usp lab.

ユニケージ開発手法と

プロジェクト推進方法

有限会社ユニバーサル・シェル・プログラミング研究所

ユニケージ開発手法技術概説資料　[1]

Universal Shell Programming Laboratory

© usp lab. 2024　Confidential Proprietary
この文書には、有限会社ユニバーサル・シェル・プログラミング研究所の機密・所有物が含まれています。
許可された者以外は使用しないでください。

目　次

1. ユニケージとは何か
2. ユニケージプロジェクトの進行
3. トッパ工程とは
4. データ基盤系システム
5. アプリケーション系システム
   5.1 アプリケーション設計のポイント
   5.2 アプリケーション開発のポイント
   5.3 アプリケーションテストのポイント
6. デビュー
7. ユニケージのドキュメント
8. プロジェクト推進上　留意すべきポイント（１）
   プロジェクト推進上　留意すべきポイント（２）
9. ユニケージ開発手法　製品ラインナップ

© usp lab. 2024　Confidential Proprietary

1. ユニケージとは何か

usp lab.

- ユニケージとは、業務システムを「やすく･はやく･やわらかく」作るための「手法」で、システム内製化に向いています。 → 資料 1
- 先に実データを移行して整理し、次にアプリケーションを開発します。（データが先＝一休方式） → 資料 2
- UNIXコマンドの拡張である usp Tukubaiコマンドを使います。 → 資料 3、4
- プログラムはUNIXの基本の考え方を発展させた独特のもので、「お作法」に従わないと効果的に動作しません。 → 資料 5

© usp lab. 2024 Confidential Proprietary

資料　1

usp lab.

やすい

コストが安い／プログラムが易しい

■安価なＰＣサーバとＯＳ（Linuxなど）
■データはテキスト、プログラムはシェルスクリプト、**ミドルウェアは不要**
■システム更新無し（ハードやＯＳが変わっても、ソフトとデータはそのまま）

はやい

開発が早い／処理が速い

■プログラムが非常に短い
■１人の技術者が設計～開発～運用をする多能工方式
■無駄な機能を削ぎ落とす＋ハード性能を最大限に引き出す

やわらかい

柔らかい

■プログラムがシンプルでカスタマイズが容易
■ソフトの各機能が相互依存しておらず、簡単に変更が可能
■整理整頓されたデータから、アプリケーションに必要なデータを生成

