HP Application Life Cycle Management Best Practices Series

for ALM Practitioners

Project Topology Best Practices

Legal Notices

Warranty

The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

Restricted Rights Legend

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notices

© Copyright 2011 Hewlett-Packard Development Company, L.P.

Trademark Notices

Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation.

Oracle® is a registered U.S. trademark of Oracle Corporation, Redwood City, California.

Documentation Updates

The title page of this document contains the following identifying information:

- Software Version number, which indicates the software version.
  - The number before the period identifies the major release number.
  - The first number after the period identifies the minor release number.
  - The second number after the period represents the minor-minor release number.
- Document Release Date, which changes each time the document is updated.
- Software Release Date, which indicates the release date of this version of the software.

To check for recent updates or to verify that you are using the most recent edition, visit the following URL:

http://h20230.www2.hp.com/selfsolve/manuals

This site requires that you register for an HP Passport and sign-in. To register for an HP Passport ID, go to:

http://h20229.www2.hp.com/passport-registration.html

Or click the New users - please register link on the HP Passport login page.

You will also receive updated or new editions if you subscribe to the appropriate product support service. Contact your HP sales representative for details.

Support

You can visit the HP Software support web site at:

www.hp.com/go/hpsoftwaresupport

This web site provides contact information and details about the products, services, and support that HP Software offers.

HP Software online software support provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by using the support site to:

— Search for knowledge documents of interest
— Submit and track support cases and enhancement requests
— Download software patches
— Manage support contracts
— Look up HP support contacts
— Review information about available services
— Enter into discussions with other software customers
— Research and register for software training

Most of the support areas require that you register as an HP Passport user and sign in. Many also require an active support contract. To find more information about support access levels, go to the following URL:

http://h20230.www2.hp.com/new_access_levels.jsp

To register for an HP Passport ID, go to the following URL:

http://h20229.www2.hp.com/passport-registration.html

Contents

Welcome To This Guide

Welcome to the HP ALM Project Topology Best Practices guide.

This guide provides concepts, guidelines, and practical examples for the best approaches on structuring HP ALM projects in various organizations.

About Project Topology

Today, more than ever, the quality of your applications directly impacts your bottom-line business results. This is the reason IT departments of most enterprises invest heavily in products, people, and processes that can maximize application quality. Application software plays a dominant role in today's business, regardless of the vertical market or core competency. Every organization must be able to guarantee high quality, working software to properly position and deliver its products to the market. Now, more than ever, software is a critical component for winning the competition.

HP is the leading provider of the market-defining solutions Quality Center and Performance Center; both are focused on the automation and performance testing arenas. These two products forever changed the landscape of what was expected in the area of software delivery. Now, with HP ALM 11, HP introduces one complete ALM solution that covers all the phases that software travels through during its existence. Embracing ALM practices and solutions enables software teams to meet the high expectations and demands of the business. Stakeholders in the application lifecycle, from the IT side as well as the business side, such as business analysts, developers, or testing professionals, have to communicate, collaborate, and connect to ensure that every aspect of the software will meet the rigorous demands of the business.

HP ALM suite can serve various companies to achieve their specific needs based on industry segment, company focus and processes, amount of applications and their type, and so forth. It is therefore an important aspect of each implementation to carefully plan the topology and internal structure of ALM projects so they best fit the requirements of a given organization. Proper project topology has a profound effect on the reporting, user management, administrative tasks such as backup/restore, etc.

The purpose of this document is to assist HP ALM customers in assessing their current development and testing practices and successfully defining ALM project creation methodology. All aspects of this process have been researched using best practice data and expertise from various sources including HP's operating system administrators, HP's professional services organization, technical documentation, books from industry experts, and the personal experience of many customer testing organizations. These guidelines will help reduce the initial creation time and achieve maximum value in operating the HP ALM.

Audience

This guide is intended for:

- Vice Presidents of Applications

- Business Analysts

- Testing CoE Managers

- Testing Automation Engineers

- Development Managers

Prerequisites

To use this book, you should have a working knowledge of the major phases of the Software Development Life Cycle (SDLC). You should also be familiar with the business processes in the actual IT organizations.

Operational knowledge and administrative privileges of HP ALM are essential for implementing these best practices.

