

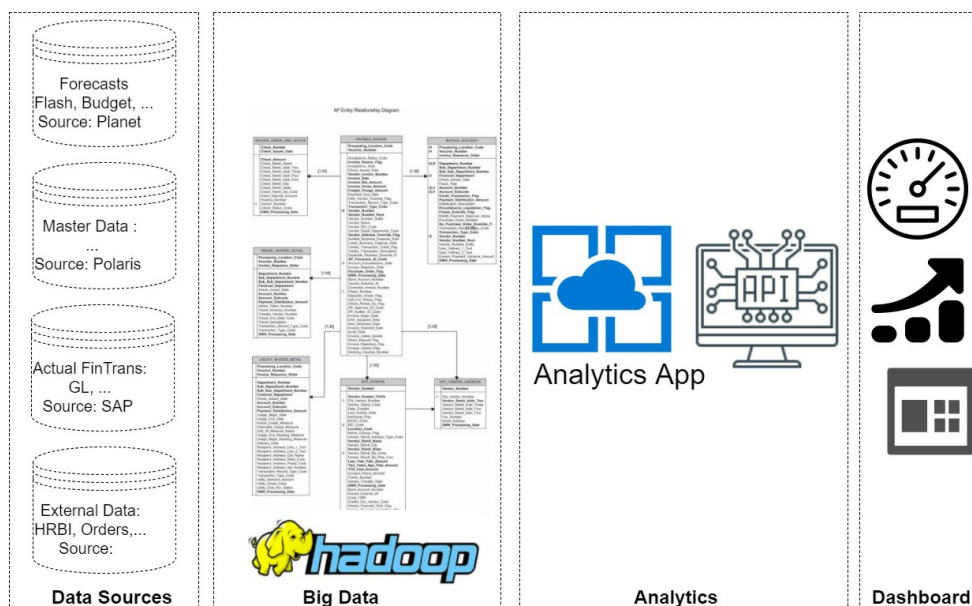
HPE Optimus Prime

Objective

- DataLake that hosts databases related to marketing department to support insightful platform for analytics
- Pull data from multiple sources and create a central hub
- Combine Actual, Flash, Budgeting, External datasets for reporting
- Fast, seamless, real-time stream of data to feed automated reporting

DESIGN

Layout:



Deliverables Roadmap (General Draft Version)

Phase CANOPY: Two Months

0. Define Plan (1 Director and Support Team)

- Write Charter--1 page/Bullets
- Roadmap Milestones
- Schedule Deliverables
- Build Team
- Setup Logistics
- Provide Budget

1. Draw Blueprint (1 Architect)

- Draw Layout
- Highlight Components
- Choose Technologies
- Specify User View (API)

- Draw swimlines
- Define Platform

Phase NEST: Two Months

2. Design Interface (3-7 Designers)
 - Code App Classes -- Class Interface
 - Define DFD, ERD, DDLs, Database Schema
 - Draw UI -- Forms
 - Specify Test Specs -- Measures and samples
 - Structure Folders and Packages -- Repos structure
3. Draft Markdown (3-7 Modelers) -- One Month
 - Docstring
 - Math, Formula, Queries, DB Views
 - Algo
 - Pseudocode
4. Develop Prototype (Army of Developers) -- One Month
 - Implement
 - Comment
 - Evaluate
5. Detail Test (Army of QA Testers) -- One Month
 - Add Exceptions
 - Optimize
 - Verify
 - Validate
 - Ship

Phase HIVE: Two Months

6. Deliver Product (Toolsmith Team) -- One Month
 - Assemble
 - Run
 - Track and Select
 - Optimize
 - Release/Deliver
7. Deploy Service (Deployers Team) -- One Month
 - Setup
 - Manual
 - Train
 - Support
 - DevOps

Organization

There are three types of teams:

CANOPY Team

This is the high level team with following roles:

- Director: Supervisor
- Architect: Lead Role
- Program Manager: Support Role
- Designers (Head of each Nest team)
- Head of Engineering/Toolsmith Team
- Head of Deploy Team

This is like a commander canopy that decides and communicates about the major parts of the system including plan, schedule, budget, scope, features, architecture, ...

NEST Teams

Each Designer is in charge of one NEST which develops one components of the system. Components must be modular and separable to reduce the complexity. Each NEST has the following roles:

- Designer: Lead Role
- Modeler
- Developers
- Testers Each NEST might have many developers and testers and/or one person might play a few roles if the team is small.

HIVE Teams

Which functionally partitions into Platform Engineers and Deployers and they might have variety of roles from DB Admin, DevOp, IT, Trainer, Marketer, ...

Platform Engineers take the developed modules from NESTs and put it into production. Nowadays, it is done by assemble them up and running on PaaS or SaaS platform.

Deployers are the one that sale, deploy, train, ... the product to the clients and create a service out of a product.

Agile Releases

Each release length is two months. There will be one release in each phase in continuous production. For instance for software x release 5.0 will be in HIVE while the next one (6.0) is in NEST, and the future one (7.0) is in CANOPY. The first Release will be longer (6 months) before going out.

Work Breakdown

Architecture: CANOPY

Integrity of software comes from single Architect mindset that communicates it to the CANOPY team and they decide on management aspect. The team, then, decides what feature to include (scope) management, how to modular the component and assign to designers (breakdown), how to manage and support the NEST teams.

Development: NEST

The NEST team will work under designer lead with tight teamwork. The designer will write the APIs, class interfaces, datatypes, ... then the modelers will write the how-to in markdown format that shows formulas, how-to, math, algo, pseudocode, ... to implement the design interfaces. This will be the conceptual design part.

The experienced and super productive developers will just write the overall code (in interpretive, prototype, dynamic-typed language such as Python). This is the logical design.

The tester will add exceptions, optimize the code, test it, validate it and plug it into the appropriate packages and ship it to production.

Delivery: HIVE

The final components by NESTs will be assembled together according to architecture layout and will be on production in PaaS/SaaS platform by Engineering/Toolsmith team.

The final deployment and marketing and training and manual writing and delivery will be done by Deployment team.

Footnote

Most of software development efforts fail. A report shows (developing software for government) deliver the exact software as promised under 5%, 78% never delivered, and 17% delivered but abandoned and never used. Software development deviates from traditional Project Management. The traditional management is based on collective knowledge mainly from construction, army, and industrial production. Software (And Data Product) Development, which becomes increasingly more critical part of economy, is different for a reasons:

- It is not tangible
- Computers produce it: they must understand what end user wants through the development team communication
- The marginal cost is (theoretically) zero: Copy code from one system to another.
- The possibility to make is (theoretically) unlimited: You just need to code
- Intolerance to error and mistake: Even a colon can bring the whole system down. Indeed, on top of mistake, many new vocabs like error, exception, bugs are there.

There are three movements in software development that influenced this documents.

- Agile Development

Agile emphasises on rapid development, early releases, early feedback from market, and continuous releases. Also it emphasis on teamwork and informal communication and lean overhead and management team.

- Code The Architecture

Or "coding is designing". A separation between code and design is a big problem in software development. This movement emphasis that since the machine does the production (byte code is production) so in software development, there is no such a thing as implementation. Coders are bind to designers. eXtreme Programming, takes it to extreme, and demands everything should start and end with coding. The diagrams and views are just reverse-engineering of coding and artifact of the design.

- Mythical Man-Month

It comes from the observation that throwing more people to boost a software, actually, delays the project. SO it emphasis on surgical teams in which there is just one super-star and a few others that help. It also emphasis that the software integrity comes from single mind architecture, and successful implementation comes from tight teamwork. NEST and CANOPY is influenced with this movement.