得意領域

・基幹業務システム処理
・大量データの高速処理 （集計・検索・帳票など）
・データウェアハウスやBI
・システム間のファイル連携処理（ETL）

等を得意とし、技術習得が容易で **‘内製化’** に適した開発手法です。

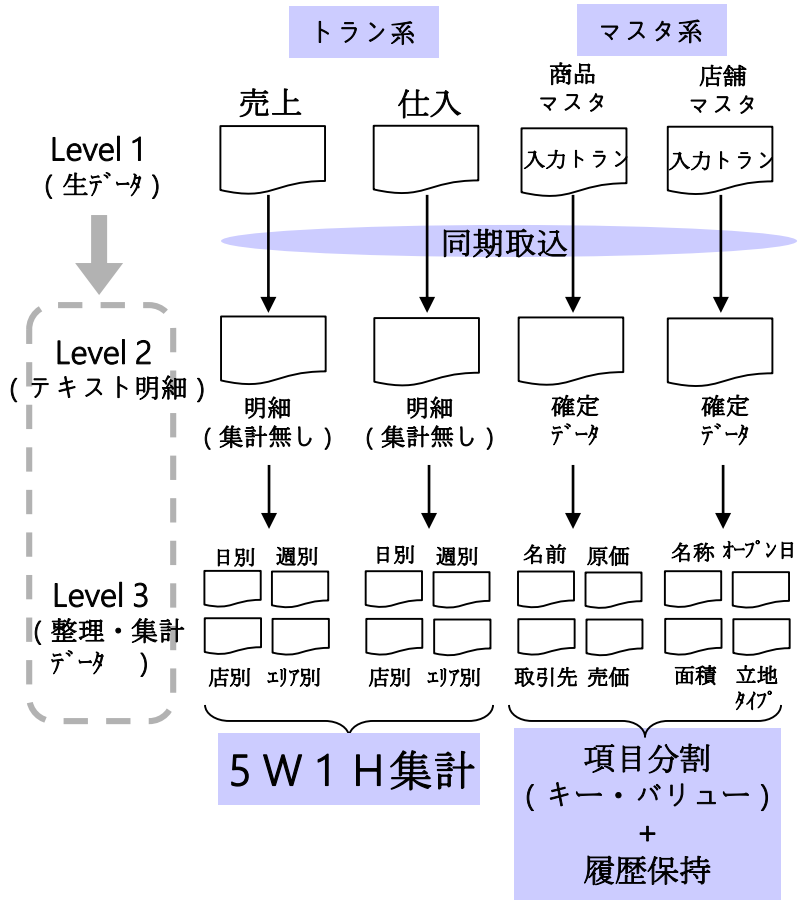
Universal Shell Programming Laboratory

© usp lab. 2024　Confidential Proprietary

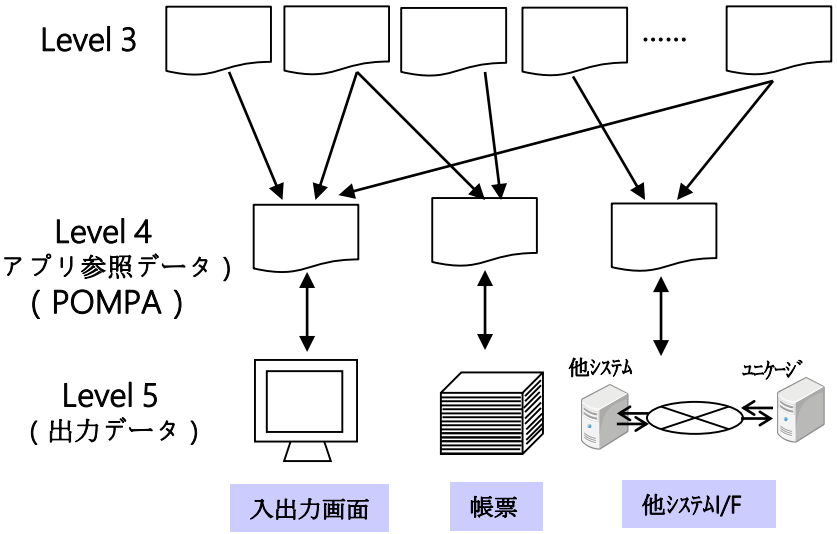
資料　2

usp lab.

第１フェーズ（データ基盤系システム）



第２フェーズ（アプリケーション系システム）



生データ取得→整理のシステムの構築（第１フェーズ）と、アプリケーション系システムの構築（第２フェーズ）が重なり合って進行する。

© usp lab. 2024　Confidential Proprietary

資料　3

usp lab.

uspTukubaiとは、
・単純なコマンドを組み合わせることで簡単に複雑な業務処理を行えるUNIX拡張コマンドセット

よく使うものは100個もない

cp	コピー	UNIXコマンド
find	ファイルの探索	
sort	並べ替え	
awk	項目間演算	
：		
self	項目選択	uspTukubaiコマンド
join0	データのマッチング	
sm2	サムアップ	
keta	桁揃え	
ulock	排他制御	
：		
mdate	日付管理	各企業用コマンド

すべてのアプリケーションをシェルスクリプトで記述します。
バッチ処理だけでなくWebアプリなどのトランザクション処理も可能です。

バッチ処理シェルスクリプトの一例

```
#!/bin/bash
 join0 key=1 MASTER URE      | データのマッチング
 self 2 3 4 5                | フィールドのセレクト
  msort key=1/2              | ソート
  sm2 1 2 3 4                | サムアップ
 sm4 1 1 2 2 3 4             | 中間計
 self 1 2 4 3                | フィールドのセレクト
 sm5 1 3 4 4                 |  合計
 map num=1                   | 縦横変換
 sed ‘s/A/Sales/g’           | 文字列の置換
 sed ‘s/B/Profit/g’          |  〃
 keta 4 6@NF-1               | 桁そろえ
 comma 3/NF                  | カンマ編集
 cat header -                |  ヘッダーを付加
 tocsv > result              |  CSV出力
 exit 0
```

Universal Shell Programming Laboratory

© usp lab. 2024　Confidential Proprietary

資料　4

usp lab.

Webブラウザからのトランザクション処理シェルスクリプトの一例

```
#!/bin/bash
dd bs=$CONTENT_LENGTH | cgi-name > name   # WEBサーバから情報取得
case “$(nameread MODE name)” in           # 処理モードによる分岐
SEARCH)                                   # [検索]
   if ulock -r MST.LK; then               # シェアードロック
      nameread KEY name                   # 検索キーの取得
      join0 key=1 - MST                   # マスターデータの検索
      mojihame -lLABEL html               # HTML へのはめ込み出力
      rm MSTLK                            # ロックファイルの消去
   fi ;;
UPDATE)                                   # [更新]
   if ulock -w MST.LK; then               # 排他ロック
      nameread -el “KEY|VAL” name > TRN.123  # キーと値の取得
      upl key=1 MST TRN.123 > MST.123     # 更新マスターの作成
      ln -s MST.123 MST                   # 同名でアクセス可能にする
      cat next_html                       # 次画面の出力
      rm MSTLK                            #ロックファイルの消去
   fi ;;
esac
exit 0
```

→更新処理では、ファイルそのものは更新せず、
新たに作成したファイルを参照する 。

大量データを集計してレポートを出力するシェルスクリプトの例

```
#!/bin/bash
clust-join1 slavefile1 key=1 master URE  # データの結合(Bigdata)
para-self 10 2/NF-1                      # 列選択(Bigdata)
clust-sm2 slavefile2 1 2 3 4             # 集計
sm4 1 1 2 2 3 4                          # 中間計
self 1 2 4 3 5                           # 列選択
sm5 1 3 4 4                              # 合計
map num=1                                # 縦横変換
sed 's/A/Sales/g'                        # 文字列置換
sed 's/B/Profit/g'
keta 4 6@NF-1                            # 桁揃え
comma 4 5                                # カンマ編集
cat header -> result                     # ヘッダー付けて出力
exit 0
```

slavefileに
スレーブサーバ名と
並列処理数を列記

※ clustコマンド群とparaコマンド群はuspBOAのコマンドセットになります。

Universal Shell Programming Laboratory

© usp lab. 2024　Confidential Proprietary

資料　5

usp lab.

