to a future-proof platform, given that transformation could easily require thousands of process apps, many of which will have multiple versions. Finally, the number of concurrent process instances running in NX scales based on the underlying hardware infrastructure. This feature is especially critical to large enterprises.

## B. Reliability

The fact that AgilePoint's stateless engine does not hold entire processes in memory during execution dramatically reduces the possibility of data loss in the event of a system outage. Consider, for example, a process that has fifty steps. Now suppose that there was a power outage while executing step number 23. None of the data from the first 22 steps would be lost—it has already been written to disk. When the system comes back, AgilePoint NX will be reloaded, and the process will resume at step 23.

## C. Hyper-Agility

IDC predicts that enterprises will shift toward a hyper-agile architecture by **2021**, a phenomenon driven by the fact that businesses need ever more sophisticated process apps to meet escalating customer demands.

To this end, AgilePoint NX was equipped with a highly adaptable process engine. The company's vision of AI driven task routing resulted in an engine that supports both manual routing and programmatic routing of tasks. With AgilePoint NX, process models are stored in XML, which is fed into the process engine at runtime. Either manually or programmatically, processes can skip back to rerun previously executed tasks or jump forward to bypass tasks. Likewise, tasks that are not present in the model definition can be created on-the-fly.

With AgilePoint's engine, live processes in mid-execution can be upgraded, or even downgraded, without restarting any component. Note that the AgilePoint NX platform, itself, can also be upgraded, without downtime. And AgilePoint NX's engine allows users to run different versions of the same process in parallel.

AgilePoint NX's engine enables the building of composite applications. A composite application draws functionality from multiple sources. This composite approach enables easy integration of new and upcoming technologies. Furthermore, the integration of new technologies into a composite app does not require the app be re-architected or reengineered. Other benefits of the composite approach are software-component reusability and quick time-to-market.