Note: some of the features discussed in this document are available only in HP Application Lifecycle Management.

Structure

This guide is organized as follows:

- Introduction to Project Topology
- Planning Project Topology
- Project Topology Examples
- Other Aspects
- Conclusions

Feedback

If you have questions, comments, or valuable best practice information you want to share, send a message to the following email address:

alm_cust_feedback@hp.com

1 Introduction to Project Topology

Importance of starting right

Most users do not work directly with databases, networks, servers, or storage systems; they work with applications. Yet, traditional IT management is oriented toward managing the individual components that applications rely on. When application problems arise—and they always do— IT must search through silos without a clear idea of what the root cause is or where it might reside. ALM comprises tools and processes for keeping IT professionals informed about the quality of critical applications, and enabling IT to address the problems. A key ALM objective is to bring IT professionals closer to a "single pane of glass"—that is, one complete view of all relevant information about how an application is performing. Achieving such a goal with ALM could allow IT to be more productive, effective, and responsive to users' problems. ALM is also about providing IT with an "end-to-end" understanding of the application quality, performance, and availability. This means starting with the user's experience and moving all the way through the components that make up a large application. This view helps IT get past the conundrum in which individual systems seem to be working properly but users are still dissatisfied with their experience.

HP ALM, the first ALM-focused solution on the market, is the way to break those silos in order to deliver best possible quality.

You start from the implementation plan that includes careful allocation of the resources according to the company business processes and divisions.

Implementation of any testing tool or process requires much thought and planning, and no matter how good the product is, it will quickly become useless unless planning starts during the sales process and extends well beyond the software delivery date. Therefore, whether your company is an existing HP client or a prospective customer, it makes sense to assess your current processes and needs according to the industry best practices. This paper should help you with defining the methodology so the very first steps of the project creation are taken in the right direction. You may also consider working closely with HP and its partners to make sure your requirements, questions, and concerns are understood and addressed at every stage.

Effective implementation of the application life cycle requires a systematic process. When your team manages multiple projects, each with its own cycles and priorities, it's easy to lose focus. Almost no organization can dedicate a testing team to one development project. At best, a testing team is part of the development group for a particular product. Additionally, you need to validate multiple functional areas, builds and baselines, different platforms and environments, and a vast number of integrations. To track multiple test cases, your testers need a process that allows them to manage multiple projects and clearly defines the objectives of each one.

"Do it right or do it over" is the compelling slogan to convince company management to allocate time to properly plan HP ALM topology based on projected growth rate.

Variety of approaches

Now, more than ever, enterprise success depends on applications that achieve their desired business outcomes. If your IT organization lacks the ability to consistently deliver high-quality applications, your business is at risk.

No company is the same and therefore business processes vary quite significantly. This is reflected in the way application development and testing is performed. Prior to rolling out HP ALM, it is required that you conduct comprehensive analysis of the quality ecosystem in place at your company. You should analyze your environment, interview your personnel, assess current process maturity and product adaption, and review supporting documentation to define the approach that will suit your business needs best.

General Concepts

As you assess your company maturity level and amount of actual development projects being carried, one important decision is to decide *what your ALM project* (database with resource repository) *represents*. This decision will have a profound effect on the way the project is managed, how requirements are written, and the approach that is taken to test the application.

While the definition of the term "*ALM project*" is different from the *Project Management Institute's* definition of "*project*", it features some very similar attributes such as definitive start and end, scope, milestones, etc.

HP **recommends** using multiple ALM projects in most of the cases as it brings clear advantages to the implementation and makes ALM administration much easier.

As there quite a few factors that may affect your choices, HP **advises** to follow best practices for your *industry* implementation detailed in the following chapter.

Types of Projects

Generally speaking, the majority of implementations fall into these categories:

- Repository Projects

  These are long living projects that maintain your IT application requirements and tests that are shared in satellite projects. There might be one or more repository projects depending on the amount of assets stored. Typical usage includes compliance requirements, general development guidance such as security and user interface requisites, etc.

- Application Projects

  These are also long-living projects for the ongoing releases and patches of a single application. Both IT organizations and ISVs use this implementation scenario.

- Model Projects

  Model project is very much like Application project but focuses on physical entities such as hardware devices, car components, cellular phones, etc. This kind of project is frequently used for branching of the main product line.

