Application Development Guide

AgilePoint BPMS v5.0 SP2

Document Revision r5.1.4

November 2011

Contents

Preface

Disclaimer of Warranty

AgilePoint, Inc. makes no representations or warranties, either express or implied, by or with respect to anything in this document, and shall not be liable for any implied warranties of merchantability or fitness for a particular purpose or for any indirect, special or consequential damages.

Copyright

Copyright © 2011 AgilePoint, Inc. All rights reserved.

Trademarks

AgilePoint, Inc. and AgilePoint's products are trademarks of AgilePoint Inc. References to other companies and their products use trademarks owned by the respective companies and are for reference purpose only.

Government Rights Legend

Use, duplication or disclosure by the U.S. Government is subject to restrictions set forth in the applicable license agreement and as provided in DFARS 227.7202-1(a) and 227.7202-3(a) (1995), DFARS 252.227-7013(c)(1)(ii) (Oct 1988), FAR 12.212(a) (1995), FAR 52.227-19, or FAR 52.227-14, as applicable.

Virus-free software policy

AgilePoint recognizes that viruses are a significant security consideration for our customers. To date, we have had no report of AgilePoint BPMS carries any virus. AgilePoint takes the following measures to ensure our software is free of viruses upon delivery:

- AgilePoint is built on top of Microsoft .NET framework. The pre-compiled executable is a.NET Common Language Runtime (CLR) application, not a native machine binary. As far as is known at this time, there are no viruses that infect .NET CLR executables.
- The virtual environment for the product packaging process in is fully isolated and protected, and anti-virus software is installed and running during packaging.
- The deliverable package is scanned by anti-virus software before upload to our customer download site.

Document Revision Numbers

AgilePoint documentation uses the revision number format **r*X*.*Y*.*Z***. The letters and numbers in this revision number can be interpreted as follows:

- **r** - Indicates "revision." This helps to differentiate the document *version* numbers, which start with **v**.
- ***X*** - The major version number for AgilePoint BPMS to which this document refers. For example, AgilePoint releases 5.0, 5.0 SP1, and 5.5 would all have an ***X*** value of **5**.
- ***Y*** - The major document revision number. This number typically changes only when either there is a new AgilePoint release, or there are major changes to the document.
- ***Z*** - The minor document revision number. This number is incremented each time the document is republished.

AgilePoint Documentation in PDF and HTML

AgilePoint documentation is provided in both print-friendly (PDF) and web-based (HTML) formats.

Advantages of HTML Documentation

- HTML is the **primary delivery format** for AgilePoint documentation.
- Unified, global **search** across all documentation. PDF documents allow you to search only within the context of a given PDF file.
- **All hyperlinks supported**. Links in PDFs are only supported in certain contexts.
- "One-stop shopping" for all information related to AgilePoint BPMS.
- The HTML documentation is updated more frequently than the PDF documentation. Web-based documentation is updated periodically between AgilePoint releases to address errors and omissions, but the PDF documentation is updated only at the time of a software release.

Advantages of PDF Documentation

PDFs can be more easily **printed**, **archived**, and **transferred** (such as by FTP or email) than HTML documentation.

For more information, see Downloading Files and Sharing Links from the Documentation Library on the AgilePoint Support Portal.

Contacting AgilePoint Sales

AgilePoint is a leading Business Process Management System (BPMS) provider created by a team of driven people who strive to incorporate the principles of relentless innovation for the benefit of our customers. Our mission is to help companies of any size attain and sustain operational success through process excellence.

Headquarters: AgilePoint Corporation 1916C Old Middlefield Way Mountain View, CA 94043, USA

Tel: (650) 968 - 6789

Fax: (650) 968 - 6785

Email: info@agilepoint.com

Web site: www.agilepoint.com

International: For AgilePoint EMEA and AgilePoint Asia Pacific, please call the AgilePoint Corporate Office for contact information.

