
Understanding K2 and Windows Workflow Foundation

June 2011

Table of Contents

Intended Audience.....	3
1. Summary	3
2. What is K2	4
3. What is Windows Workflow Foundation	7
4. Where is Windows Workflow Foundation implemented	9
5. Positioning K2 and Windows Workflow Foundation	10
5.1. Understanding the detail of WF.....	10
5.2. Understanding the detail of the K2 Platform.....	10
6. How does K2 leverage WF	14
7. ROI Models.....	18
7.1. Scenario 1 – Small Solution.....	18
7.2. Scenario 2 – Medium Solution	18
7.3. Scenario 3 – Large Solution.....	19
7.4. Descriptions	19
7.4.1. Build	19
7.4.2. Test.....	19
7.4.3. Deployment.....	19
7.4.4. Training	19
7.4.5. Maintenance	20
7.5. Small Project - ROI	20
7.6. Medium Project - ROI.....	21
7.7. Large Project - ROI	22
8. Summary	25

Intended Audience

This document is designed for solution architects, project leaders, system integrators, and business managers who have a requirement to understand the positioning of a fully featured BPM product such as K2 against the native developer-focused capabilities provided in the .NET framework by Windows Workflow Foundation (WF). In many instances customers request this reference material in order to help them understand and position these technologies to the business.

1. Summary

K2 is a fully featured process automation product comprising a scalable server architecture, multiple graphical process designers, routing and tasking, integrated security, data and line-of-business integration, process administration and management, process visualization, escalation management, KPIs and reporting.

Windows Workflow foundation (WF) is a core component of the .NET 4.0 framework. WF is a developer-based technology aimed at applications that contain “workflow-type patterns”. The entry point for anything Workflow Foundation-based is Visual Studio and is inherently the space of professional developers. K2 makes extensive use of Windows Workflow Foundation as foundational building blocks at the execution level of its workflows.

Clearly building workflow applications on native WF is possible. However, when considering business solutions that encompass the concepts of process or workflow projects, it is essential to understand that products like K2 are designed to deliver completed low / zero code business solutions in weeks. K2 has 10 years of workflow R&D, works closely with the Microsoft product teams including the .NET WF team and continues to build on the core architecture that Microsoft continues to evolve. The end result of this is a platform that is targeted at solving business issues, as opposed to a technically focused experience available with WF.

Purpose-built products such as K2 have moved significantly beyond the “proving stage”. Building workflow applications today in code would be similar to building an email server using the System.Net.Mail namespace also available in the .NET framework. A custom built email server is indeed possible and chances are that the end product may even do specific things for your organization that are not available in Exchange. However, economics dictates that organizations are no longer in the business of coding this kind of functionality. This same set of economics applies to business process management and workflow.

2. What is K2

The best businesses in the world run on process. Companies that understand their processes well and are able to implement them using automation, metrics and visibility execute better than their competitors. K2 provides a technology platform that enables customers to define and **run these processes** as process-enabled applications.

In small, medium and large businesses there are literally hundreds, if not thousands, of processes that are candidates for automation. K2 customers are using the K2 platform to automate these processes; examples include employee expense claims, insurance underwriting and claims, invoice approval, loan creation, customer on-boarding, scanning and capture processes, supply chain, employee on-boarding and so on.

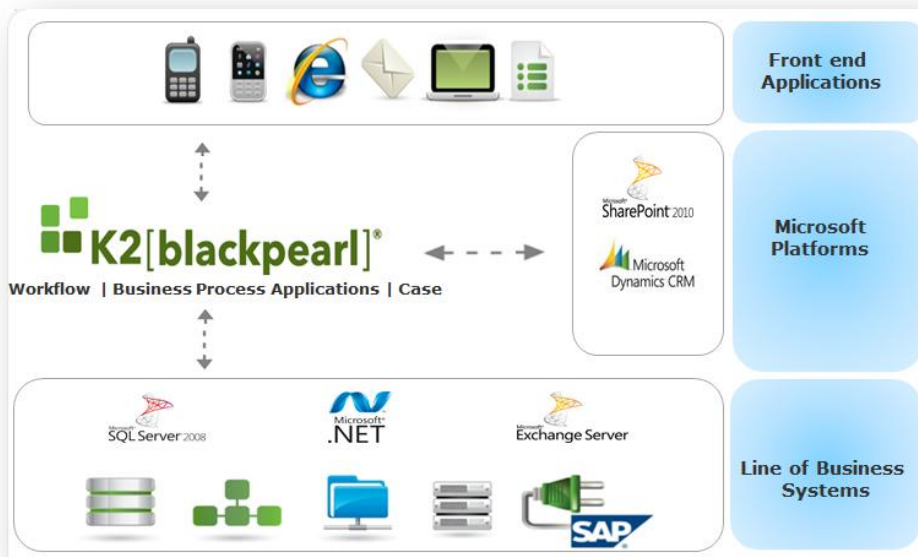
Typically this technology space is referred to as BPM (business process management) and K2 is rated as a visionary by the world's leading analysts in this space. The K2 platform is aimed at **pervasive execution** of business processes under the K2 server.

Delivering automated process applications requires a mature set of capabilities as illustrated in the following table:

K2 capability	Addresses
Visual designers	Allows users to visually design process based applications with "block and line" visual capabilities
Runtime server	A high-performance process server capable of executing thousands of concurrent processes.
User interface capabilities	Ability to route and manage work to users on a variety of user interfaces including Internet browser-based applications, SharePoint, mobile client, InfoPath and thick clients
Integration capabilities	Ability to seamlessly integrate with SharePoint, SQL databases, Web Services, .NET APIs and line-of-business systems (e.g. SAP)
Security	Native Active Directory-integrated security. Can be extended to support other authentication providers as well as run in a mixed-mode authentication environment.
Routing management	Ability to dynamically route tasks in parallel, covering common workflow patterns, based on

	business rules to individuals, groups and roles
Process management	All processes are stored and available for compliance-based reporting
Reporting	A full suite of reports allowing users to visualize and analyze process data and data linked to their processes
Administration	Version management, error management, live instance management, delegation, out of office, security
Package and deploy	Ability to wrap process assets and deploy a complete process-based solution across environments
SLA, KPI and escalations	Built-in mechanisms for servicing and managing service level agreements, KPIs and escalations.
External integrations with K2	K2 provides many methods for other systems to interact with K2. These access paths include a rich .NET API, WCF Services and REST Services.

