



Application Portfolio Management

A Dual Perspective

Application Portfolio Management Defined



Application Portfolio Management (APM), a formal IT Management discipline, was first introduced in the early 2000-s to denote emerging management practices aimed at an organization's application systems portfolio.

Although the notion of portfolio comes from the financial services world, during Y2K IT shops were – perhaps for the first time - forced to look at and do something universal to *all of their applications as a whole*, regardless of platform, technology, size, age, or even value to the business. For many organizations, this was the first time that a complete inventory of applications was developed, and the first time that applications were assessed using formal evaluation criteria applicable to all. With that process, an important realization has also cemented itself: the collection of application systems represents a major investment for the organization, it is a major asset that is critical to the day-to-day operation of the business and perhaps most importantly that it costs significant financial resources just to maintain and support this asset. It is widely reported and accepted that most organizations spend over 70% of their information technology budget on maintenance and support of existing applications.

To-date there have been a variety of definitions and interpretations put forward as to what exactly APM is or, perhaps more to the point, what it *should be*. Analysts published research papers, vendors inundated us with white papers, and journalists tried to make sense of it all in magazine articles.

An early Forrester perspective had APM defined as follows:

*"APM is a set of tools and processes that builds an application knowledge base that serves as the sole source of application truth across the enterprise and as a vehicle to collect key application metrics to help govern and manage application activity"*¹.

More recently, Gartner defined APM as:

*"APM 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."*²

From the cacophony of information, a few recurring themes persist:

- Management of the application portfolio in its totality and as a whole is no longer an optional exercise; it has become a necessity.
- APM provides the frame-work to capture, assess and evaluate applications and to develop future plans and directions.
- APM is the foundation and pre-requisite to effective application modernization efforts.

¹ Trends 2005: Application Portfolio Management (Forrester, October 2004)

² Maturity Assessment for Application Organizations: Application Portfolio Management (Gartner, July 2009)

- APM delivers an application knowledge-base - built directly from parsed source code - that offers visibility into the structure of the application.
- APM delivers fact-based application metrics that support IT management decisions.
- APM increases productivity (percentages vary from 10% to 30%) and hence helps reduce maintenance and support budgets.
- APM needs to extend to the entire application portfolio and must be adopted on an ongoing basis (i.e. not just a one-time exercise).

As organizations started to look into APM, it became clear that there was not one singular right way of adoption. There were not only differing interpretations of the popular definitions, differing emphasis on the main themes, but inevitably significant logistical issues. A large bank, for example, with thousands of applications across all existing technology platforms faces a different problem than a smaller IT shop with perhaps 3-5 major applications and a few dozen small satellite applications.

Top-down APM

Many shops with a large portfolio elect to start - simply out of logistical necessity - with a top-down, high level approach. This approach entails the creation of an Application Inventory and the gathering of a variety of attributes about each and every application. The process of collecting the information is largely manual, and the supporting tool-set is often home-grown, in many cases a series of cleverly constructed Excel spreadsheets. Critical information is captured and categorized: business functionality / alignment, technology profile, size (lines of code estimate), application owner, annual budget / maintenance spend, lifespan, etc. The data is then sorted, categorized, sliced in a number of ways to feed into assessment and planning exercises. Many organizations believe that once they enumerated all their applications and captured some key characteristics that are kept up to date over time and which are used to feed planning activities – they have adopted the APM discipline. And to some extent, they have. They are certainly much further ahead than the shop with no such information.

The advantage of a top-down approach to APM is that, within a relatively short time, the entire application portfolio can be enumerated (including package applications) and assessment and planning can commence. In addition, tool-support is fairly extensive, especially when one considers some of the Enterprise Architecture (EA) tools that offer functionality that crosses into the APM domain.

There are shortcomings, however, of a top-down *only* approach:

- it is fundamentally a manual approach; information gathering is tedious, some aspects rely on interviews, data can get out of date before the inventory is complete; some of the information is often based on estimates or best guesses;
- inter-application relationships (e.g. files shared by two applications, dependencies, etc.) are not captured in an automated fashion, rather, they are based on one-time, manual scans or interviews with application domain experts;

- there is no "insight" into the applications, their structure, intra-application relationships remain invisible as they are beyond the scope;
- maintenance is cumbersome, the information becomes soon outdated and the whole process needs to be repeated in 18-24 months.

Bottom-up APM

The bottom-up approach is fundamentally based on parsed source code. The Application Inventory is built from the underlying source code by parsing out every component of every application and all their intra- and inter-application relationships. Once the repository is built additional information (that cannot be captured from source code or other electronic footprint) may have to be added in order to support assessment and planning initiatives. The repository needs to be kept fresh always reflecting the current production state.

This approach obviously requires automated tool support and there are a variety of tools available with varying degrees of coverage and functionality.