シェルスクリプトは記述の自由度が高いため、書法「お作法」を重視する

- スクリプトヘッダーの書法
- コメント６０％
- 変数名／ファイル名のルール
- 一時ファイルの命名法
- １コマンド１行の原則
- ファイル渡しの原則
- 環境変数渡しの禁止
- インクルード処理／関数の禁止
- ファイルレイアウトの記述法
- 走行ログ記録の書法
- 起動／終了時刻の記録
- セマフォファイルの生成
- 短く書く（実効１００行以内）
- ゴミファイルの消去
- 複雑なｉｆ文はスクリプト・データを分ける
- バージョンを作らず、バックアップする
- 多層コールの禁止
- Ｃｏｐｙｒｉｇｈｔの上書き
- 処理ファイルの大きさを理解する　　等々

DBMSの機能は、ユニケージではプログラムの書き方で実現します

排他制御
・ロックファイル／セマフォを利用したアトミック処理コマンド（ulock）

順番制御
・繰り返し型サーバコマンド（server, ncat, rfifo）

コミット
・ユニケージではファイル成立と同義（システムコールの返値で確認）

ロールバック
・データ履歴が保存されているので、どの時点にでも復旧可能（時制データベースとして機能）

並列処理
・paraコマンド群＋clustコマンド群

アクセス集中
・負荷分散サーバプログラム
・順番制御プログラム
・リソース制御（cgroupなど）

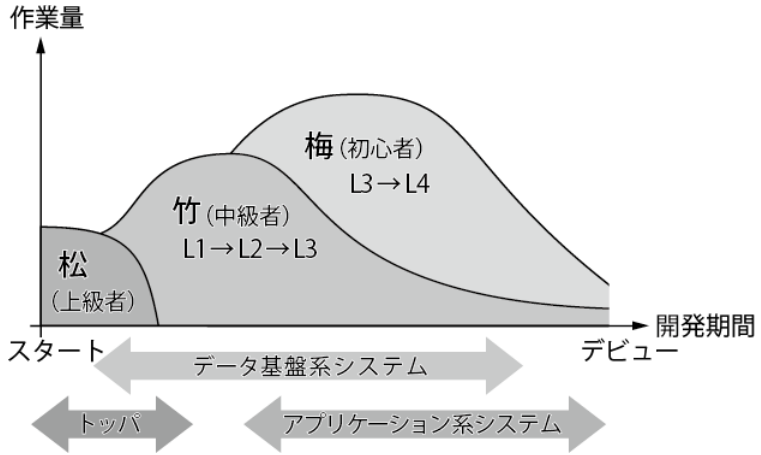
Universal Shell Programming Laboratory

© usp lab. 2024　Confidential Proprietary

2. ユニケージプロジェクトの進行

usp lab.

◆ユニケージは従来のウォーターフォールとは全く異なる進め方をします。



ユニケージではエンジニアを3種（松、竹、梅）に分類しており、各工程ごとにそれぞれに応じた役割を担当をする。

プロジェクトの工程

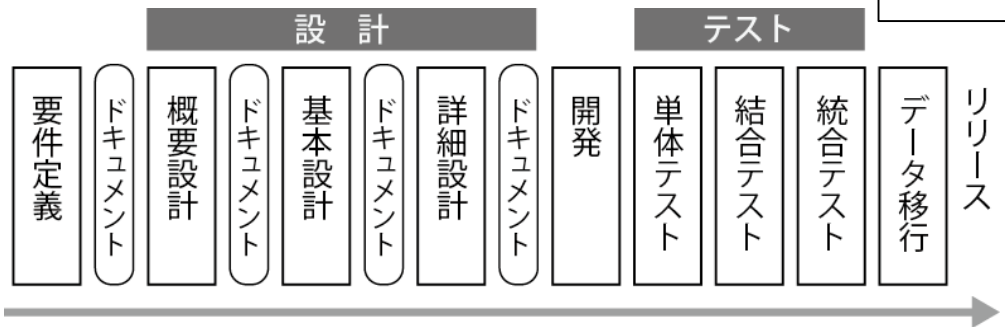
(1) トッパ工程　(松)
第１のヒアリング
全体を俯瞰した設計と方針決定
スケジュールと工数概算の提示
肝となる技術課題の解決
代表的アプリケーションの開発

(2) データ基盤系システムの作成　(竹)
第２のヒアリング
サイジングとサーバの導入
L1～Ｌ3システムを構築（データ基盤）
プロジェクトチームの管理

(3) アプリケーション系システムの作成　(梅)
第３のヒアリング
アプリ系システムの開発とテスト(基本機能＋便利機能)

