

IT Overview Benchmark Explain Text for **Application Development & Support**

Gartner, Inc.

September 2011

Controlled and Authorized by:
David Kish
Gartner, Inc.

Table of Contents

1.0 Analysis Views	3
2.0 Spending and Staffing	4
3.0 Customers	8
4.0 Service Levels	9
5.0 ADS Portfolio.....	11
6.0 Applications Support Inventory	12
7.0 Development Project Inventory	16
8.0 ERP Application Support.....	21
9.0 ADS Portfolio Size.....	24

Figure 1. Application Development & Support



1.0 Analysis Views

This analysis examines your Applications Development & Support areas in terms of Gartner’s consensus model. This model is a self-contained, independent, fully functional business model. It is critical to keep in mind that this model is based on functional definitions. This means that some of the activities (and their costs) included within your responsibility may not be part of the Gartner model. Additionally, some activities (and their costs) that are part of the Gartner model may fall outside your area of responsibility. These activities and costs must be included to provide a valid analysis.

Because the comparative cost structure of developing and implementing new applications differs significantly from applications support, development and support efforts are analyzed separately. While you may have individuals or groups that perform both of these functions, their effort and cost should be prorated based on the time and/or cost spent in each area.

Use the following guidelines to distinguish between support and development:

Support Workload	Development Workload
<ul style="list-style-type: none"> ■ Problem discovery and investigation of production application issues ■ Resolution and bug fixes of any size or duration for production application issues ■ User support for applications-related questions ■ Ad-hoc requests for system data ■ Maintenance of hard-coded data or tables (including field size changes) embedded within the programs (any size or duration) ■ Functional enhancements* to current code that take less than two person-weeks ■ Any projects that produce no new business functionality for the user, including perfective, adaptive and preventative maintenance efforts. 	<ul style="list-style-type: none"> ■ New code for new applications ■ Functional enhancements to current code that take more than two person-weeks.

*A functional enhancement is defined as a change made for a user that allows additional capabilities (from a business point of view) that were not there before.

Analysis views include:

- **Applications Support:** This represents support for the applications in the existing portfolio.
- **ERP Applications Support:** This represents support for ERP applications (e.g., SAP, Oracle or PeopleSoft). Note that each unique enterprise application, such as SAP, Oracle or PeopleSoft, will require a separate analysis view.
- **Applications Development:** This represents development efforts during the analysis period. It will include development of new applications, implementation of new packaged applications, enhancements to existing applications and functional upgrades to existing packages.

In addition to the direct costs for Applications Development & Support activities, key supporting functions such as logical database administration, planning and process management, training, etc. are covered. Non-personnel costs for hardware, software and occupancy, are also analyzed and compared.

2.0 Spending and Staffing

General Definitions

The following information must be provided when creating a view of a spending or staffing group.

The annualized Spending and Staffing data is captured in the tables for each analysis view. The major categories include hardware, software, occupancy and personnel. Spending is grouped by Sourcing Types, this is the classification of who is delivering the service for which you are providing cost or head count.

- **Insource:** This includes in-house-related spending and head count.
 - Non-personnel costs should include expense, lease, depreciation, installation and taxes, as appropriate. This also includes maintenance charges that are embedded within the purchase price of assets and, therefore, are inseparable from depreciation.
 - Personnel costs per staff function should include salary, overtime pay, benefits and “other” employee costs such as job-related travel. IS training is collected as an administrative staff function that can either be insourced or outsourced rather than included as a spending load per full-time equivalent head count.
 - Specifically excluded from this analysis are personnel-related costs associated with reductions in workforce, redundancy, relocations and retirement.
- **Contractor:** This includes the spending and head count for contract labor, which is supplemental to your staff and operationally managed by in-house staff.
- **Outsource:** This includes the fees for outsource contracts in which outsource is defined as any situation in which the full operational responsibility for IT services is completely handed over to an external service provider.
- **Maintenance:** This includes the fees for maintenance contracts (i.e., time and materials) that are not embedded in the purchase price of the asset and are, therefore, separable from depreciation charges. Maintenance is differentiated from outsource in that the asset is still operationally managed internally, with the staff calling in maintenance support, as appropriate.

Non-personnel

Software

This includes both Development Software and User Vendor Packages (Business Functionality Software). Development Software includes software (Languages/compilers/databases, Development/Testing tools, IS Management) required by the application development and support staff members to do their jobs. Business Functionality Software includes cost of off-the-shelf user vendor packages. For application development, include the purchase costs for any off-the-shelf vendor packages implemented within the study time frame. If the package implementation spanned multiple years, prorate the purchase costs included in the analysis based on the amount of effort completed on the project in the study year over the total project effort. For application support, provide the annual maintenance cost of off-the-shelf vendor packages.

Occupancy

Occupancy costs should include fully burdened costs for the facilities used by the development or support staff included in this analysis view. Some examples would include office space, furniture, electricity, maintenance, property taxes, security, and office supplies.

Unallocated (Non-personnel): Include costs here only for those non-personnel categories in which a more-detailed accounting is unavailable at this time.

Personnel

When thinking about the staff involved in application development, application support and ERP application support, personnel in some organizations may have more than one distinct role. When determining the view in which to include specific personnel, the following guidelines can be utilized. Individuals can be included in more than one analysis view, but their time should be prorated to avoid double-counting resources.

- **Application Development:** This includes staff involved in developing new applications, enhancing existing applications, installing new packages and installing major functional enhancements to existing packages. If an enterprise application such as SAP is being installed, efforts associated with that installation would be included within this analysis.
- **Application Support:** This includes staff involved in supporting applications that exist within the current portfolio. It includes those personnel who are responsible for fixing programming problems uncovered when applications are running in production. It does **not** include any personnel responsible for running the production applications themselves. If an upgrade for a packaged application primarily contains fixes for existing problems, the efforts involved in installing such a maintenance upgrade are included in this analysis. Efforts associated with supporting an ERP Application such as SAP are not included here but should be analyzed as a separate view.
- **Application Support—ERP Application:** The support environment consists of resources required to support the enterprise application itself as well as other work that is unique to the environment. For example, work to support any interfaces to existing applications, as well as support required for any bolt-on packages that extend the functionality of the primary enterprise applications would be included and analyzed within this view.

Personnel include developers, database technology, quality assurance/testing, infrastructure development, planning and process management, project tracking, business analyst, and administration. Enter only those costs/FTEs that are within the scope of the analysis view.

Unallocated (Total Cost): Include costs which are part of the chart of accounts for those categories in which a more-detailed accounting is unavailable at this time.