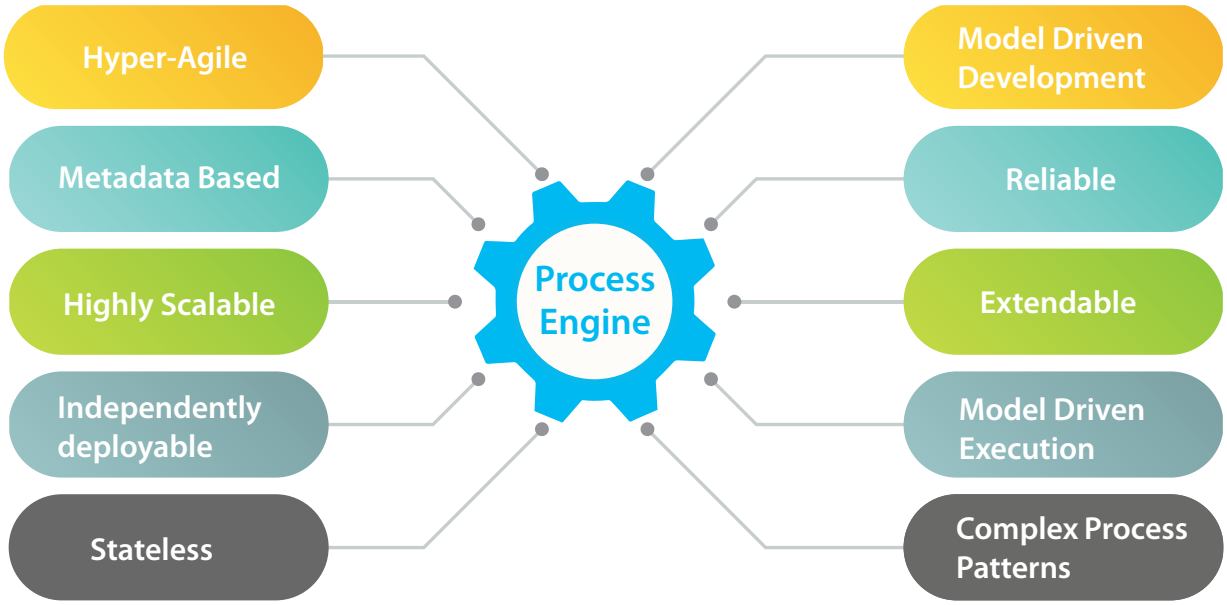



**Figure 3.** Key feature of AgilePoint NX Process engine

### D. Extendable

AgilePoint's process engine has an event-driven architecture that can be extended by developers. Every activity in the engine triggers events which, can be subscribed to, to alter the engine's behavior. For example, engineers can code and override the default exception handling, as per organizational policies. Likewise, an organization can create its process actions.

## 2. A True Low-Code Development Platform

With most low-code platforms, business users compose applications by dragging activities, forms, and other types of controls onto a canvas and then configuring each to application specifications. The result is an application model—a visual construct that depicts flow as well as inter-relationships between application components. AgilePoint utilizes this same declarative approach to app composition but is unique in how it processes application models. To fully appreciate the AgilePoint NX approach, you first have to understand the alternative.

### A. Code Generation

Some platforms take application models through a code-generation process, converting the model to computer code. This code is then compiled, linked, and executed, just as any hand-coded app would be. The biggest downside of the code-gen approach is that the finished application consists of actual code. In other words, an app generated by a code-generation-based system has just as much code as an app that was hand coded to begin with. And applications that consist of actual code tend to have baked-in feature sets, exposing them to the disruptive elements of today's storm environment.

### B. Metadata Abstraction

In contrast to code-generation, AgilePoint NX utilizes metadata abstraction to dynamically rewrite pre-configured components that reside in the AgilePoint NX process engine, including various components of the process and forms.

In plainer terms, with AgilePoint NX, each visual component placed in a model generates metadata.

For example, as a developer drags an activity into a model, the relative positioning of the activity to other model components constitutes metadata. Likewise, flow lines between components constitute metadata, and all of it, taken in totality, defines in real-time how the model gets executed.

Likewise, AgilePoint's usage of metadata simplifies the design process. For example, with AgilePoint NX's Data Mapping utility, mapping entity fields from one entity to another is greatly simplified. Consider a scenario where the same data field in two different entities had a significantly different name, making the mapping process more difficult. AgilePoint NX would assist by comparing metadata (say, the data type of each field) related to the fields to make a correct match. Likewise, while designing a form, when an entity or a variable is dropped on the canvas, the control that gets generated is based on the metadata of the entity or the variable.

Why is this metadata approach so powerful? In short, because it overcomes the limitations of code-generation. Put another way, by changing the model itself, a user is, in effect, modifying the application. This approach enables in-flight upgrades of live processes without any downtime. The execution of the model at runtime is based on metadata interpretation, which is more flexible than code written in a programming language. This type of execution requires less CPU resources than code-based execution. The net effect of this metadata-abstraction approach is that AgilePoint NX apps have the genetic capability to adapt—even self adapt—to new environments, systems, new versions of systems, and even to new technologies. In other words, the AgilePoint NX platform was designed to yield storm-proof software assets.

## C. Components

NX allows users to build, test, and deploy all the components of the application stack through drag-drop-and-configure techniques or via a click-to-add process. Stack components include data entities or business objects, database tables, integrations, workflows, native mobile apps, web forms (eForms), and web pages. Data entities can reside inside NX or in different systems, such as Salesforce, SharePoint, and so on. Likewise, the integrations can be with the internal Line of Business (LOB) systems and with external cloud services.