The K2 features come together as a suite of capabilities that allow a variety of end-users (power users, analysts and developers) to build both simple and complex business processes that are immediately available as consumable and running business processes. K2 leverages the capabilities of the Microsoft platform extensively (including .NET Framework 4.0 and Workflow Foundation) to deliver this capability. At a high level K2 can be positioned as follows:



Conceptually, K2 allows the logical process diagram to be defined and modelled on the K2 design canvases. Once the initial design concept is complete, the visual model is refined with the required assets and logic to enable the process to “run”. This includes definition and integration of the required user interface, integration with back-end data sources, routing and security and associated business logic to control the routing. Once the detailed modelling is complete, the K2 process is deployed to the K2 server which then delivers the entire process as a fully functional application to the business. In many cases this can be achieved with low / zero code.

3. What is Windows Workflow Foundation

Windows Workflow Foundation (WF) is part of the .NET framework for .NET developers. Introduced with version 3.0 of the .NET platform, Microsoft combines WF (Windows Workflow Foundation), WCF (Windows Communication Foundation) and WPF (Windows Presentation Foundation) along with the core .NET classes to bring developer productivity for common programming tasks. At an abstracted layer, programmers apply these capabilities to solve common problems associated with typical complexities around processing patterns, networking/interconnectivity and user interface. The following matrix depicts this relationship:

.NET Framework 4.0	Addresses
Workflow Foundation	Common process patterns, wait, long running, decision based..
Windows Communications Foundation	HTTP, TCP/IP, Async, REST, SOA, WS-*
Windows Presentation Foundation	Vectorized graphics, animation, gradients transparency...
.NET Classes	System IO, Data, Strings ...

Windows Workflow Foundation (WF) is not a product, server or an application. Instead, it is a framework used to build an application or a server with workflow capabilities. WF is an in-process engine that runs inside a host application. The host process is responsible for providing a set of services to Windows Workflow Foundation. WF is Microsoft's vision for embedding workflow capabilities within applications. Microsoft's WF technology is intended to provide a standardized methodology for the low-level code of workflow systems. It allows for the abstraction of application logic into a visual representation — making the logic easier to follow and understand as compared to legacy workflow approaches, which often resulted in monolithic code. The platform includes a set of development tools for designing and implementing workflow activities, a programming model for controlling and communicating with workflows and a set of workflow runtime services for persistence, tracking, transaction management.

In building the Windows Workflow Foundation, Microsoft abstracted and consolidated the following core workflow concepts that can be used across multiple types of applications:

- **Design and visualization.** In WF workflows are trees of actions (individual work items called activities). In this way the logical path of any application is shown – with representations for

both machine and human actions. The workflow tree can be manipulated directly or as an object model in code

- **Hosting.** The WF runtime is lightweight and is typically hosted in a client context. Alternatively, it is possible for custom WF hosts to be created. SharePoint Server is capable of hosting the WF runtime for example. Interaction with the host is facilitated through provider interfaces for services such as threading, transactions, persistence, and communications.
- **Semantics.** WF supports three main styles of workflow out of the box: sequential, state machine, and rules-driven.

In its own right (without K2 or another comprehensive host):

- WF does not understand things like Active Directory or any other authentication/security provider
- WF has no notion of a user called Bob or how to find Bob's manager
- WF has no notion of telling the system that Bob has a task that only he can perform.
- WF provides no functionality for real-world human task management:
 - no escalations
 - no centralized task-lists
 - limited reporting and auditing
 - limited error handling
 - no human process KPIs
- No workflow versioning
- Limited process management

The above elements are imperative for a robust, human-oriented process management system. While it is true that the above features could be built out via custom coding, it is important to factor this level of effort when evaluating a workflow project.

4. Where is Windows Workflow Foundation implemented

Workflow Foundation (WF) as an underlying technology is hosted within many products:

1. Custom written applications - Native Workflow Foundation (WF) code will require a host or more often than not a server host to run within. In the past, developers were required to code the host but going forward AppFabric will provide a framework host for hosting native Workflow Foundation schedules.
2. Microsoft SharePoint provides a host for Workflow Foundation and therefore developers can develop and extend WF-based workflow within the SharePoint context.
3. CRM also hosts WF-based workflows and again developers can extend WF activities in the context of the CRM host.
4. K2 hosts WF schedules which form the execution engine of many of the K2 capabilities. Anyone who designs a workflow in K2 leverages WF. K2 provides an abstraction layer over WF (this will be covered in detail later in this document). Additionally, developers can extend and create new WF workflows within the K2 server context if desired.

It is important to understand that both the kind of activity that you undertake and the design patterns you implement will be significantly different depending on the kind of WF coding you are undertaking. Extending a SharePoint-hosted workflow in WF is a considerably different activity to writing a natively-hosted WF application in AppFabric.

5. Positioning K2 and Windows Workflow Foundation

Now that we have a high-level picture of both K2 and Windows Workflow Foundation (WF), let's drill down into specifics of the capabilities and positioning of K2 and WF.

5.1. Understanding the detail of WF

WF is a software developer technology

- It is a programming model, engine and tools for building workflow-enabled software on Windows
- It gives software developers a model-driven tool which is higher level but still integrated with .NET code
- It is extensible to enable a wide range of software development projects that require process execution or business logic execution via custom .NET code
- It is targeted at Independent Software Vendors (ISVs), both Microsoft products and external product developers (like K2), that wish to embed core WF capabilities and extend these further. Microsoft spends a lot of time planning the features for this audience.
- WF fundamentally does not have a server process that you can directly run in a datacenter as WF is simply a way to model code. To run WF an organization needs to build the server (called a "host") or integrate with IIS and the emerging technologies around IIS such as AppFabric.

It is beyond the scope of this document to head into code-level tutorials and examples. However, if interested, the below link provides insight into the basics of writing a very simple entry-level Workflow Foundation solution.