Staffing

Full-Time Equivalent Headcount (FTE)

The aggregated FTE (Full-Time Equivalent) counts for employees (entered in the Insourced column/category), outsourcer employees (if known entered in the Outsourced column/category) and contractors (if applicable, entered in the Contractor column/category). Note that the FTE totals entered must correspond to the Personnel costs entered in the previous section of this interview.

Programmer/Analyst (Developer): This includes personnel devoted to developing new applications, enhancing existing applications or maintaining currently operational applications. These individuals may work on all phases of development including conceptual design, systems design, programming and testing of individual programs. Personnel in this category may have more than one distinct role such as performing support and development for a give application and should be prorated accordingly between the analysis views to avoid double counting. **Typical roles include: Application Programmers, Application Developer Analysts, Systems Analysts and Package Configuration Staff.**

Application Support includes staff who maintain currently operational applications. These individuals typically receive level two trouble tickets (defects). Support personnel then modify the application, as necessary, to meet these requests. A person is considered to be in support if s/he handles the project after it has been identified as a program problem. For example, if a help desk person takes initial problem calls and serves primarily as a “router,” then this person is not included in the analysis because the problem has not been established yet to be a programming issue. When a problem call is determined to be a programming issue, or at least believed to be such, the time to diagnose and fix the problem is captured in the study. Personnel who are available to business users to assist them on applications-related problems or requests also are in this group. This may involve working with the business user to assist in report generation or electronic data-gathering requirements. Consider only applications-related personnel, not hardware support personnel or standard commercial PC software package support. This also includes individuals performing enhancement projects that are less than or equal to two person-weeks.

Applications Development includes staff who work closely with customers, business analysts, and team members to understand business requirements that drive the analysis and design of quality technical solutions (new applications and functional enhancements to existing applications that are greater than two person-weeks of effort). Staff is involved in the full systems life cycle and is responsible for designing, coding, testing and implementing application software. Staff may make recommendations towards the development of new code or reuse of existing code. Responsibilities may also include participation in component and data architecture design, product evaluation and buy versus build recommendations. Has experience in systems analysis, design and a solid understanding of development, quality assurance and integration methodologies.

Database Technology: This includes personnel devoted to designing, evaluating, creating and maintaining the logical data definitions and structures used by the enterprise. These personnel may be prorated between application support and application development. This can involve evaluation of the requirements during the application design phase as well as advising developers doing logical database design and physical database design. This also includes personnel devoted to developing or maintaining test databases required for testing maintained code. Evaluation of new database technologies should not be shown here but should be shown in the planning and process management section of this module.

Security, as it applies to data, should not be included at all in this evaluation, nor should those performing the following functions: physical support of the production databases including taking care of the actual physical structure of the space, knowing and using/installing the software associated with supporting the databases, applying patches and upgrades, diagnosing and resolving systems problems, creating the data space and managing the utilities related to performance (tuning) and backing up of the data.

Quality Assurance/Testing: This includes personnel dedicated to testing or evaluating software before it is cut over to user testing or production. These personnel may be prorated

between application support and application development. These individuals typically work with the development and support personnel and, at the appropriate time in the project life cycle, assume responsibility for determining fitness of the new or revised product. This would not include testing of individual programs that normally would be handled by the programmer/analyst personnel themselves. An EDP audit staff would be included only if it reports to IT and does its job before systems are moved to production. The IS organization must have time to make corrections or revisions before passing the application onto users. Coordinating user acceptance testing is included in the scope of the analysis. However, user acceptance testing performed by business staff is excluded from the analysis.

Infrastructure Development: This area includes software developers who are building software “building blocks” that serve no immediate user functional purpose but are the components to be used by developers when building future user software, (i.e., components such as middleware or base-class libraries).

Planning and Process Management: This includes personnel associated with planning and process-related activities associated with the applications area. This would include people working on methodology, standards and metrics. Typically the total personnel in this category will be prorated between development and support. The area also includes personnel devoted to evaluating new hardware and software products. These personnel may be prorated between application development and application support. These are people who choose what to bring in for testing, arrange for the tests, carry out the tests or coordinate the testing, and report to management or make decisions on what to use. This applies only to the application development department and, hence, would not include system software or mainframe hardware (except in rare circumstances).

Project Tracking: This includes personnel who are devoted to monitoring the progress of projects using standard project management techniques or applications as well as personnel who are assigned specifically to creating project estimates. These personnel may be prorated between application support and application development.

Business Analyst: This includes staff members who are familiar with business processes and procedures that translate high-level business requirements into functional specifications and manage business requirements throughout the life of the project. These staff typically resides within the business, but work closely with the applications area. They must have strong business domain expertise and a reasonable understanding of how technologies can enable business opportunities and redesign.

Services Administration: This category includes staff dedicated to the budget, chargeback, service level reporting, product management, training and documentation, asset management, and account management.

Management and Administration: This category includes management and IS administration staff.

Unallocated (Personnel): Include costs/FTEs here only for those personnel categories for which a more-detailed accounting is unavailable at this time.

3.0 Customers

ADS Customers

- **Employees:** Indicate the count of employees (i.e., head count excluding contractors, outsourcers and consultants) in the business units associated with the projects and/or applications considered in the scope of this analysis view.
- **End users:** Indicate the count of people that may be using the applications supported or are benefiting from the projects implemented and are considered in the scope of this analysis view. If users are accessing multiple applications or will be benefiting from multiple projects, do not double-count them here.
- **Number of Applications:** Indicate the number of applications included in this assessment.
- **Number of Projects:** Indicate the number of applications projects included in this assessment.

4.0 Service Levels

ADS Service Levels – Applications Support

This table captures information regarding the level of service delivery to evaluate both planned and achieved service levels during the analysis year.

Number of Severity 1 Problems: Enter the total number of Severity 1 application problems during the 12 month study timeframe.

Time to Fix - Severity 1 Problems (hours): Enter "Target" and "Actual" service levels for the time to restore Severity 1 application problems.

Number of Severity 2 Problems: Enter the total number of Severity 2 application problems during the 12 month study timeframe.

Time to Fix - Severity 2 Problems (hours): Enter "Target" and "Actual" service levels for the time to restore Severity 2 application problems.

ADS Service Levels – Applications Development

This table captures information regarding the level of service delivery to evaluate both planned and achieved service levels during the analysis year.

Percent of Development Projects Executed During the Study Period that were or will be Completed on Time: Estimate the percent of all application development projects completed within the last year that were delivered by target deadline originally agreed with the customer at the time of formal project approval.

Average Percent Schedule Variance for Projects not completed on Time: For those projects (if any) completed that exceeded the originally agreed target deadline, please indicate your estimate of by how much (extra days divided by total elapsed project duration) they exceeded their planned lifecycle (expressed as average %) at the time of formal project approval.