これらが重なり合いながら進行します。

（比較）従来のウォーターフォールの進め方

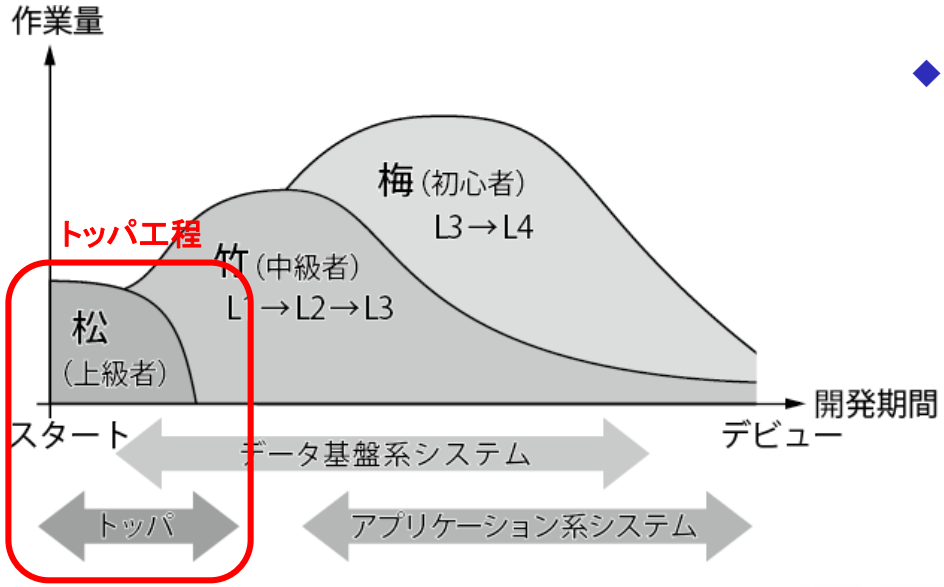


Universal Shell Programming Laboratory

© usp lab. 2024　Confidential Proprietary

3. トッパ工程とは

- トッパ工程とはプロジェクト全体を見通す作業です。
- 1回目のヒアリングをキーマンに行い、既存システムの概要と今回のプロジェクトの目的を理解し、プロジェクトスコープとシステム方針を決定します。
- 実データを取得してプロジェクトを象徴する代表アプリを作ります。
- 重要な技術的課題を見抜き、試作して解決します。
- お客様の不安感を取り除きます。