- Platform Projects

  These are long-living and frequently used projects that contain some common set of entities based on some sort of "platform" such as an operating system, a development framework, hardware device components, and so on.

- Initiative Projects

  This is the example of ad-hoc projects for large initiatives that span more than one application. This project can be archived after initiative ends and changes are usually synchronized into common repositories.

- Company Structure Projects

If a company is structured around various lines of business which do not often intersect in their activities, then ALM project topology should naturally reflect this division. On the other hand, there are quite a few examples of centralized organizations inside large companies that are aimed at addressing specific needs on a company-wide basis. In both cases, each such organization should have its own project to maintain end-to-end scenarios.

These are the possible types of projects that are discussed in detail along with examples later in the document.

2 Planning Project Topology

This chapter describes the basic steps for assessing the status of the development lifecycle and the building blocks in project topology decision making.

Before You Begin

There are some conceptual decisions that have to be made prior to any implementation of HP ALM.

Multiple Projects vs. Single Project

Sometimes the company decides to maintain all activities of all teams inside one rather large project. This type of implementation may indicate the desire to keep everything in one place and separate various types of entities by granting certain privileges to certain users. Alternatively, such projects may be dedicated to one of the various ALM activity types, for example, a defects-only project or requirements-only project for all kinds of applications.

While there are some companies that choose to take this path, in most cases, HP would **not advise** implementing this scenario. There are quite a few reasons:

- Contradiction with project management paradigm

  Projects should have a clear start and end – if you put an entire portfolio of application developments into a single project, you may lose the insight and focus.

- Access management burden

  It is nearly impossible to assign proper privileges to each and every requirement, test folder, or defect, which creates a situation where everyone can access everything.

- Too much dependency

  Having all data in one project may affect maintenance tasks such as backup/restore and upgrade path – performing these routine procedures

becomes nearly impossible due to interdependency and the involvement of multiple teams.

- Data filtering is hard

  Managing permissions for requirements and tests is hard, and filtering data in various lists cannot be easily accomplished.

- Difficult to get current status per project

  As everything is bundled together, reporting on overall quality status of the specific project is troublesome.

- Performance issues

  As usage grows and more development projects get on HP ALM to manage their quality needs, the amount of data kept in a single database may cause performance degradation.

In HP's opinion, using multiple ALM projects brings some clear **advantages** to the usage model:

- Better structure

  HP recommends examining the granularity of the project to determine the optimal size for the content.

- Better user management

  When dealing with one project at a time, it is much easier to define the necessary privileges for folders and certain entities such as requirements, test sets, defects, and the like.

- Better administration

  If you consolidate data assets between multiple projects, you are free to schedule regular maintenance tasks such as backup/restore and apply project-based timetables to ALM upgrades.

- Better data filtering

  Inside the project that is dedicated to one purpose – be it application, model, or line of business – only relevant informational resources are kept.

- Better compliance

  Using a multiple projects model helps achieve the compliance level required in many industries.

- Better performance

Since each project is smaller, both database performance and repository performance are better. Dividing a big database into smaller, more manageable parts also allows using advanced I/O features that modern RDBMSs support, thus improving the response time. File system based repositories can also benefit from increased access time if they are spread between multiple physical storage drives.

HP ALM 11 is designed from a cross project reporting and metrics point of view and therefore the request for an all-inclusive mega project is irrelevant. HP ALM allows great flexibility in mapping software development processes – hence it is imperative to analyze them prior to implementing the tool.

Industry Effect

It is well known that while no company is the same in terms of business processes, requirements, and reporting, there are some strong similarities in the way business is done in certain industries. Financial, insurance, automotive, pharmaceutical, software vendor – each of these sectors have common patterns of behavior, common usage models, and common testing principles.

For example, in the automotive industry, the defining factor in choosing a new project would be a model year. The next year model may be based on the same chassis and most of the components are also the same, but there are always modifications that alter car appearance or behavior. Therefore, the start of new model development would be accompanied by the creation of the new, year-based project.

When cellular providers test mobile phones from different vendors, they usually have a separate project for each major phone model. This way, all the requirements, tests, and defects pertaining to the specific handset are stored together as long as the phone is being sold by the provider.