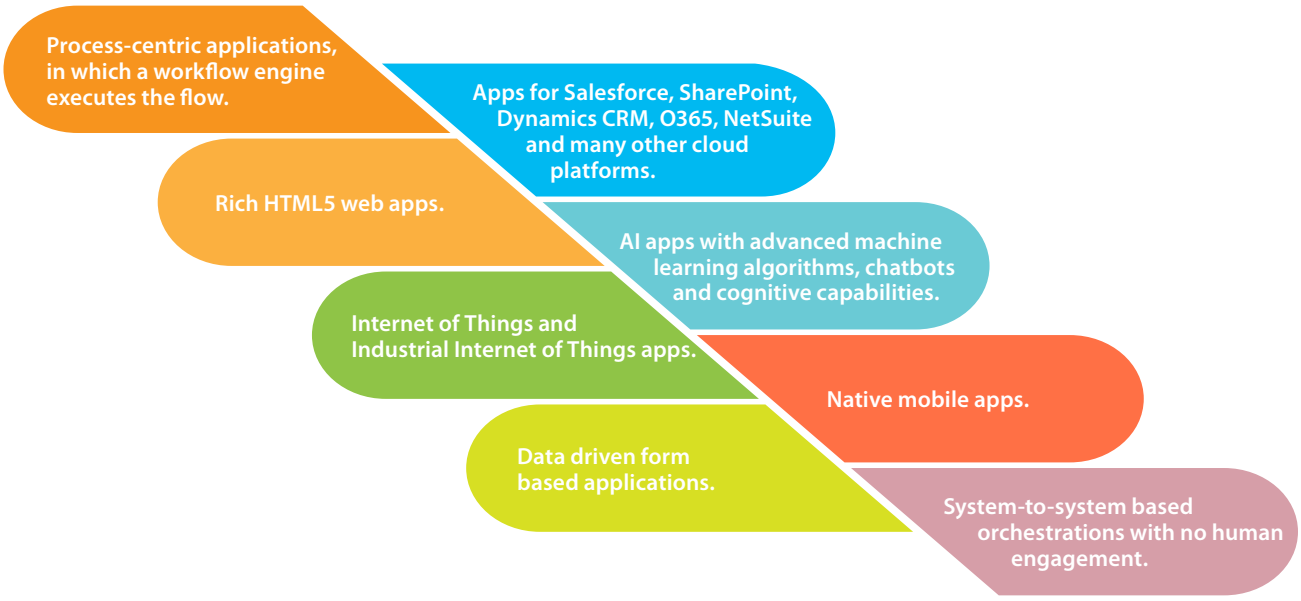



**Figure 4.** Enterprises can build a wide spectrum of application with NX

## D. Declarative Development

NX supports declarative, model-driven development and model-driven execution. As was explained above, an application model is interpreted at runtime, resulting in the execution of the application. This approach eliminates the need to generate, compile, link, deploy and execute software code.

With visual modeling and ready-to-use components, semi-technical business users, such as business analysts and domain experts, can generate applications quickly without depending on a software engineering team.

## E. Runtime Updates

Model-driven design and execution is yet another aspect of the hyper-agility discussed above, given that application models can be quickly updated, and changes can be immediately pushed to the running instances of the application via a single click and without restarting the instance.

