

**Code Migration**

1.2.6.1 Project Business Case

January 21, 2021

**Business Need**

The license for the current code management and migration[[1]](#footnote-1) tool, HP PPM, will expire in March 2021. The Company desires to find a replacement tool for code management and migration.

Stakeholders: Impacted individuals will be those involved in code management and deployment. This could be third party vendors, like AddOns migrating ABB (Ellipse) code patches and updates, as well as internal developers working on in-house developed application (see listing in **fig. 1** below)

**Analysis of the Situation**

Goal: Maintain appropriate control over code release and deployment management

Below is the scope of work covered by HP PPM:

*Introduction to HP Deployment Management[[2]](#footnote-2)*

*HP Deployment Management is an Project and Portfolio Management Center (PPM Center) product that automates the migration and deployment of software code, configurations, and content. HP Deployment Management groups these objects into packages and routes the packages along business processes modeled in HP's configurable workflow. The workflow automatically moves each package through required steps, which typically include system build, testing, quality assurance, staging, and final deployment.*

*HP Deployment Management automatically deploys the application components (such as XML content, HTML files, Java™ programs, and Oracle application configurations) that each package requires. For example, HP Deployment Management connects to the development environment, copies Java files from the version control repository, and transfers the files to the quality assurance environment. It then uses the Java compiler to compile and pack all of the files into a single Java program. The customer can use a built-in scheduler to schedule deployments. HP Deployment Management maintains an audit trail for all activities, including package information, application components altered, approvals obtained, and deployments performed.*

Without HP PPM the Company still needs to find a way to perform the following key control functions (additional requirements are noted at **Exhibit A**):

1. Change control and versioning over code - As teams design, develop and deploy software, it is common for multiple versions of the same software to be deployed in different sites and for the software's developers to be working simultaneously on updates. Versioning allows forking, merging, rollback, and audit of changes.
2. Compilation – this will occur locally using an IDE or SDK or with the deployment and migration tool.
3. Code Migration - programs are moved from one machine to another, often moving parts of its execution environment along with it. The intent is for the transferred program to continue execution on the target machine.
4. Code Migration across various platforms – Azure cloud, Oracle cloud, mobile, on-prem.

Release and Deployment Management is needed to properly control in-house developed software and some off the shelf software.

Key software requiring deployment management:

**fig. 1**

|  |  |  |
| --- | --- | --- |
| **Platform** | **Name** | **Notes** |
| Oracle | RCMS | Migrating to Oracle Cloud |
| Oracle | Land Management System (LMS) | Migrating to Oracle Cloud |
| Oracle | Revenue Management System (RMS) | Migrating to Oracle Cloud |
| Oracle | LIM Price Interface (part of RMS) |  |
| Oracle | Bond System | Migrating to Oracle Cloud |
| Oracle | HR/Payroll Hub | Migrating to Oracle Cloud |
| ABB | Ellipse | Has a Git component already |
| Oracle | ACES | Migrating to Oracle Cloud |
| Mobile Apps | various | Currently leveraging DevOps/Pipelines |

Options Considered

1. Continue using HP PPM in an unsupported state
2. Migrate to Git/GitHub Enterprise with Microsoft DevOps / Azure Pipelines
3. Migrating to Git competitor: Mercurial, Team Foundation Server, Bazaar, Fossil, CVS/SVN, etc.

Cost considerations

1. Continuing to operate unsupported has the lowest upfront cost but the highest overall cost risk.
2. Per review of GitHub website (<https://github.com/pricing>), Enterprise pricing is $21 per user/month. Azure Pipelines - 5 developers are free and it was ~$5-$8 per developer after that