Independent software vendors (ISVs) like HP Software, base their development lifecycle on the major version. Hence, the start of the new version of the product spawns a new project. This new project contains a majority of the requirement definitions and test sets, while test execution and defect data is not copied. This approach allows starting from the last point where the development stopped in the previous version and concentrate on the new features. The older project is generally backed up and is used for patch management.

More specific examples of the topology implementation per industry appear later in this document.

What to consider

HP ALM implementation may be less successful if the current situation is not surveyed, evaluated, and, if needed, changed. One of the most common misconceptions when purchasing a test management or even a complete ALM tool is that simple installation and minimal configuration of the tool will facilitate successful use and implementation.

Common Factors

There are several factors that may influence the decision making process of the HP ALM planners. The following actions are essential to define a strategy towards project creation:

— Assess organizational structure

Start by evaluating the team or group charged with the requirements' design, code development, and quality testing. For small to midsize companies, this may be a single centralized team. In larger organizations there may be separate teams. If each of the management types is handled by a separate team (with or without overlap) then aligning project topology with current organizational structure is critical in implementing HP ALM.

— Define reporting requirements

One of the most important factors is what kind of reporting, including status, is expected from the project team. If the development team is expected to report individually, then having all assets in one place may greatly simplify the task.

— Calculate projected growth

Common factors to be taken into consideration include when the software is going to be used and if the team or organization is going to grow in the foreseeable future. If the project is expected to grow both in terms of assets to be handled, such as requirements, tests, and defects, and in terms of number of users, then it would probably be a good idea to consolidate all its related entities into a separate project, even if it is currently not fully justified.

— Evaluate access needs

Granting all users access to a single ALM project may cause inadvertent security violations. A potential issue with unifying several development and/or testing projects into one ALM project is

visibility to some of the test data, like Requirements or Defects. Often, the actual testers are subcontractors from another company and should only see the data relevant for them. If development projects are split between multiple ALM projects, you must use ALM's tools for managing user permissions, while at the same time raising maintenance task levels – you need to manage more than one project. It really comes down to what you feel makes more sense business-wise.

— How much customization expected

The ability to place a workflow in each module of HP ALM is one of its most powerful features. HP ALM is used to enable and enforce your process, but you must define that process first. Once defined, you need to determine what level of workflow customization is required. Workflows can be created through wizards, or by someone who is well versed in code scripting. Decide which customizations are needed to allow the tool to adapt to your process. Obviously, as each ALM project represents a unique set of development and testing activities, there might be specific customization needs for each project. However, it is highly advisable to establish common naming conventions, a common list of values for various entities, and possibly even the same set of user-defined fields and code scripts, so that all projects have a consistent common ground. In the future, this will allow cross project reporting and sharing.

Domain

Another important factor in planning the ALM project topology is the grouping of the projects by domain. Companies do not bundle together different projects randomly – there is a business purpose behind the decision.

If we expand our examples from the Industry Effect section, then the automotive manufacturer would most probably group the projects by model year with projects representing a car brand.

For example, an imaginary *HP Motor Company* may have a *2011* domain for its *HP Runner* vans brand and *HP Site* trucks brand.
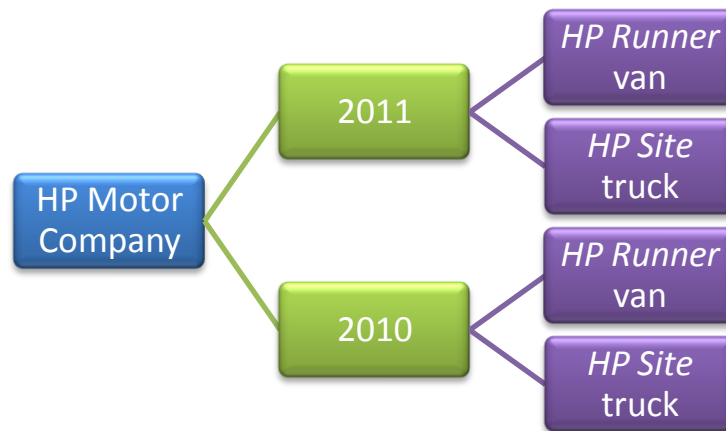
The previous year's models, grouped under the *2010* domain, also have the *HP Runner* vans brand and *HP Site* trucks brand, but their requirements, test cases, and defects come from the previous year.