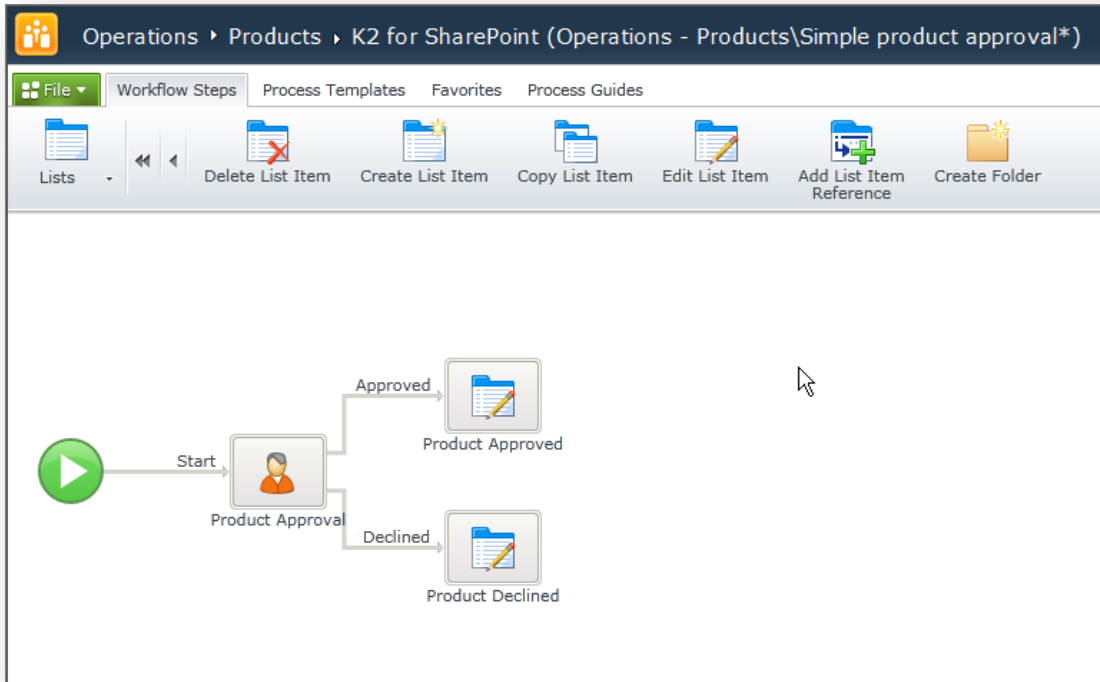
<http://msdn.microsoft.com/en-us/library/ee342461.aspx>

5.2. Understanding the detail of the K2 Platform

The basics of the K2 Software Platform

- In a process-centric solution the focus tends to narrow in on the "blocks and lines". This relates to the design canvas where a developer can fundamentally map the concepts of a flow and a decision. In reality this process mapping, while core to the application, represents a tiny percentage of the components and effort that ultimately make up a running business application based on this process.
- As with any application, process-centric applications need a user interface, integration with back-end systems, diverse workflow pattern support, integrated security, auditing, routing and dynamic capability to drive process logic at run time, user routing and tasking, role management, integration with back-end systems, deployment, management, out of office, escalation capabilities, reporting, archiving and so on.
- Each of the capabilities mentioned above cannot be classified into the "nice to have" category. These are essential capabilities that every customer will ultimately need in a running solution. Failure to account for these can limit real-world effectiveness of an application.
- K2 provides a product capability that automatically brings all of these capabilities and more to any workflow application deployed on the K2 platform.

- In the K2 world, process design is central, yet only one cog within many moving parts that make up a running process application.
- In the example below, we see a very simple entry level workflow for approving a product. Input into the workflow will be a form capture capability of product-related data. The workflow then routes the form to an individual who can approve or reject the product. Simple actions to store / update the product status follow as a result.

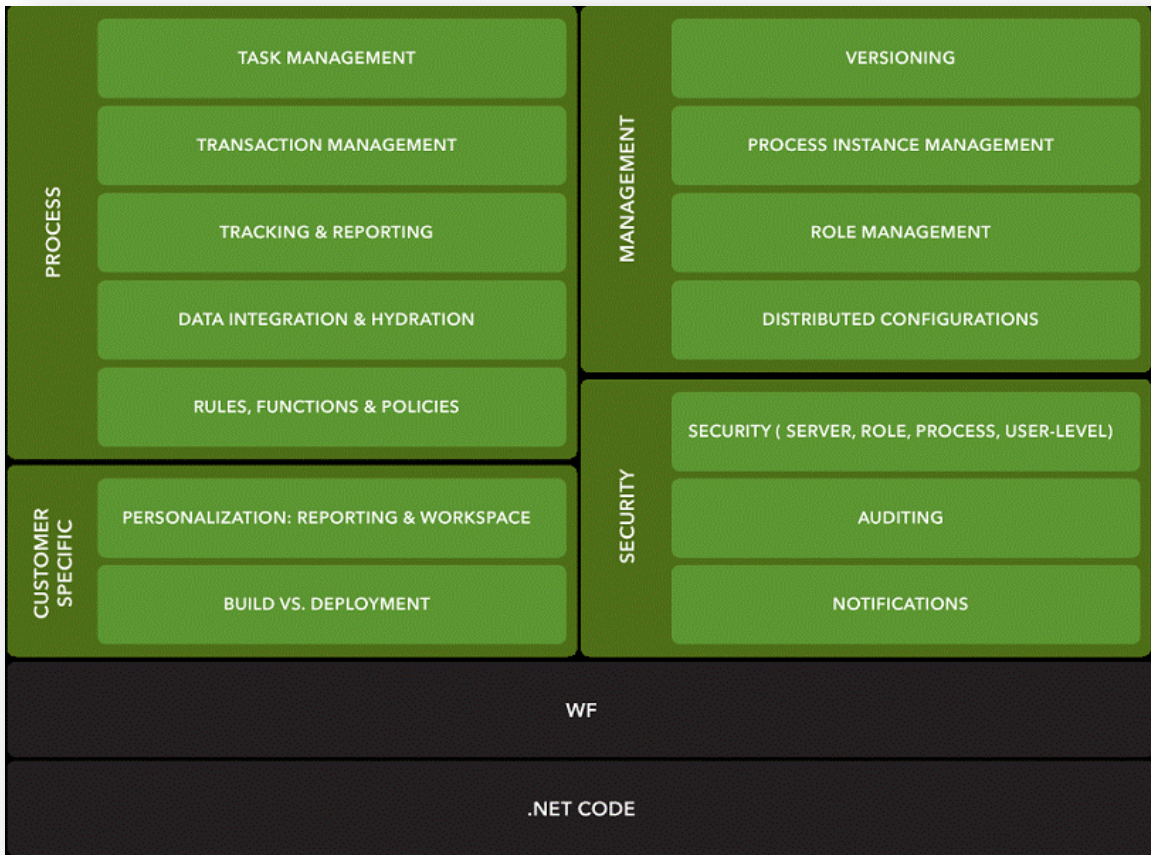


Because K2 is a platform, here are the capabilities that are **automatically provided out-of-box** as a result of completing this simple workflow “block and line” design.