◆ **トッパ工程**
第１のヒアリング
全体を俯瞰した設計と方針決定
スケジュールと工数概算の提示
肝となる技術課題の解決
代表的アプリケーションの開発

© usp lab. 2024 Confidential Proprietary

4. データ基盤系システム

◆ 次の手順でデータ基盤系システムを作ります。

(1)2回目のヒアリング

・業務のキーマンと現行システムのキーマンに2回目のヒアリングを行い、アプリが必要とするデータと、システムに入力されるデータは何かを見出します。

・必要な基礎データ（ L3）は何を作るのか、業務のキーマンのヒアリングから、データ構造を設計します。

(2)サイジングとサーバの導入

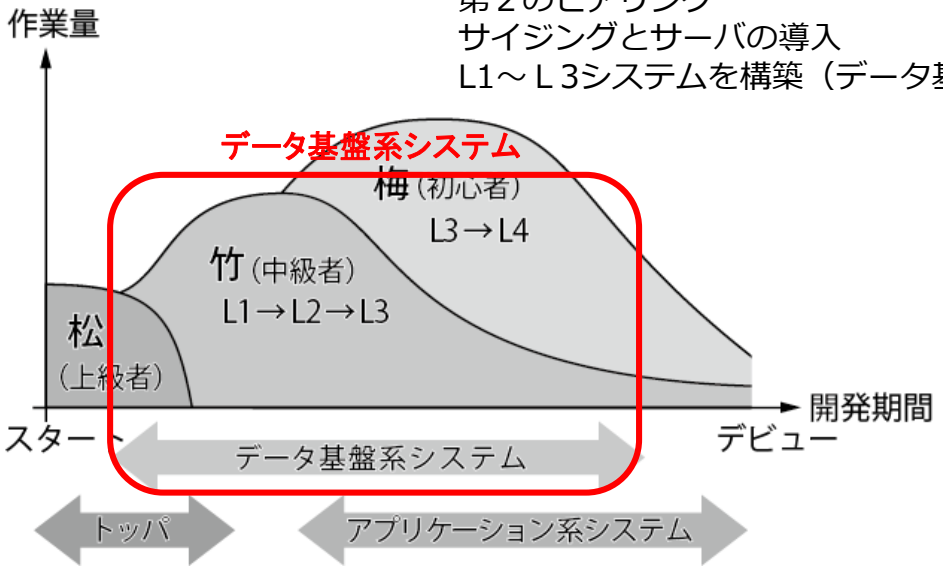
資料 6

(3) L1～L3システムの構築とデータの検証

資料 7

◆ **データ基盤系システムの作成**

第２のヒアリング
サイジングとサーバの導入
L1～Ｌ3システムを構築（データ基盤）



© usp lab. 2024 Confidential Proprietary

資料　6

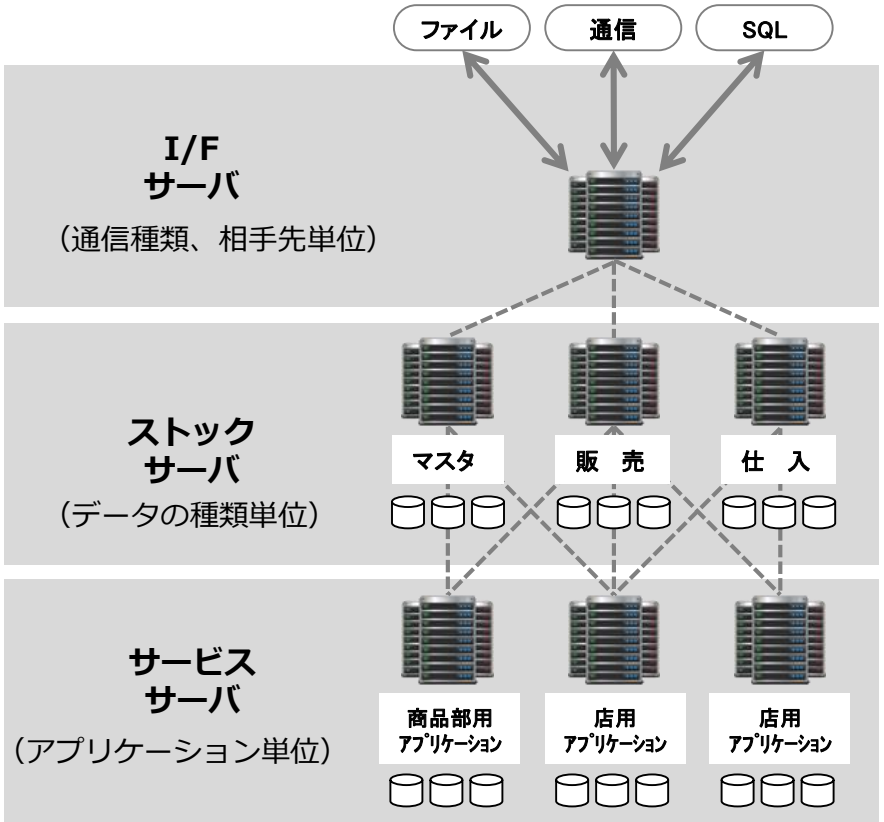
usp lab.

【サイジング時のポイント】

◆ サーバは役割ごとに、次のように分けます。
容量に余裕があれば同居してもかまいません。
・I/Fサーバ
外部システムとデータを送受信する
・ストックサーバ
業務毎に発生するデータを L1～L3 まで
整理して蓄積する
・サービスサーバ
ユーザ向けアプリケーションを提供する

◆ L1～ L5までのデータ種類と量を想定し、
ディスクの使用量は50%以下にします。

◆ サービスサーバのアクセスは1コア当たり
最大4アクセス/秒にします。



【導入時のポイント】

◆ サーバスペックはすべて同じものにします。
互いに交換ができ、運用しやすいです。

◆ サーバ名はその役割が分かる名前にします。

◆ ハードウェア障害に備えてバックアップ機なども導入します。

Universal Shell Programming Laboratory

© usp lab. 2024　Confidential Proprietary

資料　7

usp lab.

Ｌ１：Ｌｅｖｅｌ １
（イベントデータ）

生データ（非同期）
情報が発生した通りに作成するデータ
（失うと困る→多重、多層、多角バックアップ）

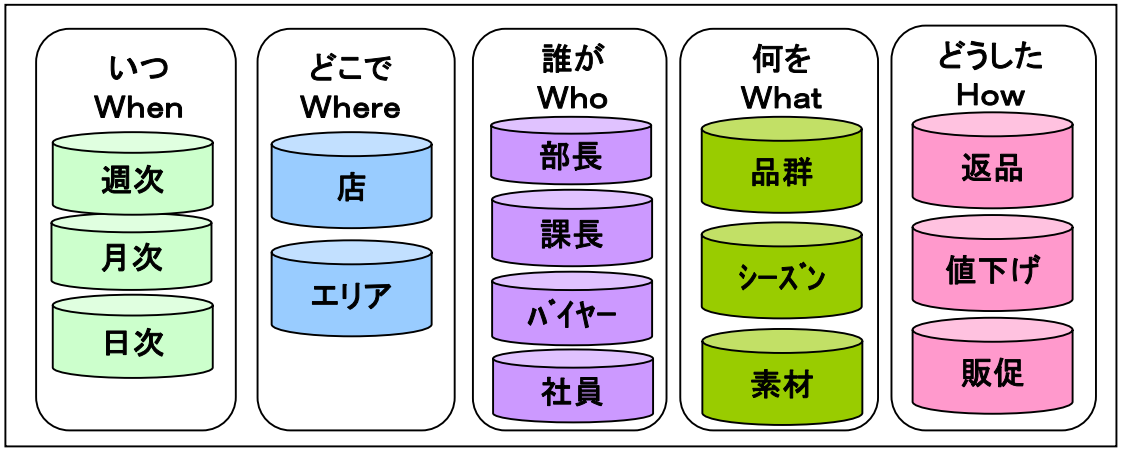
Ｌ２：Ｌｅｖｅｌ ２
（確定データ）

明細データ（同期）
生データをあるタイミングでシステムに正式に取り込んだデータ

Ｌ３：Ｌｅｖｅｌ ３
（5W1Hデータ）

Level 2データを５Ｗ１Ｈ単位に整理・集計したデータ（アプリケーション非依存）
必要に応じて追加できるので想像で作りすぎない
業務単位や指標単位で、レイアウトはシンプルにし、期間単位で分けたファイルにする
マスターはキーバリュー形式にする