Percent of Projects Executed During the Study Period that were Completed on Budget: Estimate the percent of all application development projects completed within the last year that were delivered within the budget cost originally agreed with the customer at the time of formal project approval.

Average Percent Budget Variance for Projects not Completed on Budget: For those projects completed that exceeded the originally agreed budget, please indicate your estimate of by how much (final cost divided by original budget cost - 1) they exceeded their planned budget cost (expressed as average %) at the time of formal project approval.

Number of Projects Implemented during the study Year (went into Production): Indicate the total number of projects that were implemented into production during the 12 month study timeframe.

Number of Active projects Cancelled Prior to Delivery during the study Year: Provide the number of application projects which were given clearance through the governance process, and approval to proceed, but were cancelled either before work was started, or before delivery into production.

Number of Projects Planned but not Started (backlog): Provide a count of how many applications development projects which have been given a go-ahead through the governance process, and hence approval to proceed, that are still waiting to be started.

Percent of Projects Taking More Than One Year to Complete: Express as a percent of the total number of active development projects the number of currently active development projects that have been running for more than one year.

Total Number of Defects reported during the first 3 months of implementation: Provide the total number of defects detected and logged in the first 90 days after delivery into production (from all applications that went live) during the past year.

5.0 ADS Portfolio

The Applications Development & Support project workload may cover tens or hundreds of application systems in support/maintenance and active projects in development. Gartner will work with you to reduce the risk of data-collection overload. Gartner recommends a focused approach be taken for the inventory data collection where information is gathered for a sample set of application systems and/or development projects. The criteria used to collect the sample includes: selecting the largest systems and projects, those with the highest criticality to the business and greatest effort to build and maintain. Our research shows that 80 percent of the functionality in the portfolio resides in 30 percent of the largest application systems and projects. Sensible implementation of this subset approach is a key component in the success of the data-collection phase of the project.

For the Application Support view, each application in the current inventory that is identified in the subset for the detailed inventory will need to be listed, along with some information regarding that application. Applications being supported may include applications supported in-house (that is, supported by in-house programmers or contractors, vendor packages for which a maintenance contract may be in place but for which the initial problem determination is done by your organization before the vendor is contacted), and applications that are supported in some sort of outsourced arrangement.

For the ERP Applications Support view, a single detail record will be listed. This is, in effect, an inventory list with only one application included. For example, if SAP is being supported, there will be a single ADS portfolio detail record for this particular analysis view.

For the Application Development view, each development project undertaken during the analysis period that is identified in the subset for the detailed inventory will be listed here. If there are multiple enhancement projects against a single application, they can be combined into a single detail record in this section. Projects may include brand new applications being developed in-house and off-the shelf packages being installed (and possibly customized as well). It would also include enhancements to existing applications or functional upgrades to existing packages. There may be selected applications that are being developed or enhanced by an outsourcer. Those would be included. If all application development is outsourced, those projects also would be included here. If a purchase enterprise application such as SAP is being implemented, it would be listed as a project within this view.

6.0 Applications Support Inventory

Application Name

Provide a name that identifies the particular application.

Sizing Method

Sizing documentation appears in a separate section. That documentation should be referenced for relevant information regarding sizing methods and how to size applications. Based on the Sizing Method selected, the relevant fields will be highlighted in the ADS Portfolio Size section below. Sizing options included:

- **Size Template**
- **Lines of Code by Language Type**

Year Installed

Indicate the implementation year for the application. Specify the year that the current version of the application was implemented if it has been replaced by an equivalent product (i.e., substantially rewritten).

If the application was implemented during the analysis year or discontinued during the analysis year, indicate the number of months in production in the study year.

For applications implemented during the analysis year, include a project representing the development effort in addition to the detail for the supported application.

Application Type

For each application, please select one of the following types.

- **In-House:** This application is maintained in-house. This can include vendor package software where support has been taken over in-house and a maintenance contract is no longer maintained with the package vendor. This can also include code where the development was outsourced to a third-party vendor or external service provider and the support is now performed in-house.
- **Vendor Package:** The code for this application comes from off-the-shelf purchased software (i.e., accounting package). The package may or may not contain additional customization. A maintenance contract is maintained with the package vendor.
- **Outsourced:** This application is being supported by an outsourcer.
- **Outsourced and Vendor Package:** This application is a vendor package and is being maintained by an outsourcer. A maintenance contract is maintained with the package vendor.

Application Architecture

Provide a percentage breakout of the application architecture. If the application is of one primary architect type, please enter 100% in the appropriate section. If the application has multiple architectural components, please provide a percentage breakout totaling to 100%.

Mainframe: A large-capacity computer. Traditionally, mainframes have been associated with centralized, rather than distributed, computing environments.

Client Server: The splitting of an application into tasks performed on separate, network-connected computers, at least one of which is a programmable workstation such as PC. In most cases, the “client” is a desktop computing device (i.e., a PC) or a program “served” by another networked computing device (i.e., the “server”). Applications to consider in this category include those that handle a high amount of the application logic and data management tasks, rather than having these tasks executed remotely on the server (see Web/Thin Client).

Web/Thin Client: This includes websites as well as applications that can be accessed from the website. It also refers to a type of client/server computing where applications are run, and data is stored, on the server rather than on the client. Because the applications are executed on the server, they do not require client-resident installation, although the graphical user interface and some application logic may be rendered to the client.

Service-Oriented: An application topology in which the business logic of the application is organized in modules (services) with clear identity, purpose and programmatic-access interfaces. Services behave as “black boxes”: Their internal design is independent of the nature and purpose of the requestor. In Service-Oriented Architecture, data and business logic are encapsulated in modular business components with documented interfaces. This clarifies design and facilitates incremental development and future extensions.

Desktop/Client Only: This includes applications where the application’s data management, logic and presentation reside on a desktop/client only.

Package Customization (% Effort)

If the application is a vendor package, indicate the amount of effort required during the 12-month analysis time frame to support business-specific customization.

Very low (less than 10%)

Low (10% to 20%)

Medium (21% to 30%)

High (31% to 60%)

Very high (> 60%)

Interfaces to Other Systems

Indicate the amount of effort required during the 12-month analysis time frame to support integration of this application with other applications when changes to the system are made.

None: It is a completely stand-alone application.

Low: Connections to other applications exist, but only a minor amount of work (typically less than 10% of the labor effort) is required in supporting these connections when changes are made.

Medium: Connections to other applications exist, and a reasonable amount of work (typically 10% to 25% of the labor effort) is required in supporting these connections when changes are made.

High: Connections to other applications exist, and a major amount of work (typically more than 25% of the labor effort) is required in supporting these connections when changes are made.