Contacting Customer Support

To contact AgilePoint Support, please submit a ticket on the AgilePoint Support Portal: http://support.agilepoint.com/SupportPortal/

If you do not have a Support Portal account, you can send an email to request one: support@agilepoint.com

AgilePoint Project Initial Assessment

The following questions are keys when approaching an AgilePoint development project:

- How do I know if this process can be automated?
- Should this process be automated?
- How do I gather information about requirements?
- From whom do I gather requirements?
- What kind of Development team should I create?
- If I don't have enough resources, how do I find them? From AgilePoint or AgilePoint partners?
- What kind of application or process is a good fit for AgilePoint?

Determining Whether to Use AgilePoint

When determining what projects are a good fit for AgilePoint, it is important to remember that AgilePoint is a process-driven application.

The AgilePoint engine drives the process by acting as an XML interpreter. AgilePoint does not need to generate or compile code, so it has a great deal of flexibility and adaptation, even at runtime.

In general, AgilePoint works best for applications that are process-driven, and the processes focus on:

- Human to human
- Human to system
- System to system

Projects that are a Good Fit for AgilePoint

The following types of processes are a good fit for AgilePoint.

Human-Centric Processes

Human-centric processes focus on human review, approval, etc. AgilePoint automation is especially effective when these processes require collaboration:

- Across departments
- Across geography
- Across multiple companies

Many Process Instances

Processes that are initiated frequently, transfer across departments, and need to be tracked are good fits for AgilePoint. AgilePoint can provide access to information regarding the time consumed for a task, to whom it is assigned, and the current status.

Requirements that Change Frequently

AgilePoint is a good fit for processes where requirements change frequently due to changes in legislation or company policy, and the process must change to adapt in days or weeks, rather than months or years.

Ambiguous Requirements

When you design the process, some elements may be unknown, but you know that the details will need to be added later. AgilePoint enables you to skip steps, roll back, or migrate to a new process model as requirements become clearer.

Complicated Processes

Similar to the case where requirements are ambiguous, complex processes may have many sub-branches where the details are not easy to determine up front. AgilePoint enables you to develop these sub-branches independently and improve them over time without affecting the main process. The sub-branches can be changed during runtime without interrupting the process flow.

Processes that Initiate Automatically

AgilePoint is a good fit if you want to trigger processes without human intervention – for example once a day, or triggered by an event, such as an email or a database change.

SOA is Desired

If SOA-based processes are desired in your application, AgilePoint is a good fit. AgilePoint processes invoke web services, instead of applications. SOA-based processes are excellent for human to system or system to system processes.

Projects that Require Analysis

The following types of project require some analysis to determine whether they are good fits for AgilePoint. In particular, look closely at processes that occur infrequently. Ask:

- How long is the development time for the process?
- How complex is the process?

Projects that are Not a Good Fit for AgilePoint

Processes that are heavily driven by system-to-system activities may not be a good fit for AgilePoint. These processes include:

- A high number of system tasks
- A high number of consecutive system processes

For example, a process that must migrate or transfer data from one system to another1000 per second is too high volume for AgilePoint. AgilePoint uses an interpreter model, not a compiled code model. This would be a better task for compiled code.

AgilePoint BPMS Application Development Planning

Application development using AgilePoint differs from other types application development because AgilePoint development focuses on processes. You use the AgilePoint platform to develop process-driven applications.

It is important to standardize processes and make them manageable because:

- Processes are business assets
- Processes are important to productivity
- Processes improvements can drive business competition

Whether you're working with a paper or electronic process now, AgilePoint is a platform to convert or automate the traditional business process to an electronic process.

AgilePoint is process-driven and SOA-aligned integration platform.

Bringing a next-generation platform. Makes the developer much more efficient, easier integrations, faster change.

Application Development Types

We can divide application development into two types:

- Non-Process-Related
- Process-Related

Process-related applications can be created much more efficiently using AgilePoint.