Level3：５Ｗ１Ｈで整理されたデータ
業務単位のＬｅｖｅｌ２データを業務や指標単位で集計



Ｌ４：Ｌｅｖｅｌ ４
（アプリケーションデータ）

アプリケーションが利用しやすいフォーマットにしたデータ　（別名：ＰＯＭＰＡ）

Ｌ５：Ｌｅｖｅｌ ５
（出力データ）

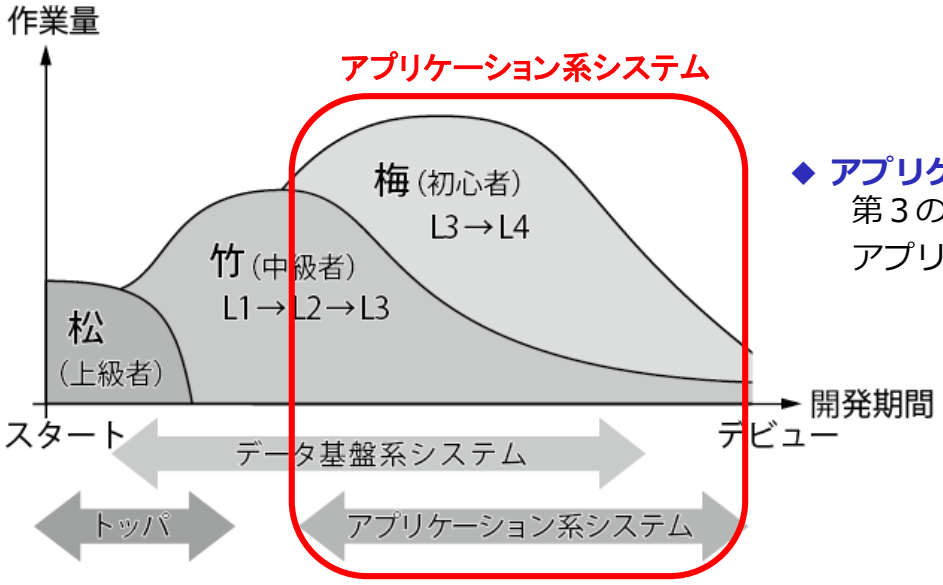
帳票イメージデータ（Ｅｘｃｅｌ、ＰＤＦなど）やログ等

Universal Shell Programming Laboratory

© usp lab.2024　Confidential Proprietary

5. アプリケーション系システム

- 画面や帳票アプリケーション、外部連携データの作成など、ユーザーや別システムに提供する、レベル３以降のデータを活用した機能をアプリケーション系システムと呼びます。
  レベル４の作成もここに含まれます。
- 梅エンジニアは、いち早くアプリケーションを完成させることが求められます。竹エンジニアは、その為に必要な指導を行います。



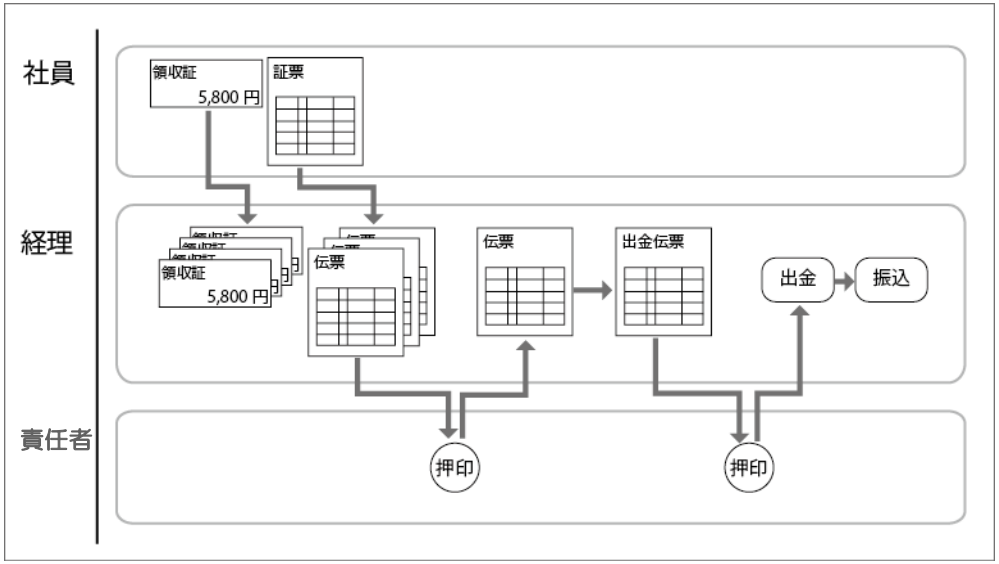
◆ **アプリケーション系システムの作成**
第３のヒアリング
アプリ系システムの開発とテスト(基本機能＋便利機能)