End-user design canvas	Provided (NOTE: K2 ships with 3 workflow designers, each targeted at different audience skillsets and project complexity)
Routing engine	Provided and integrated in to AD, SharePoint and user-defined roles
Authentication	Provided – integrated with Active Directory
Form generation	The required forms to start and run this workflow including the approval forms are all generated based on the required data in the workflow
Approved / Declined logic	All logic to manage the approve / decline choices is automatically created and the workflow automatically routes to the designated activity
Approval actions	In this simple example the required actions to

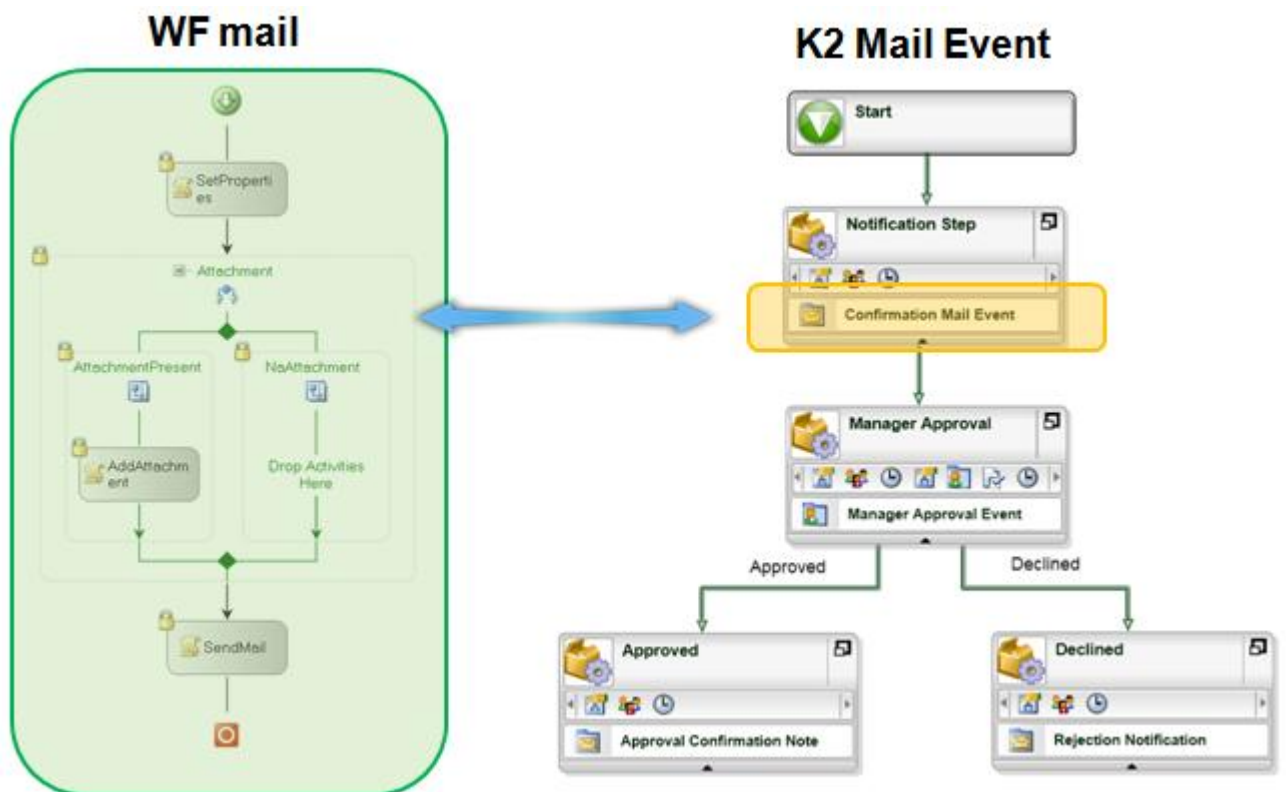
	update the SharePoint list are provided by K2. K2 ships with several out-of-box capabilities in this area. These can be as simple as updating a SharePoint list or as complex as updating SAP calling a custom BAPI.
Declined actions	See Approval Actions
Workflow Task list notification	All automatically provided and available in SharePoint as webparts, as a standalone application, via email or mobile devices including iPhone and Blackberry
Ability to delegate the approval	Provided to any authorized user
Ability to redirect the task	Provided to any authorized user
Management of the workflow instances	All provided
Version management of the workflow instances	Provided
Administration and tracking of all running workflows	Provided
Visual status checking of the running workflow	Provided
Graphical reports on workflow status	Provided
Detailed reporting and analytics on all aspects of the workflow and related data	Provided
Storage and management of workflow state	Provided
Auditing of workflow actions	Provided
Archiving and storage of all workflow instances	Provided
Health statistics and server management	Provided
Security model on who can start and view workflow processes	Provided
Lines of code required	0

The above simple scenario showcases that a workflow project is much more than simply orchestrating steps in a predesigned workflow. The following diagram gives an indication of the set of services that K2 provides:

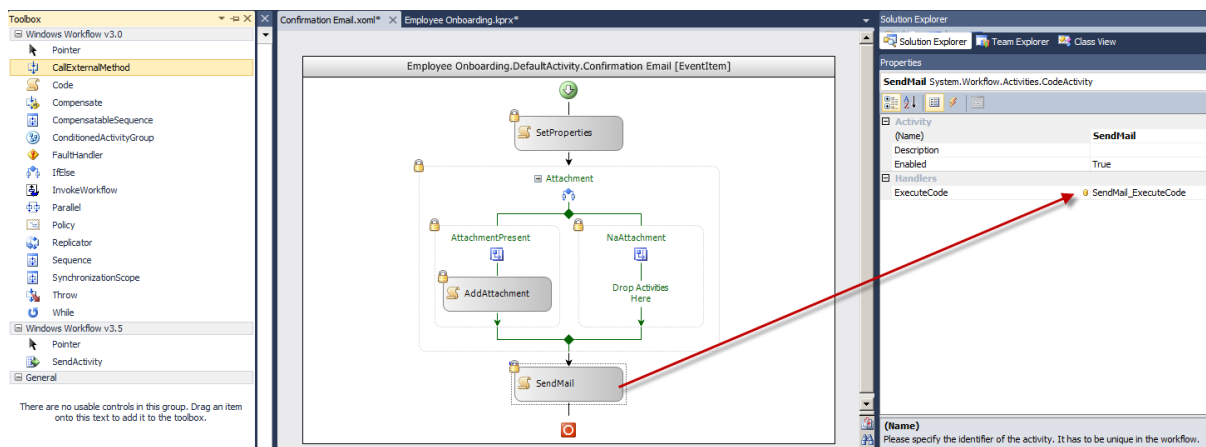


6. How does K2 leverage WF

K2 is built upon WF to leverage WF's strength of code modelling and execution. The K2 Mail event capability within K2 is a good example of how K2 utilizes WF as part of its runtime engine to achieve the send mail event. The design experience within K2 is simply to add the mail event to the process designer and configure the routing and messaging requirements for that mail. Behind the scenes K2 generates and configures a WF schedule to actually execute this capability at runtime. The following illustrates this relationship. Note that this shows a workflow modelled in K2 Designer for Visual Studio (which grants WF level access) but the same paradigm is leveraged throughout all K2 workflow designers even if the user does not have WF level access.