Non-Process-Related

In a non-process-related application, the focus is on viewing or updating data: searching, reporting, modifying data, etc. These applications:

- Do not interact with a process
- Do have permission controls
- Don't have time constraints
- Typically provide a form-based UI for the user to input data
- Include some logic in the UI, but it is in the form program, and not controlled by a process. In other words, the coding, or logic, or rule is embedded in the forms.

.NET technologies work well for this type of application. You can use a .NET development resource. AgilePoint is not required.

Process-Related

In a process-related application, the focus is on following, updating, or integrating with a process. These applications:

- Do interact with a process
- Do have permission controls
- Do have time constraints

The AgilePoint framework is well suited for this kind of functionality. It can be developed very fast with the AgilePoint framework. However, it would take a relatively longer period of time to develop using .NET.

AgilePoint Application Development Approaches

There are two basic ways to develop applications using AgilePoint:

- AgilePoint BPMS as a development platform
- AgilePoint BPMS is an integration platform

AgilePoint BPMS as a Development Platform

In this approach, AgilePoint BPMS is used as a base platform to develop all functionality for the application.

When this Approach is Best

- The developer can take time to thoroughly understand AgilePoint BPMS.
- A high percentage of the application is process-oriented (although some aspects may not be related to processes).
- The process must be human-centric, where the main purpose is more efficient collaboration between department or roles.

AgilePoint BPMS is an Integration Platform

In this approach, AgilePoint BPMS is used as a platform to integrate disparate systems, but the application logic is developed using another technology.

When this Approach is Best

- Process-related functions are fewer than non-process-related functions.
- The application already exists before they AgilePoint is adopted.
- The process must be human-centric, where the main purpose is more efficient collaboration between department or roles.

Guidelines for this Approach

- Separate the process and the application. Use AgilePoint to automate the process. Let AgilePoint handle the process and rules that must be changed frequently.
- Use AgilePoint as integration platform, such as ERP, CRM, rather than directly connecting from your application to system. Use the process to perform the integration, not in the application. This is another level of abstraction in the application that leverages AgilePoint's SOA capability.

Identifying Your First AgilePoint Process

What is a Business Process?

A business process is a set of actions that are placed in a particular order to achieve an objective. Actions include both human activities and automated actions that implement routing rules.

An AgilePoint BPMS process defines a set of activities using Visio-based AgilePoint Envision shapes.

In addition to using the out-of-the-box AgilePoint components, AgilePoint BPMS also provides a very effective framework and process-oriented, reusable component development framework, which you can use to build your own components. By developing your own, you can embed business logic to meet your particular requirement.

These components can not only be reused in different processes, the component is configurable for different process analysts and business users. The framework provides a kind of vocabulary that both the developers and analysts can understand. This is called "Business Process WYSIWYG."

Principles for Identifying Your First AgilePoint Process

Focus on People

Human end users see the greatest benefit from automated processes. Processes that make participants' jobs easier will not only improve efficiency, it will also help adoption of the AgilePoint system as more processes come online.

When you are deciding upon your first process to take on, try to focus on processes where users must touch the system frequently and/or feel a great deal of pain with the current process, and then improve the process for them.

Focus on Subprocesses

Often organizations want to start with a large or complex process first to get an immediate high impact. However, AgilePoint recommends breaking large processes into smaller subprocesses.

For example, when sending an alerts for events such as approvals or system errors, the alert subprocess can always be the same, regardless of the parent event that triggers the alert.

- Subprocesses are relatively easy to manage because they are smaller
- Subprocesses are relatively easy to define because they often occur within a single department or group.
- Subprocesses yield a high impact because they can often be reused across processes.
- Modularized subprocesses can use a standardized interface, so changes to the subprocesses do not impact the parent processes. (This is similar to the advantages of the SOA software model.)