In the world of cellular providers, domains are based on the equipment supplier so that the Supplier N domain contains projects per its current models. ISV would pick the product name to be the name of the domain where all recent versions and new version would be maintained.

An additional benefit of aligning ALM projects with the functional area or domain is the ability to assign an administrator for certain domains, so that he/she could apply expertise in this functional area and drive the underlying projects.

Technically speaking, if you divided projects over multiple domains, you could better manage storage location of the associated file system repositories, thus allowing easier maintenance and increased performance.

Development Approach

One of the factors to consider during evaluation and planning is the development approach. To respond to the demands of a hyper-competitive marketplace, IT departments today are tasked with increasingly diverse responsibilities. They must support global, 24/7 operations and integrated supply chains while quickly delivering applications to the market.

Most existing applications were developed using the classic, so called *waterfall,* approach. In the waterfall approach, each team developed separate, unrelated features and delivered them at the end of the process for final integration. Their new versions and patches would probably continue this way.

With proliferation of the modern, *agile* approach, it has become custom to see new initiatives or autonomous modules moving in this direction. Sometimes companies start on the smaller scale and transition pilot teams to work in agile fashion.

Agile teams want liberty in applying agile manifesto principles to the collection of epics, work items, and tasks they usually maintain.

Whatever the reason, HP ALM provides the capabilities to support both approaches. If a certain development team wants to continue working in waterfall, you just maintain their dedicated project and they create artifacts as they go.

As both methodologies are supported, it is recommended that organizations create some common milestones for both approaches. This way reporting can be done across projects regardless of the methodology chosen.

Then there are quite a few teams that choose a *hybrid* approach – some steps resemble waterfall, some steps are clearly from the agile world. Create a separate HP ALM project and let them use the most suitable arrangement.

HP ALM provides tools to achieve project objectives by including entity sharing between projects being developed *simultaneously*, regardless of what approach is taken, thus allowing quick synchronization of changes and traceability.

Compliance

Since many industries are heavily regulated and must pass a variety of compliance-based tests, this is yet another important parameter to consider especially when these regulations are updated regularly.

Many organizations—especially those in the finance, healthcare, and government sectors—are required to comply with specific government regulations such as HIPAA, Sarbanes-Oxley, US Section 508, and many others. It is therefore obligatory for their IT divisions to demonstrate adherence to the highest level of regulatory compliance.

Since compliance requirements usually remain without change until their new version arrives, it may be a good idea to store them in a separate ALM project which would serve as a repository, and "plug" into the corresponding

development project where execution data resides. When the repository is modified, this change is propagated into the corresponding development project. This way your company can comply with the regulators - keeping the compliance data untouched while generating the required audits accordingly.

Initiatives

Mature IT organizations often adopt an approach called business initiative. An initiative is a set of applications that are developed to support a common business goal. An example of an initiative is "Improve printing" that may include replacement of network printing software, installation of a new generation of multifunctional printers, development of printouts adjusted to these new MFP devices, etc. To support this initiative, several new applications may need to be developed and a large number of the existing software applications may need to be modified.

The scale of these developments and modifications could not be managed if every application team worked alone in their respective silos. Instead, all requirement management, development, testing, deployment and change management processes need to be coordinated across multiple applications. With each application team introducing their own requirements, tests, components, and defects, the key to a successful release of an IT business initiative is to enable visibility, coordination, and collaboration.

In most of the cases, organizations choose to maintain initiatives in separate ALM projects, thus achieving higher levels of consistency.

Star-based Project Topology

As there are quite a few parameters to consider when designing ALM project topology, one specific type of project layout may be a best fit for complex enterprise development projects.

HP **recommends** using star-based configuration where there is a central repository project, usually containing definitions, and satellite implementation projects containing execution data.

One of the common usage scenarios primarily aims to provide reusable compliance requirements and generic test case definitions through sharing in companies that need to address regulatory rules that change over time, such as Sarbanes-Oxley, HIPAA, COBIT, and many others. When a compliance requirement needs to be updated within the repository itself, the Compliance Officer (or someone fulfilling that role such as ALM Administrator) makes the modifications within the repository project. The changes are applied to the implementation project using synchronization - a changed icon and alerts let the project owners know they should resynchronize the requirements.