1. GitHub competition: pricing varies

Known Risks

1. Continue to operate unsupported
   1. Arch will not be able to obtain direct vendor support
   2. In the event of an issue that requires outside support, Arch will be reliant on 3rd party consultants. Without a 3rd party on retainer, resources may not be readily available to assist which will typically lead to prolonged issue time and increased consultant costs
   3. Hardware or software issues will eventually occur which has the possibility of unacceptable downtime of business critical applications
   4. Security and vulnerability risks by not operating on the latest Oracle versions and patch set
   5. Operating in an unsupported state for an extended period of time is not recommended
2. Relicensing and maintaining the applications on premise
   1. HP PPM has quoted a $39k dollar number to reinstate support
   2. Arch will still be required to accommodate various migration platforms – Azure/Oracle cloud and on-prem.
3. GitHub (or other platform if not Git) Cloud Migration
   1. Arch IT will need to evaluate and ensure all core requirements are met with Git. Upon migration, resources will need to be trained to execute via the new code management/deployment tool. A stated goal is to make the IT team more efficient in this space, so this should be a welcome change
   2. Cloud based tool may have security issues which will need to be continually monitored and addressed. (this is no different really than an on premise tool, security will always be front of mind)
   3. Need to manage on-prem local code compilation and cloud based versions (not sure this is a risk either, we currently maintain local (developer’s machine) and server versions. The difference here is adding cloud vs on-prem
   4. Will moving to GitHub increase or decrease Company’s reliance on internal IT personnel? How complicated will the GitHub environment be with associated scripts, Jenkins, etc. I’d add this as other considerations, not risk, especially the part about Jenkins. Bill has latched onto Jenkins as a requirement. It is not, it is a nice to have at this point. We need to have further conversations with Microsoft/SNOW about Git and the integration with SNOW[[3]](#footnote-3)
   5. Arch has limited resources with already full plates. Level of effort to migrate will need to be analyzed and prioritized with other in flight efforts

**Plan**

Arch IT has discussed with Microsoft[[4]](#footnote-4) about using Azure DevOps/Pipelines or Github. Additionally, Danny Wolf, current contractor working on mobile apps, is actively using DevOps/Pipelines. Based on the PoC it was determined migration should utilize the assistance of *xxx* consulting services to design the cloud ( *… plan in process*)

**Recommendation**

Management recommends approving the use of GitHub as our repository, and either Azure DevOps or Jenkins to deploy applications to the cloud and on-prem.

( - ***premature until further analysis and PoC*** - )

**Exhibit A**

**Requirements (**updated with current list from Paul/Philip**)**

The following list of requirements were drafted by the Arch IT Applications group (this should not be considered a comprehensive list)

Oracle proprietary applications:

1. Source code management
   1. Retain historical versions of code source and metadata from PPM
   2. Easily manage source code related processes
      1. Check-out, check-in multiple objects at once
      2. Check out source locking/not locking
         1. The ability to see who has an object locked, and a process for others to break the lock.
      3. Versioning
         1. History
         2. Reporting
         3. Deployment status
      4. Search
2. Deployment
   1. Serialized deployment and compilation, basically supporting the concepts of dependencies.
   2. Compilation feedback (success / fail) and the ability to abort deployment or continue.
   3. The ability to bypass execution in test as data updates are often against data not existing in test.
   4. Ability for the users to sign-off on testing without involving the Apps team.
   5. The concept of grouping applicable source together as a project / release
   6. Tracking release attributes and reporting
   7. The ability for the Apps personnel to individually deploy any needed change without preapproval through an “emergency” change process.
   8. The ability to deploy non-emergency CAB reviewed / approved changes during the approved deployment window without the need for Change Manager Involvement. (a nice to have)
   9. Migration to various environments dependent on applicable parameters (a nice to have)
      1. Based on the “owner” of the object, not an “application bucket”
3. SOX Compliance (not to be confused with SOC 2 compliance)
   1. Re-review and support necessary controls relating to the Sarbanes-Oxley Act on applicable applications (RMS / LMS only?)

1. Release and Deployment Management: IT Process Wiki, accessed 20 January 20201, <[<https://wiki.en.it-processmaps.com/index.php/Release_and_Deployment_Management>.](https://wiki.en.it-processmaps.com/index.php/Release_and_Deployment_Management#:~:text=Objective%3A%20Release%20and%20Deployment%20Management,the%20correct%20components%20are%20released.)> [↑](#footnote-ref-1)
2. Hewlett-Packard.(2014).HP Deployment Management User’s Guide. Retrieved from <https://tinyurl.com/yxpadd7x> [↑](#footnote-ref-2)
3. Comment from Marty Zambo [↑](#footnote-ref-3)
4. Microsoft Teams Meeting with Lily Wang ([wang.lily@microsoft.com](mailto:wang.lily@microsoft.com)), Roberto Mardeni, and Jamie Mandl about Microsoft DevOps/Pipelines and GitHub. [↑](#footnote-ref-4)