Identifying Good Processes for Automation

When getting started with AgilePoint, you must identify the first process you want to enable with AgilePoint BPMS. The following criteria can help you to identify products that are good candidates for AgilePoint processes:

- **The process has many of participants.**

  Processes that have many participants throughout the process lifecycle require many manual activities. These are often good process to automate because the greatest savings process automation has the greatest impact when it affects human activities.

- **The process has a large number of instances, and the instances need to be tracked.**

  Processes that are initiated by many people and many people need to track the process as it progresses are good candidates for automation. Good automation can offer a high degree of visibility to the process.

- **The process crosses departments or geography, and the traditional process method is paper-based.**

  Processes that require the transmission of paper over distances also require time and money to transfer the paper, such as mail or physical transfer, or electronic transfer using fax machines or scanners. AgilePoint can automate these kinds of processes to improve productivity and reduce costs.

- **The process has a long running time.**

  If a process runs for a long time -- for example, several weeks, rather than a few days -- it is difficult to track the process status. It is difficult for users to know where the process is at a particular time and what is the next step.

- **The process has tasks assigned for a short period of time.**

  When tasks are assigned for a short period of time, it is difficult to remind them that it is almost due within a process that is not automated. AgilePoint enables immediate, automated reminders (such as emails), so that individual participants can focus on their tasks. It is not necessary for participants to know about other parts of the process.

Designing an AgilePoint Process

When designing an AgilePoint process, follow these steps:

1. Design the Data Model
2. Design the Forms
3. Design the Workflow

Design the Data Model

The first step when designing an AgilePoint BPMS application is designing the data model. The data model is the key component to connect system to system, system to human, and human to human in the process.

The data model is defined by the XML schema, and it determines the metadata for the process. The data model is reflected in the form, in the rules, and in the process integration.

AgilePoint's XML-based metadata model is an excellent starting point because it is a strong data model with few limitations.

Design the Forms

Once you have designed the data model, you must design the forms.

The form is the bridge between humans and systems. The main purpose of the form is to bring the process data to the process participant.

AgilePoint supports multiple form technologies, including AgileForms, SharePoint ListForm, ASP.NET, WPF, WinForm, and InfoPath. AgilePoint can also deliver information using email notifications and approvals.

Design the Workflow

The process workflow design is the most important step. It's the bridge from human to human, system to system, and human to system. It delivers the right information to the right people at the right time according to the process design.

The process design is the collaboration system for roles and groups.

Designing a Process Branch

Remember that when you design a process, you need to focus on designing one subprocess, or process branch, at a time.

Defining a process branch is the process of defining the relationships between activities, which are typically based on the relationships between departments or roles. For example, in an approval process branch, manager approval typically comes before CEO approval.

Note that it is important to design the relationships within the branch first, before focusing on the individual activities.

Design the Path for the Success Case

When you are designing a process branch, focus first on the path for the success case. This is the case in which everything goes forward successfully with no interruptions.

The success case follows the "80 percent rule:" this is what happens 80 percent of the time. The other 20 percent are exception cases.

To design the success case, you need:

- Good requirements gathering and analysis
- Communication with the end users of the process

The end users for the success case are typically the primary stakeholders for the process branch from whom you will gather the requirements.

Design Exception Cases

Besides the success branch, there are additional sub-branches, which are typically used for exception cases.

Exception cases often occur because of a mistake made by a participant -- for example, if an approval is rejected because a required document is missing.

In the enterprise, it's often difficult to define the sub-branch and exception branch precisely. People understand the success branch better than the exception branches because it is used more frequently. Exceptions often occur infrequently and are handled in an inconsistent or ad hoc fashion.

During your process design, if you can capture 70-80 percent of the exception branches, it is a reasonable goal for the first iteration of the process implementation. In the future, you can find ways to refine and improve your process. The AgilePoint metadata platform makes these improvements simple and efficient.

Define Participants

In a human-centric business process, defining the participants is a key element.

It is important to avoid a single point of failure by hard-coding a user name. If a user name is hard coded and the individual is away, changes roles, or leaves the company, the process would fail.