Another typical scenario is to create a project with reusable requirements and test cases for one-time usage. It is often used as a repository for common requirements and artifacts that span multiple projects, for example, common GUI interface test sets, common database or security requirements, performance requirements, and so on. This type of project assets gets imported from the repository project into the implementation projects.

See the *HP Entities Sharing Best Practices* book in this series for more practical details on how to most effectively use sharing between repository and implementation projects.

The common ground for these examples is that global testing assets, such as requirements and test cases, are designed in one centralized repository project and are transferred into satellite implementation projects that utilize them. The actual test execution data is saved during implementation only, thus making global assets – usually regulatory-bound entities – unaltered until the next wave of modifications.

There are many benefits to this type of project topology:

— Better access management

You may limit access to sensitive global assets in the repository while allowing necessary changes in the satellite projects.

— Better approval process

If you keep information that requires approval in the repository, your company's review and approval process may concentrate on the central repository only. This includes limiting the number of users in the repository, special permissions to the approvers, visibility of approved items, easier access to the assets by management (they need to log into one central project instead of many), easier notification to the stakeholders, etc.

— Better reporting

Once you know which projects use a certain requirement or test case, it is much easier to report the requirement coverage status of the implementation projects. In some cases, such as in regulated environments, it may be a legal necessity.

— Better governance

This approach enables a lean and efficient set of projects that do not contain unnecessary data. Each project contains the data relevant to its purpose and repository serves as the common definition ground and "point of contact" for all satellite implementation projects.

Many large enterprises build Testing Centers of Excellence (CoE) that unequivocally choose this topology model for their operation because of the stated benefits. Numerous analyst reports attribute the importance of CoE to the significantly enhanced ability of an organization to meet or exceed the goals that the center supports. When governance, a support structure, guidance, measurements, and shared learning exist across an organization, success is far more likely. A successful organization can focus on specific project goals. With proper project topology of the main ALM tool in place, it is much more likely that your company excels in delivering better outcomes.

3 Project Topology Examples

In the previous chapters we emphasized the importance of project topology and its profound effect on the HP ALM position inside the company. We also discussed the main factors that may affect the project topology decision and a recommended star-based topology that covers multiple implementation scenarios.

The majority of ALM deployments occur in IT organizations. Since no two IT shops are identical, the decision on how to build project topology depends largely on considerations such as the company culture, business processes in place, and previous tool limitations (if any). Let's see examples of project structuring in detail.

Application

This is the common case. Each project represents one application, with all of its requirements, business models, KPIs, test sets, test resources, defects, and reports. This is the natural way to map development activities, to manage relationships between business analysts, developers, and testers. For example, Billing, CRM, and Portal are widespread names of projects and these are dedicated to their namesake applications.

In most of the cases, the focus is on an application which has consecutive releases. Therefore many customers perform a backup of the previous release's project and restore it under a new name for the next major release. Hence we may find project *Billing 5.0* that is converted into *Billing 6.0* when all development and testing activities of the former are finished and the latter becomes the current project. In this case, the older version enters a maintenance state, and the new version moves through development milestones towards a *release*.

There are some clear gains in employing this topology type:

— Good visibility

— Easy control of all related activities and processes

— Simplified access management

— Effortless restore from backup when needed

Model

Hardware manufacturers have processes similar to those in other areas – after all the development and testing concepts are the same and objectives are equivalent. HP ALM is implemented in various types of companies. Sometimes a project represents a model (physical entity) of the product, such as a cell phone, printer, or TV set. In this case too, a new project is created when a new model development is started.

Platform

If development and support activities are organized around some major "platform", then structuring the projects in a similar way is natural. Examples of platform projects include splitting activities:

— by computer operating system

  Windows servers vs. Windows desktops vs. Linux vs. HP/UX

— by smartphone operating system

  iOS vs. Android vs. Windows Phone

— by infrastructure framework

  Hadoop vs. Hybernate vs. GWT

— by common device part

  Laptops vendor with Intel and AMD processors

— by automobile chassis or engine

and so on.

