American Process Inc.
Software Development Capabilities

- **Introduction**

API has built its software development strength with a zero defect mindset, while targeting maximum performance. Maximizing software quality drove API to build its software development capabilities on three concepts: the team, methodology, and the skill matrix to support them.

- **The Team**

One of the most important things that drive the development process employed by API is the team, and the way it works on a project. API has a proven team model but, on client's demand, the team structure can be changed according to the client's needs. The Project Team is subdivided according to specific task and responsibilities as follows:

Project management

- Responds to the customer needs or problem
- Drives the team to a shared vision on how to meet the need or solve the problem
- Answers the question "Why are we doing this?"
- Coordinates the creation and owns the Risk Assessment documents
- Customer advocate to the team and team advocate to the customer
- Public relations, briefings to the customer and senior management
- Communicates the ship dates the customer expects
- Ensures customer satisfaction

Product Management

- Owns the development process
- Moves the project through the development phases and ensures that the right product is developed at the right time
- Coordinates the creation and owns the Master Project Plan and Master Project Schedule as well as the Blue Print (specification)
- Gathers resource requirements to meet the project goals
- Tracks actual costs and scheduled costs
- Provides reports to the stakeholders on costs (a symptom of failed projects is that the reports are delayed until the costs have been exceeded)
- Decides what features will be delivered to meet the agreed upon requirements

Development

- Technology consultants
- Product builder
- Input into high-level design
- Evaluates technology
- Develops prototypes
- Develops proof-of-concept systems to validate potential solutions and to mitigate development risks early in the development process
- Senior developers will be involved in architecture development of the product
- Low-level product and feature design
- In the early phases determines what impact adding/deleting features will have on the project schedule

Testing

- Testing will clearly articulate what is wrong and what is right with a product
- Must know and address the issues before releasing the product. An issue can be a fault in the Development team's code (known as a bug), a deviation from the Program Management team's specification, or a defect in the User Education team's documentation
- Develops test strategies, plans, schedules and scripts

User Education

- Enhance user performance with the product
- Participates in the design process to ensure that the product is useful and usable
- Where user support materials are required the User Education Team designs, builds, and tests the materials that will facilitate user education

Logistics Management

- Responsible for product operation, support, help-desk and distribution channels
- Participates in the design phase to ensure that the product is deployable

- **Methodology**

The API software development framework was developed starting from the Rational Unified Software Development Framework. Like its ancestor is an incremental and iterative process aiming to achieve increments in development through smaller and controllable iterations. The software development framework divides the project lifecycle into several phases, every phase having clear goals and metrics, deliverables and responsibility.

Phases

In the **Request** phase the client starts a discussion requesting assistance from API in developing an application. An API representative contacts the client, views its requirements and the first vision of the application takes shape.

Following the Request the project advances in the **Proposal** phase where a common vision of the project is built among the key stakeholders. This vision should include a mutual understanding of the business needs being addressed and a solid estimation of the project constraints.

The main goal of the **Specification** phase is to formalize and capture all the client's requirements and put them into the Blue Print document. The Blue Print contains exhaustive information on all the functionality and constraints imposed on the project. It will be a joint effort of the customer and developer.

The **Client Acceptance** phase consists of the client's formal acceptance of the Statement of Work and mark the shifting of focus from describing and understanding towards building the application.

The **Application Design** phase is when the application's architecture is defined. The application architecture is based on the Conceptual, Logical, and Physical Design Models. In addition, the three variables with which the team must work: schedule, resources, and application features, are more accurately defined during this phase.

The **Development** phase is based on the understanding of the precedent phases' deliverables. Here the production team performs the project execution, which leads to the application coding being completed and the application being released. The most

important task is to build the application. Iterations, which have been used during the earlier phases, become even more important during the Developing Phase. The team can expect to do multiple iterations of the application during this phase, typically named alpha, beta, and golden release candidate. Additionally, all known bugs should be addressed by the end of this phase. The goal of the Developing Phase is to deliver an application that meets all stated expectations and is ready for external testing.