There are two approaches to avoiding single points of failure with user names:

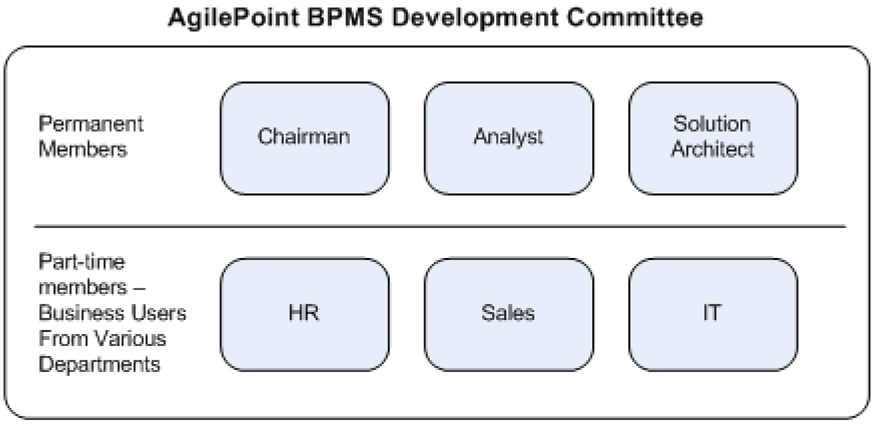
- Define roles or groups to associate as participants. You can use roles or groups in AgilePoint or in Active Directory.
- Define participants using process metadata, such as form a form or web service. This enables you to select a participant based on the process instance.

Process Development Committee

AgilePoint recommends forming a committee for the process design and implementation. The members of the committee consists of 3 people:

- AgilePoint systems and applications analyst.
- Solution architect.
- Business users and/or business analysts. A business analyst often serves as the committee chair.

Any of these roles can be filled by resources can be dedicated to the project, or they can be shared or outsourced. Their commitment to the project may be full time or part time, depending on the project size.


AgilePoint BPMS Development Committee
Permanent Members
Chairman
Analyst
Solution Architect
Part-time members – Business Users From Various Departments
HR
Sales
IT

Development Committee Responsibilities

The AgilePoint application development committee completes the following tasks.

Categorize Processes

Define categories for processes, such as "sales" or "HR", and determine the scope of each project. In this way the committee creates a kind of taxonomy for processes.

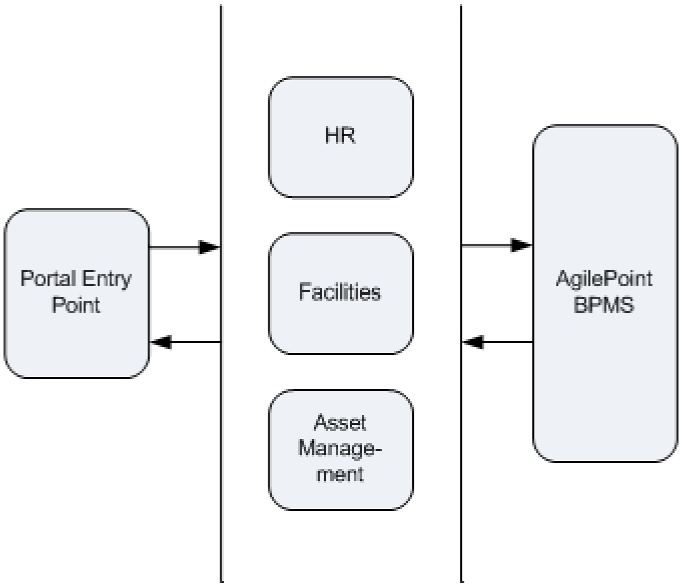
Determine the Number of Applications

The solution architect determines the number of applications to create -- whether there will be only one application, or whether particular groups of processes will get different applications.