As long as this topology maps actual business processes, its advantages are self-evident:

— Common requirements

— Component-based testing helps in removing redundancy and improves progress tracking

— Entities sharing is done within one project

Initiative

This is also a common usage pattern found in many IT organizations. For example, suppose there is a plan to add support for a new version of an internet browser. As the requirements are mostly specifics to this browser, this initiative is translated into the series of changes of existing applications.

To develop this new browser support, a new ALM project is opened to cover all aspects of planning, coding, testing, and bug tracking. Integration of existing components and/or systems usually falls into this category too.

The main benefit of the approach is the focus on the activities directly related to the initiative, thus making it easier to report progress.

LOB

In a world of huge companies and endless mergers and acquisitions, one Line of Business (LOB) may be quite separate from another, or it may just maintain independent sets of data, thus making the case for an implementation separated by lines of business.

A frequent scenario is when each LOB maintains one project where it keeps all its applications or compliance information. In bigger organizations, such grouping may go one level up to the domain – each domain representing one LOB with its specific projects.

While it may sound like support for the silos within the company, the business sense is the main factor to be considered – if different LOB's matters do not intersect, then having them separate would be a proper project topology decision.

Internal Service

In some companies, there might be an internal organization which provides services to all users regardless of their LOB association – as opposed to the LOB only activities (see above). This organization may work with many different applications which are being tested, improved, and deployed but are grouped and supported together to provide an holistic customer experience. An example of a department within an organization may be *Messaging* which

covers the whole spectrum of messaging applications – from Outlook to Outlook Communicator to Exchange Server to centralized fax service to Short Message Service Gateway. Other departments may concentrate on voice and data communication, databases, company-standard desktop software, and so on.

Another example of an internal organization that serves the entire enterprise is Centers of Excellence. Many IT organizations have adopted the Center of Excellence (CoE) model as a practical way to consistently and continuously improve IT operations. A CoE provides the entire organization with visibility into standardized quality metrics and key performance indicators (KPIs). This helps keep all stakeholders informed, and keeps applications aligned with business objectives. Many companies formed Centers of Excellence, such as: Performance CoE, Security CoE, Six Sigma CoE, and the like.

It would make a lot of sense to consolidate the internal service activities inside one ALM project, thus setting a basis for a company-wide report on the object of service.

Service Provider

There are numerous examples of HP ALM implementations on service provider sites. Service provider areas of operation are different – be it cellular operators, cable and satellite TV suppliers, internet providers, and the like – but they all have a common ground of offering their services according to some business model. For example, a cable TV supplier may organize its work by region of service, whereas internet providers usually break their activities by type of internet infrastructure or speed. Cellular operators divide their customers by business type – private, small-medium, or enterprise, as well as by other parameters, such as mobile phone vendors.

Therefore, the imaginary *HP Telecom* provider would build its project topology based on vendor-specific domain with its related mobile phone models. Hence we would see *Palm* domain with *Palm Pre, Palm Pre 2,* etc. models as well as *iPaq* domain with *iPaq 5910, iPaq 211 Enterpris,e* etc.
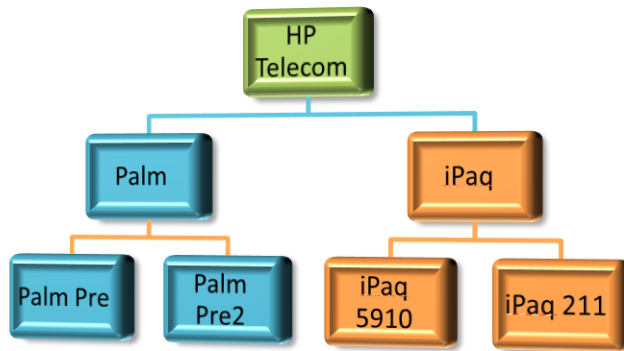
When a new phone model is unveiled, HP ALM administrator will open a new corresponding project to concentrate all relevant activities, such as regulatory pollution, radiation and frequency requirements, test cases, and subsequent defects.

4 Other Aspects

In addition to the project topology planning and implementation, there are some other aspects of the HP ALM planning process that need to be addressed in order to better utilize its capabilities.

Folder Structure

Folders are the containers of the assets stored in HP ALM. Therefore, organizing them in the best possible manner would simplify the initial deployment and forthcoming usage.