ADS Portfolio Labor and Costs

Indicate the amount of Programmer/Analyst (Developer) effort required to support each application during the 12 month analysis time frame. Support includes defect repair, routine user support, technical upgrades and conversions (i.e., language or operating system upgrades) and preventative maintenance. Support also includes functional enhancements that take less than two person-weeks to complete and provide fewer than eight function points. Support may be provided by in-house personnel, contractors or by an outsourcer. Remember to include project leader time. Outsourced support occurs when outside vendors are used to support the application (including the day-to-day management). Outsourced software may have originated as in-house developed, package or purchased custom software.

Full-time equivalent person-years (FTEs) should be measured in calendar time (i.e., if an individual works full time on a project for a full year, that is one person-year). Please note that you would not subtract such things as vacation time, sick days and administration time. If your project-tracking application shows, for example, 220 days actually worked, that may represent close to a full person-year for you. The conversion factor you use to go from hours or days into years depends upon the amount of overhead time, how you record overtime and the way you enter data into your tracking application

Outsourcer Costs

This includes the costs for outsourced support for this application.

ADS Portfolio Size

Sizing documentation appears in a separate section. That documentation should be referenced for relevant information regarding sizing methods and how to size applications. Based on the Sizing Method selected, the relevant fields will be highlighted.

See Section 9.0 for more detail.

ADS Portfolio Vendor Packages

For any application that is listed as a vendor package, the following information is required under “ADS Vendor Package Detail.”

Vendor Package Name

Provide the name of the package and/or module.

Version

Provide the version number of the package.

Vendor Name

List the vendor of the package.

Annual Maintenance Fee

List the annual maintenance charges for the package.

ADS Portfolio Defects

A defect is an unplanned problem that causes an application to fail to execute or creates the possibility of erroneous results. The failure of an application to meet its intended functionality, if not previously known and approved, is also considered a defect. If a defect occurs at multiple locations running separate copies of the code, then it should count each time it is reported. If it is decided not to correct a defect, then it still should be counted.

Total Number of Defects

For the support view, indicate the number of valid defects that occurred in support or production for this application during the 12-month analysis time frame. Note: Applications that were installed or deinstalled during the study period should not be included if there is not a full 12-month period of defect tracking.

Defect Severity

Provide a breakout of the support defects, totaling to 100%, for the following severity levels:

Critical: Application will not run, aborts spontaneously, or is providing serious data errors or data corruption; it impacts the customer.

Major: Application runs, but major data errors are possible, or major functionality is unavailable; it may impact the customer.

Minor: Minor data errors are possible, or minor functionality is unavailable, with no significant impact on the customer.

Cosmetic: Examples include heading errors, minor fields missing and formatting problems with little true impact on customer.

Service Requests

Service Request (SR) data for supported applications include the number of valid requests for technical assistance (logged or unlogged) that occurred in support of the application portfolio during the 12-month analysis time frame. Note: Applications that were installed during the study period should include the total number of SR's opened and closed during the partial year.

Information requested includes:

Number of Service Requests Currently Open: the total number of active, open or outstanding Service Request at the end of the 12-month study time frame.

Number of Service Requests opened during the study year: the total number of service requests that were opened during the 12-month study time frame

Number of Service Requests closed during the study year: the total number of Service Requests that were closed during the 12-month study time frame. This can include Service Requests that were closed during the study year, but opened prior to the analysis time frame.

7.0 Development Project Inventory

For each development project listed in the portfolio, certain detailed information that will need to be captured.

Project Name

Provide a name that identifies development project. If there are multiple enhancements to an existing application, all enhancements may be combined for the purposes of reporting here.

Sizing Method

Sizing documentation appears in a separate section. That documentation should be referenced for relevant information regarding sizing methods and how to size applications. Based on the Sizing Method selected, the relevant fields will be highlighted in the ADS Portfolio Size section below. Sizing options included:

- **Size Template**
- **Lines of Code by Language Type**

Project Dates

Indicate the month and the year that the project started and was implemented or the target for implementation.

- **Start date:** The date that requirements analysis began for this project.
- **Implementation date:** The date the project was first released to the Customer.

Development Category

For the development view, indicate the development category. Please select one of the following types.

- **New:** Primarily a brand new application. If an existing application is being replaced by a new application and functionality is being redesigned or expanded, then it is a new application
- **Enhancement:** A functional enhancement is defined as a change made within the bounds of a current application for a user that allows additional capabilities (from a business point of view) that were not there before. The size of the enhancement(s) should reflect all functionality added, changed and/or deleted for the project.

For example, if the ability to handle a special discount structure were being created, then that would count as additional functionality for the user. If, on the other hand, an application allows a sale of up to \$99,999.99 and the programs are being expanded to include more than seven digits, then this would be considered technical in nature and not additional user functionality. Labor associated with this effort would be captured in the support view rather than the development view.

Functional enhancements to current code that take less than two person-weeks and typically add fewer than eight function points are counted in the support view.

In some environments, major enhancements actually can be added in fewer than two person-weeks. If this is the case, and more than eight function points are added (about 800 lines of

COBOL or 300 lines of a database language), this enhancement is recorded as a project and marked as an enhancement.

When considering new vs. enhancement, consider an enhancement to be changing or adding to existing code in such a way that you are substantially constrained to using the technology of the existing application.

If the project involves re-engineering or reworking an application to use relational database technology but without other functional redesign or is a straight conversion (e.g., porting from one database to another) with little or no design effort, this effort should be categorized in the support view rather than in the development view as little to no new business functionality is being developed from a users' perspective.

Application Type

For each project, please select one of the following types.

- **In-House:** This application is developed in-house.
- **Vendor Package:** The code for this application comes from off-the-shelf purchased software (e.g., accounting package). The package may or may not contain additional customization.
- **Outsourced:** This application is being developed by an outsourcer.
- **Outsourced and Vendor Package:** This application is a vendor package and is being installed and customized, or maintained by an outsourcer.

Year-End Status

Indicate the status of the development project at the end of the 12-month study year analysis time frame.

Done: Completed and turned over to production

In progress: Continuing on into next year

On hold: Not active at all for up to three months; continuation uncertain

Dropped: Abandoned, decided not to continue to next phase, or on hold for more than three months with continuation uncertain

Application Architecture

Provide a percentage breakout of the application architecture. If the application is of one primary architect type, please enter 100% in the appropriate section. If the application has multiple architectural components, please provide a percentage breakout totaling to 100%.

Mainframe: A large-capacity computer. Traditionally, mainframes have been associated with centralized, rather than distributed, computing environments.