© usp lab. 2024 Confidential Proprietary

5.1 アプリケーション設計のポイント

① ３回目のヒアリングを、業務のキーマンや現場の担当者に行います。
② ヒアリングは業務ストーリー（現場担当者の一連の作業）単位で行います。
③ 業務のフロー図を書き、誰が何をいつ何のためにするのか理解します。
④ 画面や帳票のイメージ図を書き、項目の意味やロジック、画面の遷移を正確に理解します。

業務フロー図



⑤ アプリケーション別にディレクトリーを分け、必要なL4データ、発生するL1を格納します。

© usp lab. 2024　Confidential Proprietary

5.2 アプリケーション開発のポイント

usp lab.

① 業務ストーリーの設計ができたらすぐに開発します。
出来上がったらすぐに、業務ストーリー単位で、ユーザーに見せます。
何度も修正開発を繰り返し、ユーザーに見せて確認しつつ全機能を完成させます。
② 現実の業務における最大量の本番（ダミー）データを使って開発します。
③ 必要ならば L3 を竹に追加依頼します。
④ アプリケーション開発は「床屋方式」で行い、設計開発の時短に努めます。

業務重要度7割

業務の骨格となる設計

＝

早期に決定できる

短期間

1次開発

実データ入手
開発・テスト・リリース

元データと入出力画面設計のみで開発可能
（設計負荷が小さい）

少人月

業務重要度3割

便利機能設計

動くアプリを目前にすると比較的短期に決まる

短期間

2次開発（仕上）

開発・テスト・リリース

短いプログラム
＋
相互依存が無い
↓
修正開発が容易

少人月

⑦アプリケーションはシェルスクリプトで記述しますが、記述方法は「お作法」に則ることで、第三者の可読性を高めます

Universal Shell Programming Laboratory

© usp lab. 2024　Confidential Proprietary

5.3 アプリケーションテストのポイント

◆ 以下のテストをします。テストは画面や帳票の目視だけでなく、出力ファイルの内容をチェックしてテストの確度をあげます。

① 開発者テスト
開発者自身が、実データを使い、仕様通りの出力になっているか目視し
性能要件を満たす実行時間になっていることを確認します。
run / run-cgi コマンドを活用します。

② 第三者テスト
開発者でない第三者による、お作法テスト、網羅テスト、極値テスト、
バッチ連携テストを行います。障害が無くなるまで、①②を続けます。

③ ユーザーテスト
業務シナリオテスト＋使い勝手テスト
これはユーザーの受け入れテストです。

◆ L４をつくるなら１秒以内のレスポンスを目標にします。
それ以上かかるならファイルの配置を再考します。

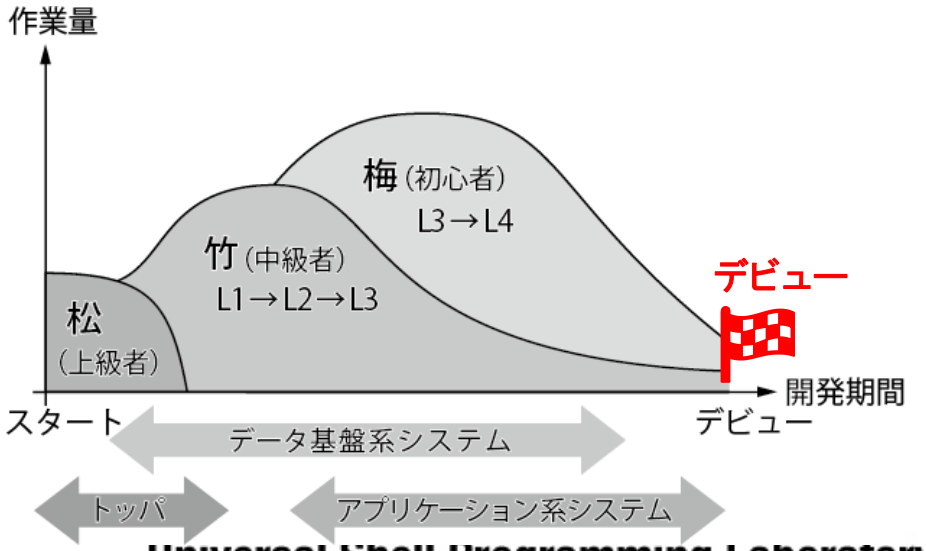
© usp lab. 2024 Confidential Proprietary

6. デビュー

- ユニケージではリリース（手放す）とは言わず、デビュー（初めてステージに立つ）と言います。システムに完成はなく、日々進歩するからです。
- ユーザー側の負荷が小さくなるよう、順次デビューも検討します。

デビュー後

- 操作性や見た目の向上に取り組みます。
- プログラム記述の統一に取り組みます。
- 無駄な項目、ファイル、アプリの除去を心がけます。
  （システムの手入れをします）
- 保守開発環境を構築します。



© usp lab. 2024 Confidential Proprietary

7. ユニケージのドキュメント

usp lab.

