

LINE ではグループに参加できる人数に 200 人という上限を設けることによって、際限なく負荷が増加することを防いでいます。

ラウンドロビンアルゴリズム

## P2P(Peer to Peer)

ドラゴンクエストXでお客様から寄せられるトラブルに対しては、サポートスタッフが対応しています。具体的にはたとえば、不具合でどこかに閉じ込められてしまった場合やアイテムが消失してしまった場合に救済したりといったことです。姿を現す必要があれば「サポートスタッフ」と名乗りゲーム内に登場します。

かつてゲーム業界で使用するのはアセンブリ言語と決まっていたましたが、もう昔の話です。その後 C 言語を経て、今は C++ が主流です。C++ は CPU ネイティブコードで動くオブジェクト指向言語として非常に優れています。

C++ での記述は自由度が高いため、アセンブリ言語や C 言語時代の書き方をしてしまうベテランプログラマーや、変換後の機械語がイメージできず実行効率の悪いコーディングをしでしまう若手プログラマーも想定する必要があります。

紆余曲折ありましたが最終的には、**Oracle Exadata** という高性能のソフトウェアおよびハードウェア製品を導入しました。

**Oracle Exadata** の検索速度だからこそ実現できているところもあり、稼働中に更新や計測の可能な範

囲が広いなど保守性も高く、あらゆる面で高性能なため助けられています。

「サービス」はスクラップ&ビルドを前提として、スピード感や開発体制の確保のしやすさを重要視します。

ゲーム業界ではエンドユーザーのことを「ユーザー」と呼びます。作りきりのゲームは改修できないため、開発側からユーザーへの一方通行しかありません。しかしオンラインゲームはユーザーと双方向になりますので、自分たちは開発業でもありサービス業でもあると考えるようになりました。

筆者はそれまでの一方通行に慣れてしまい、正直なところユーザーに対して上から目線になっていました。同様のドラゴンクエストX開発スタッフも多かったと思います。

そこでドラゴンクエストX開発コアチーム内では「ユーザー」を使用禁止にし、代わりに「お客様」と呼ぶことで、サービス業でもあるとう意識を高めることにしました。これは単なる言葉遊びでしかないかもしれませんが、現在お客様対応を全員が高い意識で行えていることと無関係ではないと思います・。

しかし、サービス開始後に「私たちをお客様扱いするな」という旨のお叱りを受けたことがありました。ドラゴンクエストXのお客様の中には、その世界の「冒険者」として筆者たちサービス提供側とともに盛り上げてくださる人も多く、「これからも協力して盛り上げていきましょう」というたいへんありがたいお叱りでした。それからは筆者も、「お客様」と「冒険者」の両方を使用しています。ドラゴンクエストXはこれからも冒険者のみなさんとともに進化し続けます。

ITにおける技術的な進歩は速く、少し前まではできなかったようなこともハードウェアのスペックが上がって可能になったり、思いつきもしなかったようなソフトウェアによって処理を高速化できたりすることがあります。安定しているからとそのままにするのではなく、最新の技術をいち早く検証、導入

することも**重要**だと考えています。

そもそも動画は、大雑把に噛み砕いて言うと「連続した静止画を順番に表示しながら、それに合わせて音楽を鳴らしたもの」です。連続した静止画を1秒間に何枚切り替えるかが**フレームレート**と呼ばれ、静止画一つ一つの大きさが解像度となり、それらの画質が**ビットレート**となります。

これは無圧縮動画と呼ばれ、いわゆるパラパラマンガと同じですべての動きが情報として存在します。当然ながら最も画質の良い状態となりますが、すべての情報を保持するために情報量が多くなってしまいますので、これをなるべく画質を落とさずに情報量を減らすことが**エンコード**と呼ばれる処理です。

▼表2 主要要件定義書

担当	成果物
Webディレクター	ワイヤーフレーム、サイトマップ
Web開発エンジニア	機能一覧、システムフロー図

キックオフ会議終了後、開発スタートです。先述の通り DMM.com では、プロジェクト承認からサービス公開までの期間が短いため、**兎にも角にも開発スピードが求められます**。常に流行が変わる現在において、アイデアを具体化するスピードはとても重要であり、少し遅れるだけで競合他社に差を付けられると考えているためです。

## デプロイ

~~~~~

デプロイとは、主にネットワークを通じて提供される Web アプリケーションなどのシステム開発工程において、システムを利用可能な状態にすることである。デプロイ (deploy) という言葉は元々、配置する、展開するといった意味の英語である。

~~~~~

冪等性べきとうせい

~~~~~

「冪等性 (べきとうせい)」という概念があります。冪等性とは「何度同じ操作をしたとしても同じ結果を得られる」というものです。例えば「abc というユーザーを指定の属性で作成せよ」という「プログラム」があったとします。この「プログラム」を何回実行したとしても、「abc」というユーザーが指定の属性で作成されているという結果に変わりはないという事です。つまり先の例で言えば、パッケージがインストール済みならば「何もしない」のが望ましい動きになります。

ただ「自動化システム」がサポートしていても、全ての操作で冪等性を保証できる訳ではありません。代表的な例は外部コマンドの実行です。一般的に外部コマンドは何度実行しても常に結果が変わらない事を保証できません。つまり複数回実行すると、「あるべき状態」であったものを、わざわざ「修正を要する状態」に変更してしまう可能性があり、冪等性が保証されていない事になります。

<http://www.intellilink.co.jp/article/column/devops02.html>

~~~~~