Client Server: The splitting of an application into tasks performed on separate, network-connected computers, at least one of which is a programmable workstation such as PC. In most cases, the "client" is a desktop computing device (e.g., a PC) or a program "served" by another networked computing device (i.e., the "server"). Applications to consider in this category include those that handle a high amount of the application logic and data management tasks, rather than having these tasks executed remotely on the server (see Web/Thin Client).

Web/Thin Client: This includes websites as well as applications that can be accessed from the website. It also refers to a type of client/server computing where applications are run, and data is stored, on the server rather than on the client. Because the applications are executed on the server, they do not require client-resident installation, although the graphical user interface and some application logic may be rendered to the client.

Service-Oriented: An application topology in which the business logic of the application is organized in modules (services) with clear identity, purpose and programmatic-access interfaces. Services behave as “black boxes”: Their internal design is independent of the nature and purpose of the requestor. In Service-Oriented Architecture, data and business logic are encapsulated in modular business components with documented interfaces. This clarifies design and facilitates incremental development and future extensions.

Desktop/Client Only: This includes applications where the application’s data management, logic and presentation reside on a desktop/client only.

Package Customization (% Effort)

If the application is a vendor package, indicate the amount of effort required during the 12-month analysis time frame to support business-specific customization.

Very low (less than 10%)

Low (10% to 20%)

Medium (21% to 30%)

High (31% to 60%)

Very high (> 60%)

Interfaces to Other Systems

Indicate the amount of effort required during the 12-month analysis time frame to support integration of this application with other applications when changes to the system are made.

None: It is a completely stand-alone application.

Low: Connections to other applications exist, but only a minor amount of work (typically less than 10% of the labor effort) is required in supporting these connections when changes are made.

Medium: Connections to other applications exist, and a reasonable amount of work (typically 10% to 25% of the labor effort) is required in supporting these connections when changes are made.

High: Connections to other applications exist, and a major amount of work (typically more than 25% of the labor effort) is required in supporting these connections when changes are made.

ADS Portfolio Labor and Costs

Show the appropriate Programmer/Analyst (Developer) work effort required or expended for each project. Development includes new code for a new application and functional enhancements to current code that take more than two person-weeks or typically add greater than eight function points. Development includes all phases of development including conceptual design, systems design, programming and testing of individual programs. Development may be provided by in-house personnel, contractors or by an outsourcer. Remember to include project leader time. If an outside vendor's package has been chosen,

show only the person-years by your staff in choosing the package and preparing it for production.

The three labor numbers requested for development projects are:

- **During Study Year:** Represents the amount of labor expended during the 12 month analysis time period.
- **Before Study Year:** Represents the amount of labor expended on the project before the analysis time period. This may be zero if the project started within the study year.
- **Remaining:** Represents the amount of labor expended on the project after the analysis time period. May be an estimate if the project is in progress or zero if the project is completed. For incomplete projects, use the best latest estimate for the total project requirements.

Person-years should be measured in calendar time (i.e., if an individual works full time on a project for a full year, that is one person-year). Please note that you would not subtract such things as vacation time, sick days and administration time. If your project-tracking application shows, for example, 220 days actually worked, that may represent close to a full person-year for you. The conversion factor you use to go from hours or days into years depends upon the amount of overhead time, how you record overtime and the way you enter data into your tracking application

For the development view, provide a breakout of the labor by the following categories: Insource, Contractors and Outsourcer.

Insource: This includes in-house staff managed by the organization.

Contractors: This includes contractors that are used as extensions of the in-house staff and are managed by the organization on a day-to-day basis.

Outsourcer: This includes staff that is managed by a third-party vendor or external service provider. Outsourcers differ from contractors in that an outsourcer is engaged typically to provide a service or deliverable rather than a specific skill set and typically is not managed by the organization on a day-to-day basis.

Outsourcer Costs

This includes the costs for outsourced development for this project.

ADS Portfolio Size

Sizing documentation appears in a separate section. That documentation should be referenced for relevant information regarding sizing methods and how to size applications. Based on the Sizing Method selected, the relevant fields will be highlighted.

See Section 9.0 for more detail.

ADS Portfolio Vendor Packages

For any project that is listed as a vendor package, the following information is required under “ADS Vendor Package Detail.”

Vendor Package Name

Provide the name of the package and/or module.

Version

Provide the version number of the package.

Vendor Name

List the vendor of the package.

Purchase Price

Provide the full purchase price of the package.

8.0 ERP Application Support

ERP Application Name

Provide the name of the ERP Application (i.e., SAP, Oracle, PeopleSoft, etc.)

Year Installed

Indicate the implementation year for the application. Specify the year that the current version of the application was implemented if it has been replaced by an equivalent product (i.e., substantially rewritten).

Named Accounts

Provide the total number of users with the ability to access the PeopleSoft application.

Concurrent Users

Provide the total number of users expected to access the application concurrently at a given time during a representative time period. Use average number of concurrent users expected rather than peak users.

Package Customization (% Effort)

If the application is a vendor package, indicate the amount of effort required during the 12 month analysis time frame to support business-specific customization.

- Very low (less than 10 percent)
- Low (10 percent to 20 percent)
- Medium (21 percent to 30 percent)
- High (31 percent to 60 percent)
- Very high (> 60 percent)

Support Labor During Study Year

Indicate the amount of Programmer/Analyst (Developer) effort required to support the ERP application during the 12 month analysis time frame. Support includes defect repair, routine user support, technical upgrades and conversions (i.e., language or operating system upgrades) and preventative maintenance. Support also includes functional enhancements that take less than two person-weeks to complete and provide fewer than eight function points. Support may be provided by in-house personnel, contractors or by an outsourcer. Remember to include project leader time. Outsourced support occurs when outside vendors are used to support the application (including the day-to-day management). Outsourced software may have originated as in-house developed, package or purchased custom software.

Person-years should be measured in calendar time (i.e., if an individual works full time on a project for a full year, that is one person-year). Please note that you would not subtract such things as vacation time, sick days and administration time. If your project-tracking application shows, for example, 220 days actually worked, that may represent close to a full person-year for you. The conversion factor you use to go from hours or days into years depends upon the amount of overhead time, how you record overtime and the way you enter data into your tracking application

Outsourcer Costs

This includes the costs for outsourced support for this application.

Version

Provide the version number of the package.

Vendor Name

List the vendor of the package.

Annual Maintenance Fee

List the annual maintenance charges for the package.

Annual License Charge

List the annual license charges for this package, if applicable. Some applications do not have a single purchase price but are licensed year to year based on payment of an annual license charge. Because the annual license charge may represent a combination of purchase price and an annual maintenance charge, provide a breakout of the charges for the right to use the software versus the annual maintenance charges.