◆ 「開発時のドキュメントは少なく、保守時のドキュメントは実用的に」
お作法がベースにあるので、ドキュメントが少なくて済みます。

◆ 「**紙さま**」と称するシステム全体図を作成します。「紙さま」の内容は、
・サーバ配置
・ネットワーク
・データフロー
・アプリケーション配置　等
「紙さま」によってシステム全体像を知ることが出来ます。

◆ データ基盤系ではレイアウトファイル（L1～L3データの諸元）を作成します。
アプリケーション系では、業務フロー図を作成します。
処理ごとに処理概要とイン･アウト（入出力仕様）を記述します。
処理の詳細はプログラム中にコメントとして記述します。

© usp lab. 2024　Confidential Proprietary

8. プロジェクト推進上　留意すべきポイント（１）

usp lab.

プロジェクトチームの作り方

- ◆ 現行業務に詳しいキーマンと、現行システムに詳しいキーマンを必ず参加させます。
- ◆ 全員でテーブルを取り囲んで座れる、10名以下の少人数チームにします。
- ◆ 大きな仕事でも、少人数で順番に開発します。
- ◆ 会話による対面コミュニケーションを重視します。

顧客とのキックオフのポイント

- ◆ ユニケージプロジェクトの進め方を丁寧に説明し、納得してもらいます。
- ◆ 待っていてはいけません。ユニケージ・エンジニアは、顧客に積極的にアプローチし、仕様の行間を埋めて設計開発します。
- ◆ 黙っていてはいけません。明るい声で先に自分からコミュニケーションします。
- ◆ 顧客の会社組織･役割･歴史･業績･将来の方向性を知ります。
- ◆ 顧客の業務と意味を知ります。
- ◆ SEはその会社に就職するくらいのつもりで顧客業務を理解します。

データプログラミングのポイント

- ◆ １行書いたら実行し、データ加工結果が正しいのかを確認しながら進めます。
- ◆ 現実的な量のデータを用いて処理時間を測ります。その処理速度で本当に良いのかチェックします。
- ◆ 竹は、梅が上の2つの原則を守っているか常にチェックします。

© usp lab. 2024　Confidential Proprietary

プロジェクト推進上　留意すべきポイント（２）

usp lab.

性能改善のタイミング

◆ ユニケージではファイル配置（設計）がプログラムと処理性能を決めます。
満足できる処理性能が得られるまで、ファイル配置とプログラミングを繰り返し進めます。

短期間で開発するための心構え

◆ 開発は短時間を心がけます。用意周到な準備と集中力を高めた作業がポイントです。
① プログラムの全体が構想できるまでキーボードに触りません。構想は紙やホワイトボードを使って明確にします。
② 構想ができたら、一気に高速に間違わずにプログラムします。
③ 処理ブロックができる度に実行し、出力結果を目視確認します。
全部プログラムを書いてから初めて実行はNGです。
④ 離席せず、走行ログを眺め、処理時間を感じます。
ユニケージらしい高速処理を確認します。

◆ 合い言葉は、「間違わずに、速く短く打て！」です。

ユニケージエンジニアが達成すべきポイント

◆ ユニケージエンジニアが必ず達成すべき２つの目標とその順位は、
① 締め切りよりも早く終わらせる。
② 期待以上の品質を実現する。
全てのお作法はこの為にあります。お作法はあくまでベストプラクティスであり強制ではありません。
◆ ①②を実現するために顧客に積極的に働きかけ、仕様を早期に決定します。
◆ 作業を速くし、考えたり待つ時間を短くし、何事も「スピードを上げる」よう努めます。

© usp lab. 2024　Confidential Proprietary

9. ユニケージ開発手法　製品ラインナップ

usp lab.

標準

usp Tukubai

ユニケージ開発手法で中心となる、OS標準のコマンドと組み合わせて使用するコマンドセットです。OS標準のコマンドセットでは不足している機能を補ったり、プログラムを短く書くための工夫がなされています。高速処理も特長です。

クラスタ

usp BOA

ユニケージ開発手法でクラスタシステムを開発するためのコマンドセットです。
MapReduceなどの特別なプログラミングは不要。クラスタコマンドを使用して通常通りスクリプトを記述するだけで、並列処理が実現します。

統計

usp STAT

ユニケージ開発手法で統計解析処理を行うためのコマンドセットです。
データ分析言語「R」と同様の約80種類の基本的な統計手法をコマンド化。
usp Tukubaiと組み合わせてデータクレンジングからレポーティングの一連の処理を、1つのスクリプトで高速実行できます。

個人向け

Personal Tukubai

個人向けの、非商用・期間限定「usp Tukubai」です。
企業向けのコマンドセット「usp Tukubai」を、個人での独習用に安価でご提供しています。
Windows版、Mac版、Ubuntu版、CentOS版があります。

© usp lab. 2024　Confidential Proprietary

本資料に関するお問い合わせは以下までお願い致します。

koho@usp-lab.com

© usp lab. 2024 Confidential Proprietary