Let's review the design-time experience of the above mentioned email wizard in K2 tooling as compared to WF workflow modelling experience. When designing within a WF design canvas the components are very code context-focused, as the following screenshot shows:



Notice how the WF activities on the left fall within typical developer/coding paradigms such as “while”, “throw”, “if/else”. Within the workflow canvas, in the middle pane, the actual workflow has very little correlation to the business process on hand because it is dealing with the specific coding logic to handle this flow. Beyond that, to alter an activity it is not uncommon to have to invoke a method in the underlying C# file such as below to execute logic:

```

private void SendMail_ExecuteCode(object sender, EventArgs e)
{
    SmtpClient client = new SmtpClient();

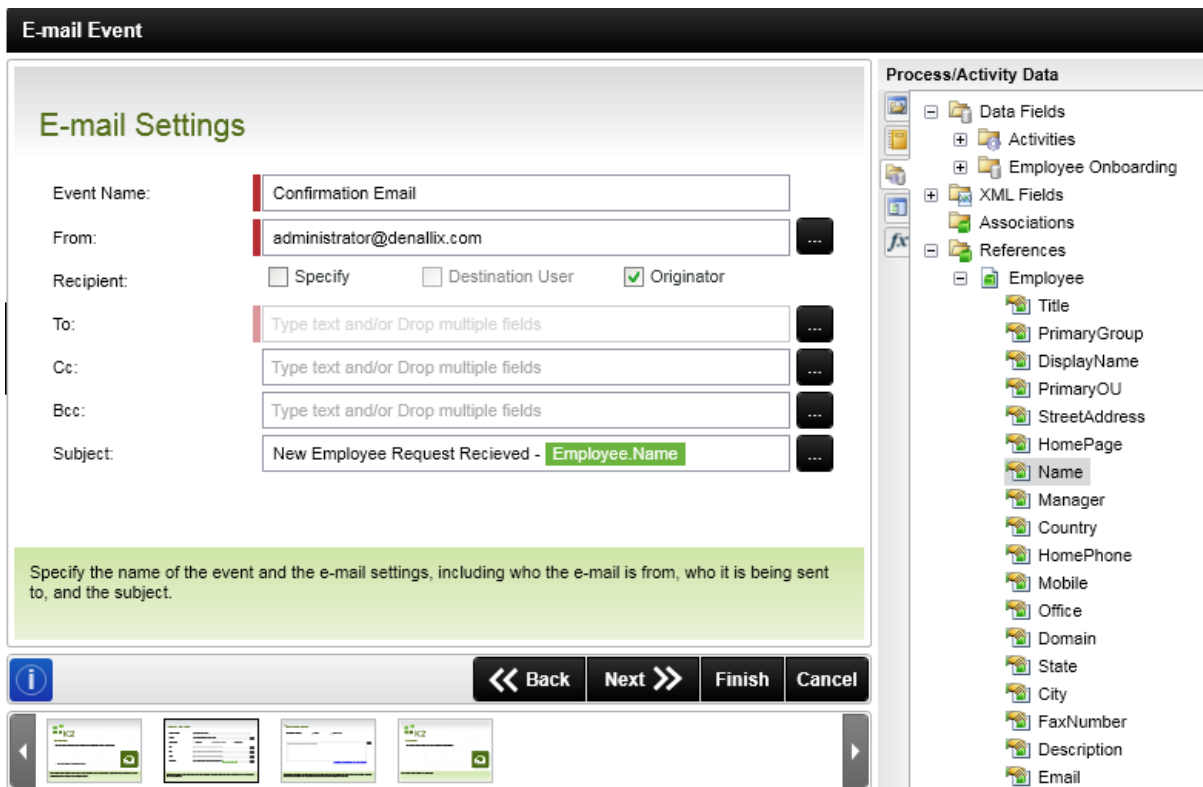
    if (K2.Configuration.SMTPServer.IndexOf(":") == -1)
    {
        client.Host = K2.Configuration.SMTPServer;
    }
    else
    {
        string[] connection = K2.Configuration.SMTPServer.Split(

        Int32 I = 0;

        if (string.IsNullOrEmpty(connection[0]) || string.IsNul
        {
            throw new Exception("Invalid SMTP Server / Port argu
        }
    }
}

```

In contrast to the above, when a K2 workflow event is leveraged a wizard interface is displayed that allows the user to visually configure the elements of that discrete step. For example, in the email event the user is first prompted for the email settings.



E-mail Event

E-mail Settings

Event Name: Confirmation Email

From: administrator@denallix.com

Recipient: Specify Destination User Originator

To: Type text and/or Drop multiple fields

Cc: Type text and/or Drop multiple fields

Bcc: Type text and/or Drop multiple fields

Subject: New Employee Request Recieved - Employee.Name

Specify the name of the event and the e-mail settings, including who the e-mail is from, who it is being sent to, and the subject.