Functional Class

Check the appropriate box(es) that lists general enterprise application systems functions (e.g., ERP, CRM and SCM). If the function is not listed, check "Other," and list the function in the field provided.

ADS Portfolio Defects

A defect is an unplanned problem that causes an application to fail to execute or creates the possibility of erroneous results. The failure of an application to meet its intended functionality, if not previously known and approved, is also considered a defect. If a defect occurs at multiple locations running separate copies of the code, then it should count each time it is reported. If it is decided not to correct a defect, then it still should be counted.

Total Support Defects

Indicate the number of valid defects that occurred in support or production for this application during the 12 month analysis time frame.

Level of Support Defects

Provide a breakout of the support defects, totaling to 100%, for the following severity levels:

- **Critical:** Application will not run, aborts spontaneously, or is providing serious data errors or data corruption; it impacts the customer.
- **Major:** Application runs, but major data errors are possible, or major functionality is unavailable; it may impact the customer.
- **Minor:** Minor data errors are possible, or minor functionality is unavailable, with no significant impact on the customer.

- **Cosmetic:** Examples include heading errors, minor fields missing and formatting problems with little true impact on customer.

Service Requests

Service Request (SR) data for supported applications include the number of valid requests for technical assistance (logged or unlogged) that occurred in support of the application during the 12 month analysis time frame.

Information requested includes:

- **The number of open Service Requests:** the total number of active, open or outstanding Service Request at the end of the 12 month study time frame.
- **Service Requests opened during the study year:** the total number of service requests that were opened during the 12 month study time frame.
- **Service Requests closed during the study year:** the total number of Service Requests that were closed during the 12 month study time frame. This can include Service Requests that were closed during the study year, but opened prior to the analysis time frame.

9.0 ADS Portfolio Size

The unit of measure used for sizing is function points. The function point represents a unit of measure of the automated functionality delivered to the end user by an application. It quantifies the work performed for the end user and is a unit of measure that is independent of the technologies, processes or platforms that are utilized within the enterprise. It is a unit of measure that is linear, scalable and comparable. For example, an application that is considered to have 1000 function points is twice the size of a different application sized at 500 function points. An application that is sized at 3000 function points that is written in COBOL is functionally equal to an application written in Java that also is sized at 3000 function points.

Function point analysis has been applied within Gartner for benchmark analyses to normalize data across heterogeneous environments. The standards and definitions for precisely counting function points are controlled by the International Function Point Users Group (IFPUG). In summary, function points are used to examine five activities that a system can perform for the user:

- Maintaining internal data files
- Allowing a system to use external data files from other systems
- Allowing input from users or other applications
- Providing user reports and other outputs
- Allowing user inquiries

IFPUG provides detailed instructions for counting these activities. Complexity factors as well as 14 general system characteristics are then applied to arrive at a weighted result of the actual count for each application or project. By assembling this information and doing the carefully prescribed calculations, a total function point count for the system is determined. Currently, this is still considered the best measure of the size of an application or project. Utilizing the full function point counting process can be fairly time-consuming, so several estimating methodologies have been developed, and the choice of which to utilize depends on the level of accuracy required for the analysis.

When actual function point counts are not available, the sizing options that are available consist of the following:

Size Template: This represents a very effective estimating methodology, based on analyzing a large number of applications that have been sized. It provides a fairly high degree of accuracy for applications developed using any technology and is recommended for most 4GL applications or any packaged applications or applications in which code generators have been utilized when a full function point count is not available. It can be used with any technology.

Lines of Code Counting: This method is used with older architected environments, although it can be effective when used with procedural languages, scripting languages. It can also be extremely effective with applications that have a high percentage of batch processing and batch customization in a scheduled environment.

Size Template Method

For Size Template, count all the requested entities in the underlying application for supported applications and new development. For enhancement projects, count only those entities that were added to the underlying application(s), that were changed or that were deleted from the underlying application(s).

Logical Tables/Structures

Provide the number of logical tables and structures that are maintained by the application and used by the application.

Logical tables and structures that are maintained by the application.

If this application can write to a data structure, count it here.

If you have a logical and normalized data model for this application, provide the **number of logical tables** that contain user-updateable or viewable data. Do not include tables that contain system-related data (e.g., transaction logs) unless an end user can read or maintain the data through the application itself.

If a logical data model is not available, count the number of physical tables in the production copy that contain user-updateable or viewable data. Do not include tables that contain system-related data (e.g., transaction logs) unless an end user can read or maintain the data through the application itself. For DB2, Oracle, Sybase and MS Sequel server-based applications, this is the number of physical tables that contain user-updateable data. For COBOL, it's usually the number of 01-level record types in master files. Check with your research analyst for assistance in discovering similar structures for other data structures.

The following table can be used to find equivalent data structures if the data storage method is not a database.

Data Organization Method	Type	Data Structure
Flat files	Flat	Record type
VSAM	Indexed	Files
DB2	Relational	Tables
IMS	Hierarchical	Segments, records
NOMAD	Relational	Tables
NOMAD	Hierarchical	Segments
Oracle	Relational	Tables
IDMS	Hierarchical	Records
ADABAS	Relational	Sets
DBASE	Relational	Tables
Access	Relational	Tables
Sybase	Relational	Tables
Indexed	Indexed	Files
Relational DB	Relational	Tables

ISAM	Indexed	Files
Paradox	Relational	Tables
Object-oriented	Relational	Objects

The following examples would qualify as files, if the application can add, change and/or delete information within that file:

- A table of business data
- A logical table of security, audit or history data that is called for by user requirements
- A table of lookup data, edit criteria, help or error messages and more that is not embedded in program code

The following do not qualify as structures:

- A temporary, work or sort file passed between job steps
- A table that is embedded in program code
- A backup file used only for standard backup/recovery

Logical tables and structures that are used by the application.

If there are any databases or other structures that this application can reference but cannot update, indicate them here. Typically, these are structures that are read-only for the application being analyzed (i.e., “pulls” by this application from another application). Note: Pulls that update data are handled under inputs, and pushes from this application are considered in outputs. The structures counted in this section for the receiving application are for read-only or reference purposes. These structures also would be counted within the bounds of the application that maintains those structures under structures maintained by this application.

An example of a structure used by this application is a billing system that has a customer address table that is used to print the addresses on bills. The customer addresses are maintained within the bounds of another application with no billing system tables being updated with this data.

Common structures used by this application (read-only) include validation tables and display-only data.

Logical Inputs

Provide the number of screens/forms that maintained unique business data and that browse, inquire or display existing data.