Based on the Test Plan Document, in the **Testing** phase, significant performance and environmental testing occurs. All known issues are resolved before delivery, and any tasks needed for support and ongoing maintenance of the product are completed. The transition between the Development Phase and the Stabilizing Phase is characterized by the transition from coverage to usage testing.

The **Stabilizing** phase starts when the team shifts its focus from code development to stabilizing the product and ends when the customer accepts the product as complete. A significant aspect of this phase is that the customer and users begin to pilot-test the product. This phase is also the training ground for the organization's operations and support teams. Stabilizing runs in parallel with Testing, and executables and documents are continuously exchanged between the teams involved in the two phases.

The **Deployment** phase concludes the API project development process. At the end of this phase all project deliverables are at the client. The phase is completed when the customer signs the Application Received Receipt.

- **Skill Matrix**

Skill Level	
	4 - Expert (can coach others) **3 - Experienced** (autonomous) **2 - Good** (needs coaching for very difficult problems) **1 - Beginner** (needs constant coaching)

Programming Languages	**Number of skilled employees**	**Average Level**
C/C++		
Linux C/C++	1	3
Visual C++	5	2.9
Java		
Java	6	2.8
.NET		
C#	17	3.9
VB.NET	10	3
J#	3	2.8
Other		
Visual Basic	6	2.8
FoxPro	2	2.8
Delphi	3	3
UML	8	3

Script & Markup Languages			
	Script Languages & Web		
	Jscript / Java Script	8	3
	VB Script	9	3.2
	PhP	5	2.5
	Perl	2	2.5
	Markup Languages		
	HTML	10	3.2
	DHTML	8	3.1
	XML/XSL	8	3.2
	WML	2	2.5
Databases			
	Microsoft SQL Server	12	3.25
	Mysql	7	3
	Informix	2	2.5
	Oracle	5	2.5
	DB2	1	2
	Paradox	1	2
	Sybase	3	2.3
	Xbase	1	2
	Access	6	3
Operating Systems			
	Windows 9x	25	2.9
	Windows NT 4.0	22	2.9
	Windows 2000	25	3
	Windows XP	25	3.8
	Windows 2003	22	3
	Linux	1	3
Networks			
	Windows NT Server	8	3.1
	Windows 2000 Server	7	3.1
	TCP/IP	8	3.1
Programming Technologies			
	Web		
	Cold Fusion	3	2.3
	ASP	6	3.1
	ASP.NET	10	3.2
	JSP/ Servlets	6	2.9
	Microsoft		
	MFC	8	3.1
	.NET	19	3
	STL	4	2.2
	ATL	4	2.8
	COM/DCOM	4	2.7
	MTS/COM+	3	2.1
	MSMQ	5	2.1
	CDO	9	2.2

	DOM	2	2.5
	ISAPI	6	2.5
	SOAP	10	3.3
	ADO	14	3.8
	ADO.NET	17	3.9
	Java		
	Applets and Multimedia (JMF)	2	2.5
	Desktop (Swing, JNI,...)	4	2.25
	J2EE (EJB, JNDI, Servlets/JSP, JDBC, JavaMail, ...)	8	2.8
	Other (JCE, J2ME, ...)	3	2.3
Development Tools			
	Visual Studio .NET	11	3.2
	MS Visual C++	5	3
	MS Visual Basic	3	3
	MS Visual FoxPro	1	3
	MS Visual InterDev	8	3.1
	Reporting		
	Crystal Reports	7	3
	Oracle reports	4	3
	Reporting Services	7	4
App Design			
	Web Site Design	6	3
	Standalone App Design	7	3.1
	Distributed App Design	7	3.1
	Rational Rose	12	2.8
	Database Design	7	3.7
Other Area			
	Agent Design & Programming	2	2.5
	Security (Encryption Techniques)	3	2.8