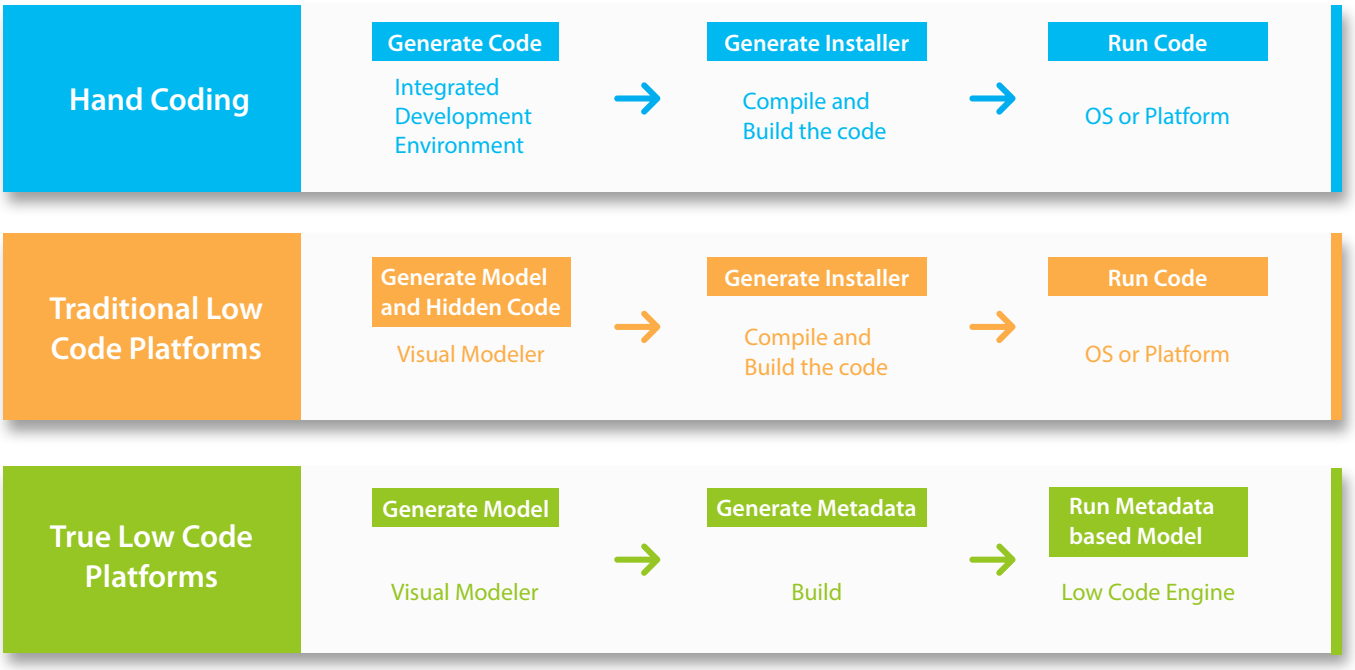



**Figure 5.** Metadata based, model driven development and model driven execution

## 3. Governance

For any LCDP, strong governance capabilities are critical, given that users are often spread across an organization, rather than being restricted to the engineering department. AgilePoint NX provides granular access controls and rights management for all the features at the overall system level, at the individual tenant level, and at the individual app level. AgilePoint NX system administrators can control 85+ permissions to users and groups in the system. AgilePoint NX is shipped with pre built roles such as application designers, business users, etc.

## 4. Manageability

AgilePoint NX organizes applications into three layers:

- **Business layer** (presentation layer), where users perform routine tasks
- **The process layer**, where the actual process execution happens
- **The data layer**, where the business data resides

These three layers can be deployed, accessed and managed independently.