These are logical transactions that enter the application and whose main purpose is to maintain one or more of the *Logical Tables/Structures*, or to control or alter the behavior of the application. Indicate the number of unique logical inputs that maintain (i.e., add, change or delete) data within the application. These are screens, forms and more that add, change, delete or otherwise maintain and save end-user data. Security functions should be included if, for

example, screens are provided to maintain security data. Menu screens that do not update any end-user data should not be included.

It is important to note that a logical input may take up more than one physical screen or form or may exist with other unique inputs on a screen or form. The basic question to ask is whether the user action can be performed independent of any other user action. For example, if a user can add an employee dependent (e.g., a newborn child) to a payroll application independently from adding or changing employee data, then the action of adding the dependent is a unique transaction and should be counted.

Definition of Logical forms/screens/inputs that maintain unique business data: These are the typical online screens and forms that end users use to add, change or otherwise maintain the data in the application. This also includes Web pages for Web-enabled applications.

Definition of Logical screens/forms that browse, inquire or display existing data: Typically, these are inquiry screens that display data without permitting the user to update it. Remember to exclude literals (e.g., text and pictures) and other static elements when considering display or browse screens, particularly when considering Internet or intranet screens.

Background Processes

Provide the number of background processes

Typically, these are buttons or menu selections that activate or kick off a complete background process. An example is a menu selection that kicks off the batch nightly update process or that processes all data currently in the application. Another example would be a mathematical calculation that is performed to update a data element.

Logical Outputs

Provide the number of reports.

These represent functional transactions that leave the application, and have the main purpose of, presenting data to the user. To be counted, each output can stand by itself or could conceivably be executed independent of any other output. Typical examples are reports and batch files.

Reports: These are the typical programmed reports that users can request. They may be batch or available online. They may or may not include selection criteria. If a report uses selection criteria, count each data element in the selection criteria as part of the total number of data elements of the report. A report is considered unique if the processing logic is different. For example, detail/summary reports, reports of different media and reports with different data elements are considered unique.

In addition to programmed reports, for each Logical Table/Structure for which SQL or other generic reporting capabilities exist that users can use to perform ad hoc queries (e.g., via Business Objects on the end-user desktop), count as one output.

Interfaces

Provide a count of the number of interfaces (EXCEL, EDI, BIZTALK, etc.).

The count should include feeds either coming in to this application from other application or leaving this application.

Inputs from other applications: Typically, these are batch inputs into the applications or also may be “pushes” that update business data and are from another application. They may flow continuously or may arrive at set periods or only after operator or user intervention. If batched, remember that many logical inputs might be included in one physical file, especially for older applications. For example, the number of inputs often equates to the number of tables that the batch file can update. The idea is to reduce each batch input or dynamic file to the number of independent business transactions.

Outputs to other applications: Typically, these are data files sent periodically to other applications. They may be batched or may flow continuously or on an event-driven basis. Remember to count each logical output separately. With many older batch files, many different logical outputs may be included in a single physical batch file.

Some clients split up the Size Template data gathering amongst staff, particularly for larger applications.

Counting the repository can be accomplished by either a DBA or developer. It is important that only tables that retain business information are included in the mix. Often vendors ship the entire repository. Some organizations will create a subset of key transactions that will be executed to see which tables are touched.

Data Feeds entering or exiting an application through a number of APIs can often be counted by review of the Data Flow diagram(s) of an application.

Screens and Reports represent the bulk of the work when counting an application. There is often no short cut. As mentioned above, only count what the business user can get to. This would not include screens or reports that the user does not use, but you would include if they are available to get to them. It is also important to have super user privilege when accessing the application.

Often a Business Analyst that is technical can assist with these components, but there is complexity in screen counting and report counting. The logical model should be adhered to and not the physical. Some organizations split the work up if the application can go down separate paths and there is no intersection of screens and reports.

Lines of Code Method

If you have selected this sizing option for an application or project, provide lines of code for the following categories:

- Low-level languages (e.g., assembler)
- 3rd Generation Languages (Fortran, etc.)
- COBOL
- 4th Generation Languages (Visual Basic, SQL, etc.)
- Code Generators

Some general guidelines are provided here. This method should be used only for procedural-type languages and is not recommended for 4GL languages like JAVA, ASP, .NET, C-Sharp often used with WEB front-ends. Most procedural languages are comprised of data definitions and executable statements with comments and blank lines used for clarity and information. The Gartner model includes lines of code for data and executable code, discounting comments and blank lines.

The following general concepts can be applied to most procedural languages being counted.

- Count or estimate all lines of code by language generation level (e.g., 2GL for JCL, assembler, 3GL for COBOL Fortran and other 3GL languages).
- Blank lines and comment lines should not be included in the line-of-code count.
- Each statement or instruction is considered to be one logical line of code. For example, a line of code can be a mathematical expression, data definition statement or action statement. Also keep in mind:
 - Multiple instructions per one physical line are counted individually.
 - An instruction that spans multiple physical lines is counted as one statement.
- Programs that “paint” panels and include embedded screen layouts should have the screen design lines counted.
- Copy code should be counted once per enterprise. Most programmers use parts of other programs to reduce the amount of coding required or have set up file descriptions or commonly used routines in libraries that can be included readily in programs with a COPY statement or an INCLUDE. This type of code should be counted once.

When code has been written and stored with the specific goal of reusability in mind, that code is counted once in every application that it is used. This code can be in the form of objects or modules that are built with “generic hooks” to the outside world so that methods and functions can act on them. Typically, this is the case when utilizing base class libraries and can occur in the unique situation in which procedural code has been written with the same type of “generic hooks.”

Some clients have used a librarian product to derive a gross line-of-code count and then used a percentage decrease for those lines to be eliminated. To determine the overhead, some sample programs would need to be examined visually and hand-counted to get the estimated adjustment ratios for comments, concatenated lines and single statements on multiple lines. To determine the cross section of systems to be hand-counted, look at where the code in the

enterprise varies (e.g., at a department level, by language as well as before or after a particular chronological year) and derive the sample from those breakdowns.

Some clients have chosen to write an internal counting tool for the commonly used procedural languages within the organization employing the guidelines outlined above.

In counting lines of code for supported applications and new applications, count all lines of code according to the guidelines above. For enhancement projects, count the number of lines of code added/changed/deleted according to the guidelines above.

IT Overview Benchmark Explain Text for Applications Process Maturity

Gartner, Inc.