AgilePoint recommends having a portal-type user interface where users access the applications they require from a single web page. Applications are developed independently to support various types of processes, such as sales, customer support, or HR. AgilePoint BPMS operates as the engine serving all of these applications

AgilePoint BPMS can serve any number of required applications, so the decision is unimportant from the AgilePoint perspective. Likewise, any number of applications can be served from the same user-facing portal. So the decision is primarily a solution of architecture that depends upon your organization's requirements.

The following diagram shows a sample architecture with a middle layer that includes multiple applications, but with a single, shared entry point and AgilePoint BPMS.



Gather Process Requirements

The committee works with business users to gather and analyze process requirements, design the process, and document the process.

A best practice is to keep the process documentation in a SharePoint library.

Prioritize Processes for Automation

Assess which process is best to be automated using AgilePoint BPMS. For more information, see Principles for Identifying Your First AgilePoint Process on the AgilePoint Support Portal.

Identify Shared Subprocesses

Determine subprocesses that can be modularized and shared. For more information, see Principles for Identifying Your First AgilePoint Process on the AgilePoint Support Portal.

Develop the Process Application

If the company has a development resource, the developer can create the application. Otherwise, this work can be outsourced to an AgilePoint partner.

Application Development Team Responsibilities

AgilePoint recommends you divide application development between two teams.

Component Development

The component development team develops modular AgilePoint components such as AgileParts, AgileConnectors, and AgileExtenders.

The component development resource requires strong understanding of a model-driven architecture in order to make components better at abstraction, generalization, and reusability.

Application Development

The application developers must have a concept of the process data, and how to separate it from the business logic.

Application Development Team Organization

You can organize the application development team in two ways, depending upon your organization's requirements.

Larger Team, Shorter Time Span

If you want to implement a large number of processes in a short period of time -- say, 100 processes in 6 months -- you would need a large number of developers for that time, and then a few people performing maintenance once the development was complete.

There are two important considerations for this type of model:

- All of the resources involved must be familiar with AgilePoint development methodologies. You must train all of them before the project starts.
- Because of the large team size and short time span, effective project management is required.

Smaller Team, Longer Time Span

If you want to implement a smaller number of processes over a longer period of time -- say, 3-10 processes per month -- you would need a relatively small number of developers on an ongoing basis.

In this type of model, the initial costs are lower, but it takes longer -- typically 2-3 years for a large enterprise.

Case Studies

Case Study 1

A company in the financial industry bought the AgilePoint BPMS Suite in 2005 to facilitate an IT development team supporting the company's growth for the next 5-10 years.

In the first two months, the company:

- Established a committee to manage the development.
- Enlisted business users (mostly financial industry professionals) and installed AgilePoint Envision on their machines.
- Trained the appropriate staff.

After the initial setup and training, the business users designed the draft of the process drafts. Then the committee worked with them to prioritize the processes and come up with a plan for implementation.

The first application they created was a financial customer service application to increase the quality of service and productivity. They had 5 developers coding ASP.NET. They also used an on-site AgilePoint consulting services for this first project. The system was online within 3 months.

Once the first application was online, they separated the development responsibility:

- 2 senior developers managing AgilePoint components, such as AgileParts, AgileConnectors, AgileWorks.
- 3 developers responsible for application development.

By 2007 (two years later), they had a total of 6 AgilePoint applications running, and the same IT team was still able to support the company's business.

Case Study 2

A company in the banking industry implemented AgilePoint BPMS to support their credit card customer service system. Their objective was to increase the number of requests fielded by of their support staff by 3-5 times, without increasing staff.

Because the company wanted the complex system completed in a short period of time, they opted to outsource the development to an AgilePoint partner. Within 17 months completed the application form analysis to production, including more than 70 processes and 100 ASP.NET forms.

This became the company's first ASP.NET application, and the first SOA-aligned application. It integrates with Oracle, Unix LDAP servers, and an ERP system. The application also adds its own organizing structure to handle parallel approvals that follow the organization's unique structure.