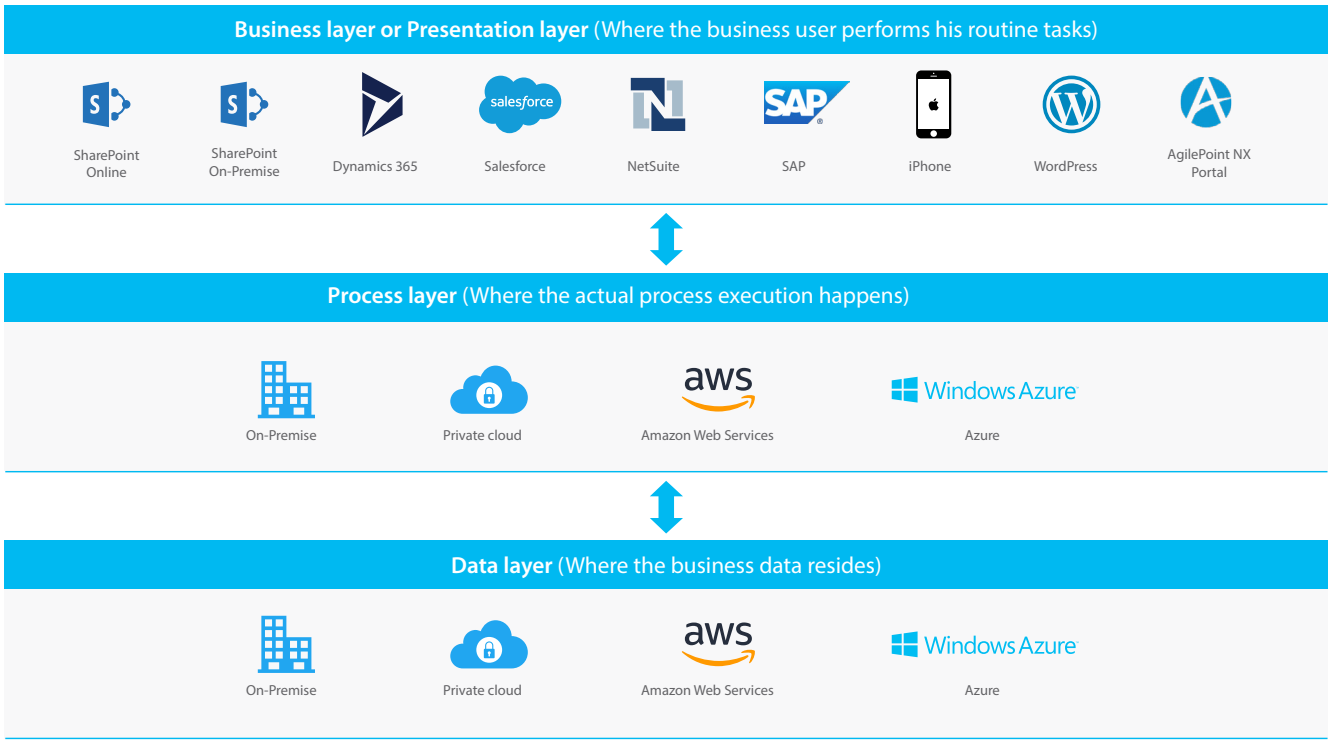



**Figure 6.** Manageability on NX

### A. Deployment Options

AgilePoint NX does not lock organizations into any type of deployment, supporting deployment on-premises, in a private cloud, in a public cloud, or in hybrid environments, all with a single code base that spans each. Users can migrate from, say, an on-premises installation to a public-cloud installation without refactoring applications.

### B. Self-Containment

AgilePoint NX, itself, runs in a self-contained portal. Users can access a fully-integrated set of application development and management tools in the same location. AgilePoint NX is a powerful, enterprise-class, BPM-enabled, general-purpose, Low-Code platform that provides a rich, desktop like user experience in a browser.

### C. Universality

Because NX is built with industry-standard technologies, such as HTML5 and CSS3, AgilePoint NX can be embedded in 50+ popular business platforms, including Salesforce, SharePoint, SAP, NetSuite, Dynamics CRM, and O365.

In short, any platform that can host custom pages is accessible by AgilePoint, allowing enterprises to add AgilePoint functionality to the systems their employees live in every day.

### D. Standalone Process Engine

Because AgilePoint NX's process engine resides on its own layer, the engine, itself, is standalone. In other words, when a process is triggered from the business layer, the process lives in the AgilePoint engine. The benefit of this approach is NX apps are largely unaffected when an integrated business platform undergoes an upgrade or is restarted. The same live process instance can simultaneously be accessed from different business platforms.

## 5. Application Development

NX provides a library of configure-and-use application components. These components connect and communicate with other systems, such as databases, active directory, web services, etc. through open standards. AgilePoint NX's User Interface (UI) is implemented using open standards such as HTML5 and CSS3 and is transmitted as Java Script Object Notification (JSON) files. NX does not utilize any proprietary scripting language.

### A. Integrations

AgilePoint NX 7.0 includes connectors to dozens of popular web-based business services. A sample of some of the integrations supported out of the box areshown below.