September 2011

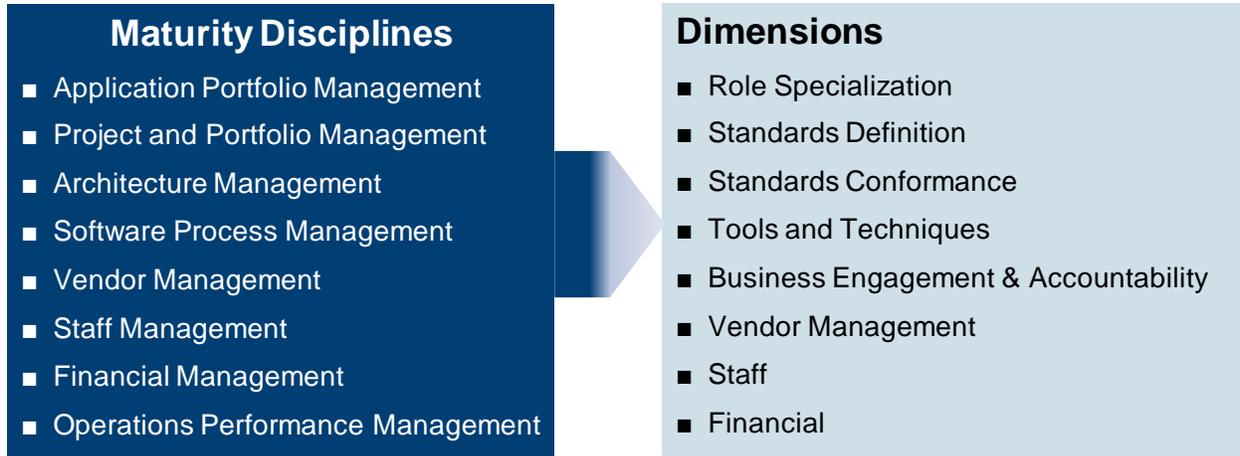
Controlled and Authorized by:
David Kish
Gartner, Inc.

Table of Contents

1.0 Overview Applications Process Maturity Assessment.....	2
2.0 Definitions of Critical Disciplines	3
3.0 Definitions of Major Activities or Items	5

1.0 Overview Applications Process Maturity Assessment

Gartner's Application Process Maturity Assessment is designed to provide a rapid assessment of the critical disciplines that are key indicators of the overall maturity of your application organization. Each discipline is scored based on the maturity level of dimensions such as role specialization, standards definition, standards conformance, use of tools and techniques, business engagement and accountability, vendor management, staff management and financial management processes.



This assessment is based on years of Gartner research focused on application management practices, with insights gleaned from thousands of engagements and interactions with application organizations around the globe. It is important that you honestly assess which statement describes your environment most closely to get the most out of the comparative assessment.

Instructions

This assessment is designed to provide an indication of maturity rather than an absolute measurement. Therefore, your estimates will be sufficient. Please provide an honest and objective answer to each question.

Please use the drop down to select an answer for each discipline. If a discipline is out of scope, please indicate by selecting Not Applicable. If a discipline is in scope but you are unsure of an answer, please select Don't Know.

2.0 Definitions of Critical Disciplines

Application Portfolio Management (APM)

This is the set of activities that documents and drives how an organization measures and responds to the business value, cost, performance and risk of its portfolio of application assets.

Project Portfolio Management (PPM)

This is the set of activities that drive the effective allocation of scarce resources (such as capital and people) on a project-by-project basis. This category includes the disciplines of application project and program management, coordination with project management offices (PMOs), and with the investment and prioritization processes for projects.

Architecture Management

This is the set of activities that brings together the three primary design viewpoints of application architecture: business, information and technology.

Software Process Management

An organizational set of processes, each targeted toward a specific work type. The processes fall under the basic form of a software process architecture, which includes a framework level (what must be done), a method level (how software is created or how an integration effort is done), and a template or quality-assurance level (how to create a deliverable).

Vendor Management

Practices include understanding the vendor's strategy; managing vendor relationships or engagements; sourcing, developing and maintaining commercial processes; and optimizing vendor performance management. Vendors may include external service providers (ESPs), outsourcing providers, and vendors of business applications, content and collaboration tools, and development tools.

Staff Management

People, whether direct employees or other ESPs, comprise the single-most-significant asset in an application organization. This assessment focuses on the processes that an application leader can control or influence, including role and competency definition, skills and knowledge, training and development, succession and backfill planning, sourcing, retention, productivity and performance management, and culture.

Financial Management

Financial Analysis and Budgeting processes include operating and capital expense estimation and forecasting, variance analysis, scenario planning, business case tracking and processes to handle unexpected events.

Operations Performance Management

Processes include the collaboration between the application organization and infrastructure & operations organization (I&O) in managing incidents, problems, changes, configurations, releases, disaster recovery and capacity to ensure the applications in production meet service-level requirements such as stability and ease of use.

3.0 Definitions of Major Activities or Items

Project Management Office (PMO) or comparable

PMO's responsibilities include designing and facilitating the PPM process and creating a standard set of project management practices to ensure consistency in managing software projects. This may include purchasing a methodology or developing one in-house. Typically, the PMO will focus on a repeatable process, which must be defined in a "just enough" fashion.

Data modeling and database design

Includes analysis and development of complex logical data models, database design and relational data definitions that depict the flow of data within and between technology systems and business functions/operations.

Application analysis and design

Tasks include determining, interpreting, translating and reconciling diverse requirements into comprehensive and, often, idealized technology directions, approaches or solutions that resolve the full range of needs. On the all-application level, these solutions may be represented as technical direction. On an individual application level, the solutions are represented as conceptual designs or technical process models.

Software engineering process/methodology

These actions include the key tasks for developing the application or implementing acquired software or services; configuring the tools, processing rules and metadata; and integrating the solution with all required processes.

Change/Configuration/Release management

New or modified applications must be deployed into the operational environments in coordination with colleagues in infrastructure and operations (for example, change control, configuration management, release management, and so on).

Testing

Includes testing and exercising the solution to ensure it performs within standards and requirements.

Quality Assurance

Includes a set of defect-removal activities that occur during the full life cycle, along with a set of phase gates that ensure that these processes are followed. — Defect-removal activities may include testing methods and test measures, such as status (for example, test cases, number executed, number failed and number to test), quality level (defect arrival rates charted) and conformance to quality requirements (mean time to failure).

Production Turnover

Deployment tasks include communication (what is changing and how, description of new functionality, and impact of change), training, planning for change and overseeing the change (responding to issues; making decisions on proceeding, stopping or revising and so on).

Sourcing (Applications and/or Services)

The application function relies on vendors that supply a range of products and services; sourcing management practices may include understanding the vendor's strategy, managing vendor relationships or engagements, optimizing vendor performance management and effectively utilizing the vendor ecosystem for innovation.

Business process architecture and models

That part of the enterprise architecture process that describes — through a set of requirements, principles and models — the future state, current state and guidance necessary to flexibly evolve and optimize business dimensions (people, process, financial resources and organization) to achieve effective enterprise change.