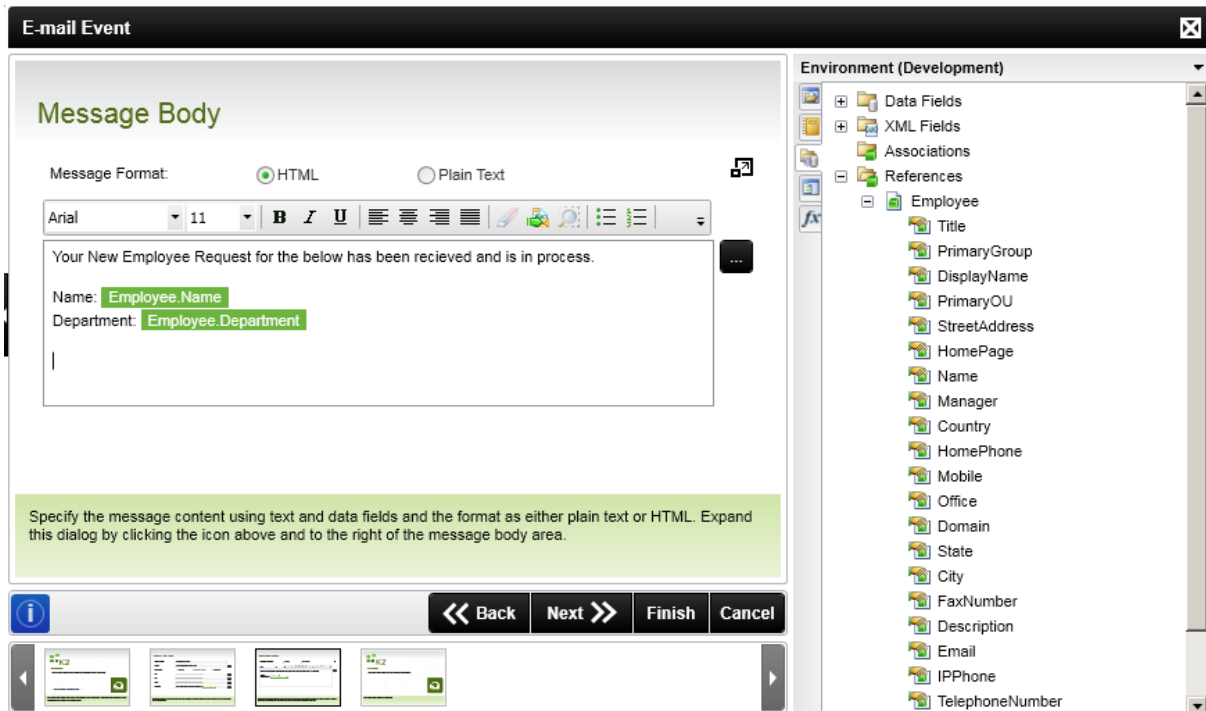
Process/Activity Data

- Data Fields
- Activities
- Employee Onboarding
- XML Fields
- Associations
- References
 - Employee
 - Title
 - PrimaryGroup
 - DisplayName
 - PrimaryOU
 - StreetAddress
 - HomePage
 - Name
 - Manager
 - Country
 - HomePhone
 - Mobile
 - Office
 - Domain
 - State
 - City
 - FaxNumber
 - Description
 - Email

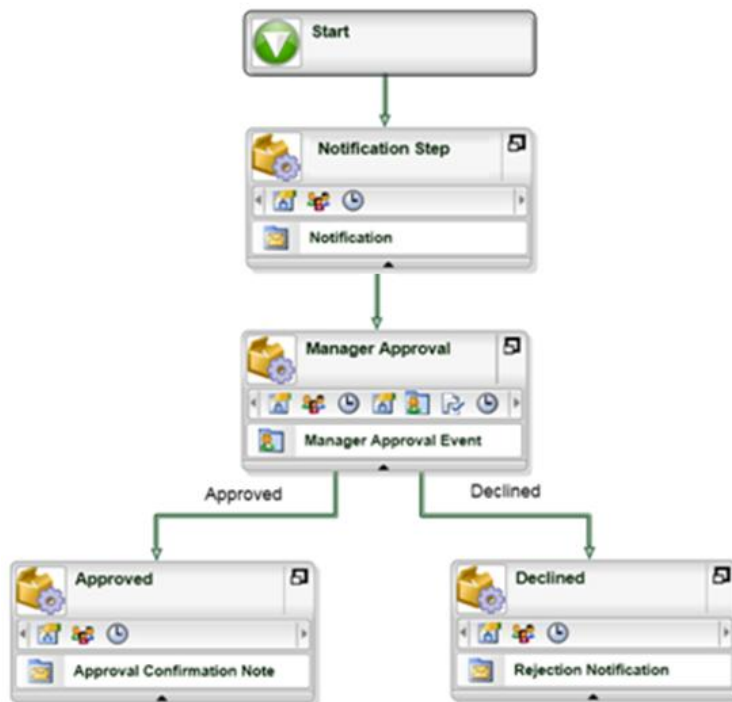
Back Next Finish Cancel

Notice in the “Subject” how there is the ability to blend static text with dynamic data; where “Employee Name” was dragged over from a relationship mapping on the right side. This ability to drag-and-drop dynamic data is carried throughout all K2 wizards, greatly reducing the level of

custom code that must exist within a workflow model. Below shows how the email body can be configured:



Thus, the intent of K2 process modelling is to hide the complexities of the underlying WF from the process designer by walking the designer through a wizard to gather the configuration data and then at runtime executing the configured activity against the underlying WF schedule. The person designing the K2 process does not generally need to worry about this WF level or the underlying code level; he/she simply walks through the K2 wizard and configures the step appropriately. The end result of this experience is a workflow model that closely resembles the process at a business level rather than a technical view of the process. Thus making this a much more accurate reflection of what is being automated as well as promoting proper reporting and data auditing:



K2 leverages WF's strength in using it as a way to structure the underlying execution of K2 processes yet abstracts workflow designers from needing to understand code level intricacies of working with WF directly.

7. ROI Models

In order to demonstrate ROI we need to start with some baselines. For the purposes of this exercise, we classified applications as follows: Small Solution, Medium Solution – SharePoint-based, Large Solution –SharePoint-hosted, complex rules and integration, ASP.NET-based.

7.1. Scenario 1 – Small Solution

In this first scenario, we will assume that the application is 100% SharePoint-based and the overall complexity is light. Examples include simple document routing, small HR-based self-service applications like a leave request or training request. Integration to other systems or data will be light as well.

- Interface – InfoPath or Form Services
- MOSS – Lists and document libraries in a single site
- Process – Light process (6 or less activities) with simple routing logic. (sequential)
- Users – Regional-based only
- Integration – None to pre-existing Web Services built.
- Reporting – Light reporting needs

7.2. Scenario 2 – Medium Solution

In the second scenario, we will add a bit more complexity to the first. The solution will still be 100% SharePoint-based, however we will increase the requirements to include more complex routing logic and involve more activities, users, and data to the application. Examples include IT support ticketing, purchase requests and expense claims.

- Interface – InfoPath or Form Services
- MOSS – Lists and document libraries in a single site. Site provisioning.
- Process – Light process (6 or more activities) with rules-based routing logic. (sequential and parallel)
- Users – Regional and Corporate level
- Integration –Pre-existing Web Services built and new data capture and storage.
- Reporting – Full process and status reporting and some custom reports

7.3. Scenario 3 – Large Solution

In this final scenario, the solution will involve SharePoint as the portal interface, but the overall solution will involve custom forms via ASP.net or Smart Clients. Process logic is assumed to be complex and rules-driven. In addition, data will be pulled from and pushed to various systems. Examples include contracts management, litigation hold, costing and employee on-boarding

- Interface – ASP.NET or Smart Clients
- MOSS – Use of all MOSS features such as search, libraries, site provisioning, records management and security)
- Integration – Various systems such as SQL, Web Services, AD, ERP
- Reporting – Full process and status reporting with highly customized reports that mix process and LOB data.