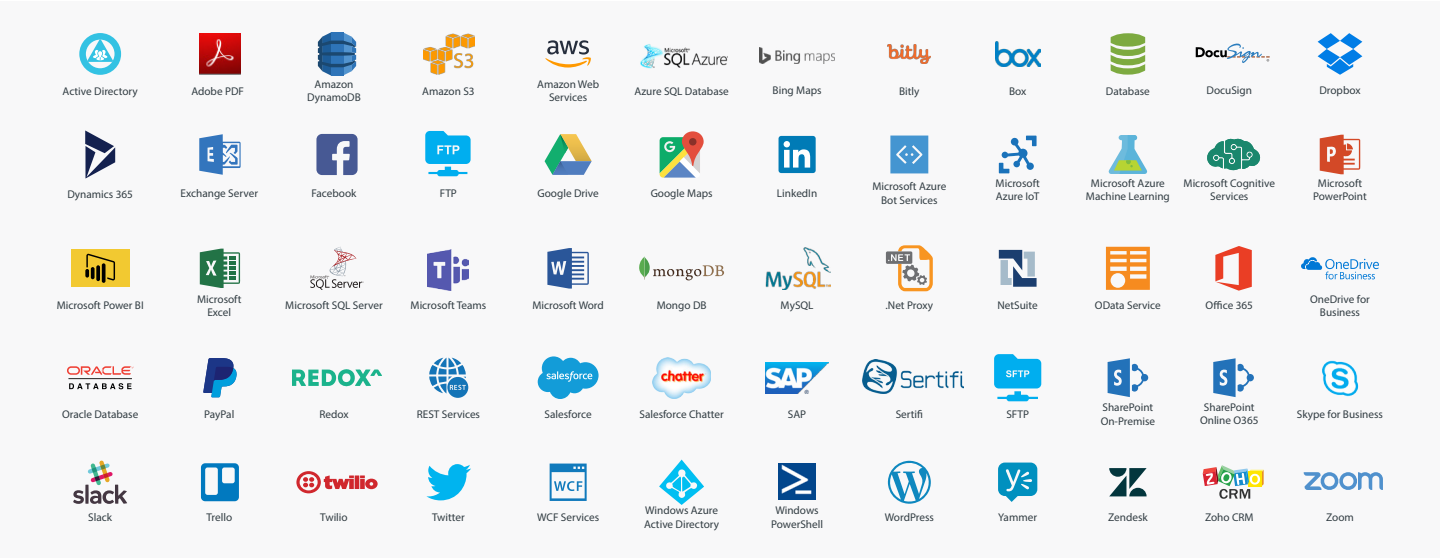



**Figure 7.** Examples of some of the Integrations provided by NX

There are three popular approaches to integrate with any cloud based application via the Internet:

- **REST (Representational State Transfer)**
- **SOAP (Simple Object Access Protocol)**
- **WCF (Windows Communication Foundation)**

NX supports all three methods, meaning that AgilePoint NX can integrate with any system on the internet that exposes its services via open standards.

## B. Form Builder

AgilePoint NX includes one of the most powerful form building utilities in the industry. Non-technical users can build powerful and engaging HTML5 forms, utilizing more than eighty control types, including auto-lookup, CAPTCHA, geo-location and geo-fencing, maps, credit card, star ratings, timer, data grid, HTML5 file uploads, and many more. NX forms work on all modern browsers such as Chrome, Firefox, Edge, Safari, and others.

AgilePoint NX enables users to design device-specific visualizations. The user can selectively include or exclude controls and page elements for various devices or resize and reorder controls for different devices. For example, a table or a grid in a browser could be displayed as a card view on a mobile device.

Forms built with Form Builder consist of HTML5 and JavaScript and can be embedded in any HTML5-based application. While Form Builder requires no coding, advanced users can work directly with a form's HTML and JavaScript, making the form extensible in any direction.