The advantages of the bottom-up approach are significant:

- information is based on the ultimate system of record, that is production source;
- application metrics are directly derived from the source code, hence they provide a true and accurate picture of size, complexity and quality;
- the repository serves as a central, institutional knowledgebase (a form of automated systems documentation) that can support everyday tasks.

There are, however, some challenges with a bottom-up *only* approach:

- for very large organizations with potentially hundreds, or perhaps thousands, of applications the creation and ongoing maintenance of a single repository may well be a logistical nightmare and is nigh impossible;
- if multiple repositories are created (for logistical and organizational reasons), capturing and maintaining inter-repository relationships becomes a difficult task;
- package applications cannot be included (unless source code is made available);
- parsing every small, fringe application will likely provide little or no tangible benefit.

A dual approach

In most organizations the application portfolio consists of dozens, hundreds or perhaps even thousands of applications. Applications vary widely in terms of size, technology, business value, volatility and life span. Some are in-house built, some are vendor supplied (some with source code, some without), some are developed by third party and brought in-house for maintenance and support, some are outsourced. Not every application is the same. There is usually a core systems portfolio - maybe 15% to 20% of all applications – that is central and critical to the day-to-day operation of the business and which attracts up to 70% to 80% of the annual maintenance and support budget.

The bigger and more heterogeneous the portfolio, the more likely it is that a combination of the top-down and the bottom-up approaches will be the most effective in terms of logistics and costs.

First step is the creation of an all-encompassing Application Inventory. In most organizations this task is undertaken by the Enterprise Architecture group. This group is typically tasked with developing and maintaining the organization's systems portfolio blueprint; to determine the as-is picture, the desired target state and then to chart a path from here to there. The pre-requisite of such a blue-print is the documented understanding of what applications are in place. The size and complexity of this task is obviously a function of the size and complexity of the portfolio. In large shops with possibly thousands of applications, this task is often enormous and takes months if not years to complete. Many groups struggle with seemingly simple questions:

- What is an application?
- Do we need to capture all? Do we need to include end-user developed / sourced applications as well?
- What key attributes need to be captured? What and where are the sources of those attributes?
- What tools are available?
- How do we ensure currency of information?

Once the Application Inventory is complete, applications that are core, critical and merit or require further analysis and a more detailed understanding are identified. In most IT shops, 80% of the maintenance and support budget is spent on 15% to 20% of the applications in the portfolio; relevance and priorities lie where most of the money is spent.

In the second phase of fully embracing APM, the core systems portfolio is subjected to the bottom up approach. A repository of application knowledge and fact-based metrics is built from parsed source code. This is a task that requires a special purpose APM tool.

The best of breed tools should exhibit some key characteristics:

- source code should always be parsed, not scanned; that is gleaning semantic interpretation based on the specific language grammar, instead of looking for pre-defined text patterns only;

- a well-balanced tool should provide language / platform coverage for all technologies within the target scope, and each should be parsed with equal depth;
- parsers should be flexible (i.e. customizable) to handle non-standard use of language constructs, custom-written exits and add-ons. This is a fundamental need for most older, legacy systems;
- system connections across technologies (e.g. Java to MQ to CICS to Cobol) should be seamless and complete;
- application metrics should include size, complexity and quality indicators derived directly from the parsing and a dashboard facility should present the information in report and graphics format as well;
- typical functionality at minimum should include inventory, component maps, visualization (i.e. dynamically generated diagrams), a robust search facility, impact analysis, reporting, and a variety of housekeeping functions.

The Information Balance approach

At Information Balance we strongly believe in a dual approach to APM and hence we define it as follows:

Application Portfolio Management is a management discipline that provides the frame-work for capturing the necessary input to strategic systems planning and application modernization. At least for the core systems portfolio, relevant information and key metrics about applications should be obtained by the direct parsing of production source code.

In summary:

- Organizations should have a complete inventory of all applications based on a top-down approach.
- For application systems characterized by custom code, high rate of change and high maintenance costs, detailed bottom-up application portfolio management (discovery) is a must to provide productivity to developers and provide accurate planning metrics for management.
- For custom applications that are changed infrequently, consider a one-time detailed bottom-up application portfolio assessment in order to capture the detailed knowledge of the application before it walks out the door.
- For packaged software applications, detailed bottom-up application portfolio management is not practical due to the lack of source code for the assessment. However, detailed assessment of the interfaces between custom and package software is essential for effective management of high-cost, high usage vendor packages such as SAP.

About Information Balance, Inc.



Information Balance, Inc. develops software solutions to support large scale application development and maintenance practices across mainstream technology platforms. Its flagship product **Application Road Map (ARM)** is a unique Application Portfolio Management (APM) solution. The company's clients include leading financial services, insurance, telecommunications and retail organizations across North America.

For more information, please call 416-962-5235, e-mail infobal@infobal.com, or visit our website at www.infobal.com.