7.4. Descriptions

Below is a list of assumptions for each description in the costing sheets which follow. Each description has a general high-level of hours associated to the development life cycle.

7.4.1. Build

- Interface – General UI development that includes client side validation and display logic.
- Process – All business logic including routing, rules, tasks, and any abstraction done outside the UI
- Data Integration Process – Any integration such as Web Services, SQL stored procedures and all other data access layer needed within the process or server-side logic.
- Data Integration Interface – Any data access layer calls that are needed for the UI only
- Reporting – Reports include all logging and auditing capabilities, user activity and process statistics for generic status and process detail.

7.4.2. Test

- General Debugging – The general level of effort that goes into debugging an end-to-end solution. This includes logging errors, setting break points and using any tools to aid in finding bugs.
- Interface and Process Testing – The general testing of all UI and process logic for the solution.
- System Integration Testing – The general testing of all data access layer points in the solution.

7.4.3. Deployment

- MOSS Artifacts – Deployment of any MOSS dependencies to any environment. This includes libraries, features, Web parts, etc.
- Interface – Deployment of InfoPath forms or ASP.NET forms.
- Process – Deployment of workflows across multiple environments (Dev,QA,Prod)

7.4.4. Training

- Documentation – General high level functional documentation to be used by BA or end-users.
- System Training – General end-user training of the solution.

7.4.5. Maintenance

- Process Changes – Any change to the workflow or business logic of the solution.
- UI Changes – Any changes to InfoPath or ASP.NET pages.
- Deployment/Versioning – Once changes are complete, this is the effort behind deploying changes and versioning current solution and new solution across all environments.

7.5. Small Project - ROI

Small

Description	WF hours	K2 hours	
Plan			
Requirements	32	32	100%
Build			
Interface	24	10	240%
Process - Modelling and Business Logic	60	8	750%
Data Integration in Process	8	1	800%
Data Integration in Interface	8	1	800%
Reporting (minimum)	32	0	100%
	132	20	660%
Test			
General Debugging	24	8	300%
Interface and Process testing	12	3	400%
System Integration Testing	8	3	267%
	44	14	314%
Deployment			
MOSS Artifacts	1	1	100%
Interface	1	0	100%
Process	8	1	800%
	10	2	500%
Training			
Documentation	16	4	400%
System Training	3	2	150%
	19	6	317%
Total Solution	237	74	320%

Maintenance -year 1

Process Changes	26	4	650%
UI Changes	2	2	100%
Deployment/Versioning	8	1	800%
	36	7	514%

Maintenance -year 2

Process Changes	26	4	650%
UI Changes	2	2	100%

Deployment/Versioning	8	1	800%
	36	7	514%

Maintenance -year 3

Process Changes	26	4	650%
UI Changes	2	2	100%
Deployment/Versioning	8	1	800%
	36	7	514%

Total Maintenance over 3 year	108	21	514%
--------------------------------------	------------	-----------	-------------

Total Solution + Maintenance	345	95	363%
-------------------------------------	------------	-----------	-------------

7.6. Medium Project - ROI

Medium

Description	WF hours	K2 hours	
Plan			
Requirements	60	60	100%
Build			
Interface	60	24	250%
Process - Modelling and Business Logic	160	16	1000%
Data Integration in Process	24	4	600%
Data Integration in Interface Reporting (minimum)	48	8	600%
	316	56	564%
Test			
General Debugging	60	10	600%
Interface and Process testing	45	8	563%
System Integration Testing	10	6	167%
	115	24	479%
Deployment			
MOSS Artifacts	6	3	200%
Interface	4	0	100%
Process	16	1	1600%
	26	4	650%
Training			
Documentation	32	8	400%
System Training	6	4	150%
	38	12	317%
Total	555	156	356%

Maintenance -year 1

Process Changes	60	5	1200%
UI Changes	8	2	400%
Deployment/Versioning	14	1	1400%
	82	8	1025%

Maintenance -year 2

Process Changes	60	5	1200%
UI Changes	8	2	400%
Deployment/Versioning	14	1	1400%
	82	8	1025%

Maintenance -year 3

Process Changes	60	5	1200%
UI Changes	8	2	400%
Deployment/Versioning	14	1	1400%
	82	8	1025%

	246	24	1025%
--	------------	-----------	--------------

	801	180	445%
--	------------	------------	-------------

7.7. Large Project - ROI**Large**

Description	WF hours	K2 hours	
Plan			
Requirements	120	120	100%
Build			
Interface	140	40	350%
Process - Modelling and Business Logic	320	20	1600%
Data Integration in Process	60	8	750%
Data Integration in Interface	40	8	500%
Reporting (minimum)	80	12	667%
	640	88	727%
Test			
General Debugging	115	16	719%
Interface and Process testing	76	11	691%
System Integration Testing	36	13	277%
	227	40	568%
Deployment			
MOSS Artifacts	8	3	267%
Interface	8	7	114%

Process	40	6	667%
	56	16	350%
Training			
Documentation	40	10	400%
System Training	8	5	160%
	48	15	320%

Total	1091	279	391%
--------------	-------------	------------	-------------

Maintenance -year 1

Process Changes	100	8	1250%
UI Changes	32	6	533%
Deployment/Versioning	30	2	1500%
	162	16	1013%