**Figure 8.** eForms powering your enterprise applications

Without deployment, users can perform live previews and validations for multiple browsers, tablets and mobiles (in both landscape and portrait) from within the designer. Most of the UI elements are made with CSS without images. So they look pixel-perfect on high resolution screens, such as 4K monitors and higher.

Users need not learn any scripting language to build UI in AgilePoint NX. Users can create dynamic and interactive forms by incorporating business rules to perform a number of actions, such as showing and hiding fields or skipping certain pages based on selections while filling out forms. This built-in validation capability ensures submitted information is accurate. One can group multiple rules or nested rules in a friendly way, which is easy to understand and maintain.

### C. Form as a Service (FaaS)

FaaS generates form-based applications from any SQL database (Oracle, My SQL, MS SQL Server, etc), AgilePoint NX's built-in data entities, Salesforce entities, and SharePoint entities. Form-based applications can be embedded into mobile apps, web pages, form pages, SharePoint apps, Salesforce apps, and more. These form-based applications can be used for managing data in the underlying systems.

FaaS enables users to create embeddable widgets, such as maps, lookups, etc. And users can rapidly replace existing custom and non-portable UI-based apps, such as InfoPath apps on SharePoint. AgilePoint's form-based apps are future-proof, given that the apps are not tied to any platform, device, or channel. As new business needs come up, the same apps can be made available in new channels.

### D. Data Entities

With Agilepoint NX, non-technical users can design business objects or entities via visual modeling. Database tables can be built easily using a wizard that runs in any modern browser without any knowledge of SQL. As discussed earlier, the user works on the data layer, which is separate from the process and business-presentation layers for management-and-scalability purposes.

NX provides readymade pick-lists, such as lists of countries, currencies, languages, etc. AgilePoint NX provides a library of entities and pick lists to save users time when modeling applications. Users can model real world scenarios by creating parent/child relationships between entities.

### E. Page Builder

Many teams and departments need to rapidly build custom web pages for day-to-day business operations. NX allows non-programmers to create customized user interfaces consisting of rich portlets, forms, and widgets and to integrate them with their existing business apps and business data. These web pages are built on open technologies and can be accessed from any business application platform.

## 6. Mobile App Accelerator (MAA)

Enterprise adoption of mobile technologies will continue to grow. MAA enables the generation of native iOS apps, Windows apps, and Android apps from existing NX apps in minutes, without writing code. These mobile apps can be customized with company branding elements, such as logos, color schemes, and images. MAA allows pre-configured apps with organization-wide settings. AgilePoint NX apps can be shared with users over the air (OTA) via secure deployment through Mobile Device Management (MDM) using Microsoft InTune.

## Summary

The digital storm shows no signs of letting up. In fact, many market analysts are predicting an exponential increase in the rate of change over the coming years. Organizations, faced with this reality must hedge against potentially catastrophic market events by adopting a paradigm for custom software development that will enable them to not only transform while the storm rages, but to sustain transformation with minimal effort. AgilePoint NX may be the only existing platform that can facilitate this strategy.

# About AgilePoint

In 2009, Ascentn Inc, founded in 2003, rebranded itself to AgilePoint Inc after its flagship product AgilePoint, a leading BPM solution at that point of time. Today AgilePoint is a global player that provides future proof Digital Transformation platform to Fortune 2000 companies, Small and Medium enterprises and Governments. AgilePoint NX platform, powers core and mission critical business application on a large scale in several large firms across the world.

This document is the result of research performed by AgilePoint's market research team and its published content is the best available analysis at the time of research.

**1,500+**
INSTALLATIONS WORLDSIDE

**28**
NATIONS

**75+**
WORLDWIDE PARTNERS

**~800**
MILLION PROCESS RUN IN 2017

## AgilePoint Offices Worldwide


**AgilePoint**

1916B Old Middlefield Way
Mountain View, CA 94043, USA
For more information, email us at info@agilepoint.com

Connect with us