Here are some tips that would help in everyday work:

— Naming conventions

Folders and test sets should be named according to the defined company policy. Test set names should be unique so HP ALM does not calculate results together. Consider having prefixes in the name of the test set, such as <test set name>_build5.1

— Hierarchy

Based on functional area or feature, it should address certain parts of the whole system and provide clear distinctions for users. For example, one would build a requirements tree in the following manner:

- Common Requirements
- Product-specific Requirements
  - Functional
    - Feature 1
    - Feature 2
  - Non-Functional
    - Performance
    - Security

— Categories

You may consider adding a standardized user-defined field (UDF) of folders to categorize them. It is helpful in filtering and reporting.

Template Project

If we take the concept of naming conventions one step further, it is a good idea to coordinate user-defined fields, lookup lists, workflows, and customizations according to company-wide policy. Plan in advance for common terminology for:

— Priorities

— Severities

— Test Phases

— Test Environments

and use them throughout the entire ALM implementation.

Then there is the concept of cross-project customization that allows the HP ALM administrator to define the most effective workflow and key project attributes in a template project, link it to the projects that need to comply, and automatically propagate it to all projects that are linked to the template.

Since template projects are created per domain, it allows for great flexibility in the actual implementation – as per selected project topology approach. For example, if projects are structured around LOBs, then it is recommended to create a template for each line of business which would contain its specific customizations. If the workflow changes or the process requires a new set of fields, the changes can be made only to the template project. All projects linked to the template can easily be synchronized to pick up the changes.

Template projects provide visibility into the status and quality of all projects and enable entity sharing and a number of other critical features and concepts in HP ALM. We recommend using cross-project customization to make sure that the fields used in all your projects have common definitions, making it easier to import and export libraries. Only the fields that have common definitions can be shared across projects. Although template-based projects have a common customization, HP ALM is flexible enough to allow each project to keep its own individual customization as required.

5 Conclusions

The ALM space is being driven by time-to-market issues, while seeing an increased need for governance and regulatory compliance. Although many companies begin by looking for the tool, most are in need of improved practices and guidance. Issues, such as where should we focus, what roles are essential, or how can automation be effective, are at the root.

HP ALM 11 meets the needs of the modern application lifecycle by providing increased alignment between teams, including integration into strategy and planning teams, an offering of best practices to spur innovation and prevent tactical delays, and a bridge to the critical last mile of the operations organization. HP ALM 11 is a unified platform designed to master the application lifecycle from end to end.

All of us are interested in improving the quality of the applications we produce on behalf of our respective departments, lines of business, and enterprises. What is required to make this goal achievable is a lifecycle approach to quality, structured planning and design of the environment, consistent use of best practices, state-of-the-art tools such as HP ALM, and intimate knowledge of company business and "political realities."

The properly structured ALM projects that map real business processes bring unrivaled advantages to the organizations such as:

— Rework reduction

   Through entities sharing and reuse, companies maintain centralized libraries for managing requirements, test scripts, test resources, change and configuration management, and keep track of shared entity traceability including execution history.

— Always-on reporting

   If HP ALM implementation resulted in the creation of multiple projects according to the recommendations, then a built-in feature of cross-project reporting provides customizable visibility into application projects, trending analysis and insight, dashboard, aggregate data and metrics for quality, requirements coverage, defect trends in order to assess the overall quality, and readiness of the release.

— Addressed business demands

With the process standardization described in this document, companies can address the definition of standardized lifecycle process, enforce consistent best practices, maintain centralized customization of project templates, and execute an automatic update push to projects at change time.

— Freedom in administration

Once assets are spread between multiple projects, the ALM site administrator is free to schedule maintenance tasks such as backup/restore on a project basis without the need to stop the entire operation.

— User management under control

Multiple projects may have different sets of users with different privileges granted. From the ALM site administrator's point of view, it allows assigning a project manager to handle access to the specific project and grant permissions, and therefore limit project content exposure. From the end user prospective, be it a business analyst, a developer, or a testing person, it allows concentration on the issues at hand and removes the burden of unrelated entries.

These are just some of the benefits of properly implemented ALM project topology. We hope that best practices listed in this document will better help utilize HP ALM usage in your organization.