Maintenance -year 2

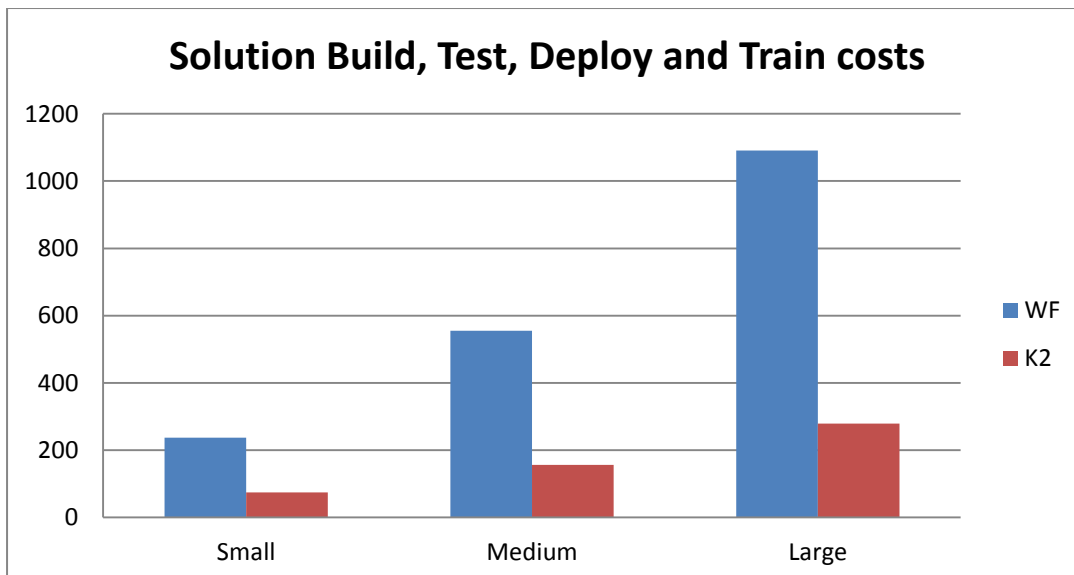
Process Changes	100	8	1250%
UI Changes	32	6	533%
Deployment/Versioning	30	2	1500%
	162	16	1013%

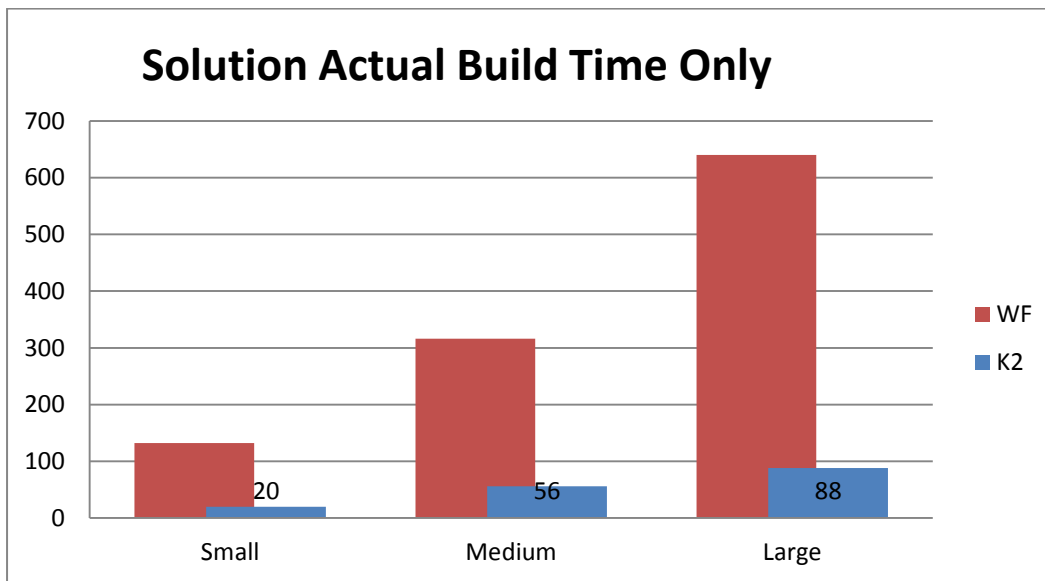
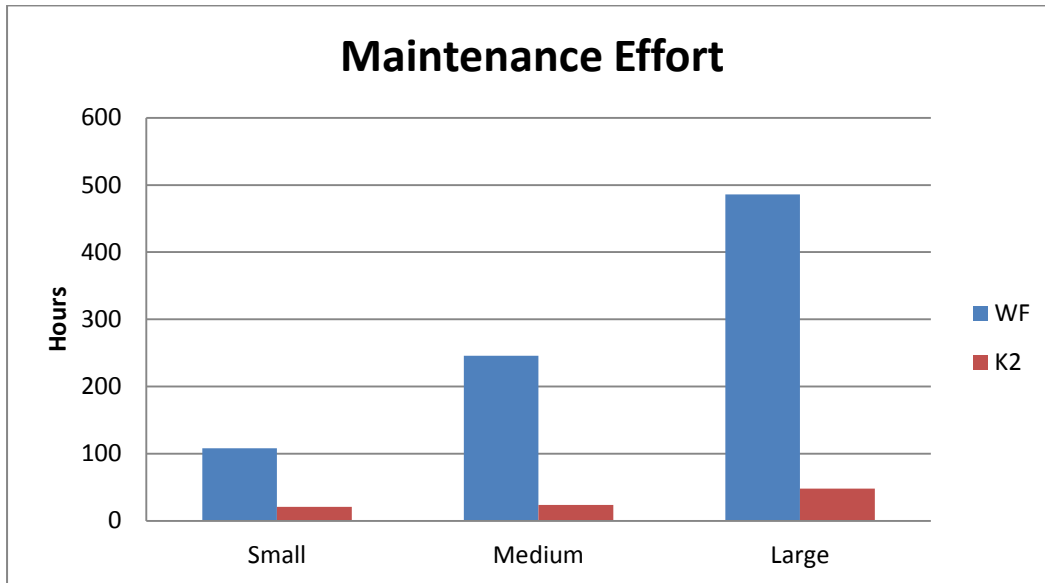
Maintenance -year 3

Process Changes	100	8	1250%
UI Changes	32	6	533%
Deployment/Versioning	30	2	1500%
	162	16	1013%

	486	48	1013%
--	------------	-----------	--------------

	1577	327	482%
--	-------------	------------	-------------





8. Summary

It is very important to understand the roles and design goals of Windows Workflow Foundation (WF) and a business process management platform like K2. Fundamentally a comparison exercise is the wrong departure point. K2 leverages WF as the underlying workflow execution capability. K2 leverages all the strengths of WF while adding significant value in the areas where WF has no coverage, such as robust task management and integrated data provisioning.

Ultimately customers should apply the right technology for the task at hand. Where there is a requirement to build a very low level workflow service as part of an application architecture or perhaps as part of a product, WF is a consideration. If the objective is fundamentally to provide a business solution with workflow or process capabilities, K2 is the correct choice.

Understanding the components that make up a running workflow application is important. The WF “block and line” paradigm along with an execution API has no design goal to be a fully capable workflow, process or BPM platform.

Coding at the base level for workflow applications is similar to the early days of relational databases where initially traditional programming models and indexed flat files defiantly ignored the reality of a new emerging capability around data persistence.

K2 provides a robust process platform built on the .NET platform. The execution capabilities of the K2 server are provided by the WF component of the .NET platform. K2 is recommended for any business facing workflow process and significantly reduces the cost, complexity, time to delivery and maintenance cost of workflow and process-based business solutions.