失敗から学ぶ
Webアプリケーション性能管理の秘訣

エンピレックス株式会社

Customers. Performance. Loyalty.

エンピレックス社製品の特徴

- エンドユーザの視点での性能を管理
  - Web・電話の技術を使ったユーザ操作を再現させたテスト・監視方式を採用

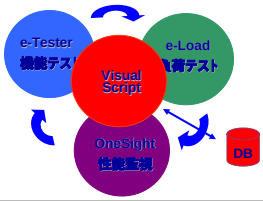

Web技術

Webアプリケーション

音声アプリケーション

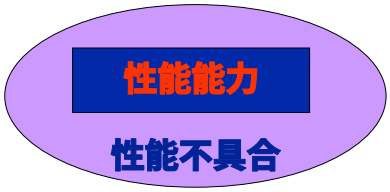
電話技術

ライフサイクル視点で性能改善を実施
- テストスクリプト(資産)を機能テスト・負荷テスト・性能監視で共有が可能

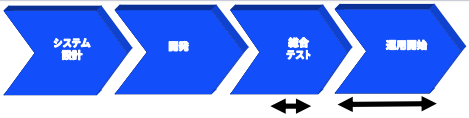


性能問題には2種類ある

- システム設計時に決定する性能能力
- プログラミング時に発生する性能不具合



現在抱えている性能問題



負荷テストの実施　性能問題の発生

- 性能能力が上がらない
- 性能不具合の検出により性能能力を測定できない
- 適切な負荷テストができなかったため、性能不具合を検出できず

システム設計時の性能テスト計画

システム設計

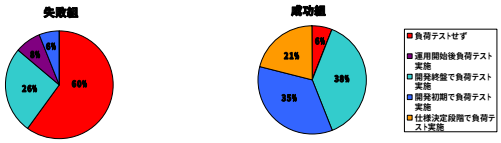
性能テスト計画の概要が決定

- 性能要件のレビューおよび決定
- テストすべき箇所
- 単体レベルでのテスト計画およびその合否
- 負荷テストのスケジュール
- テスト回数
- 性能監視の是非
- 予算

負荷テストの現状

- 現在の日本では、負荷テストが必要にもかかわらずテストが行われていないケースが多数ある

システムリリース後に性能問題があったグループ(失敗)と性能問題がなかったグループ(成功)



* Source: Newport Group, Inc.

負荷テストの必要性

- システム設計時に負荷テストの必要性を考慮する
  - 考慮すべき項目
    - システムを使用する人数
    - 同時アクセス数
    - インターネット OR イントラネット
    - 使用する業務
    - 受託開発 OR パッケージシステム

負荷テストの必要性

- 他の考慮点
  - パッケージシステムの場合
    - カスタマイズ有り無し
    - サイジング情報のありなし
    - USパッケージ商品のサイジング情報には要注意

性能要件のレビューおよび決定

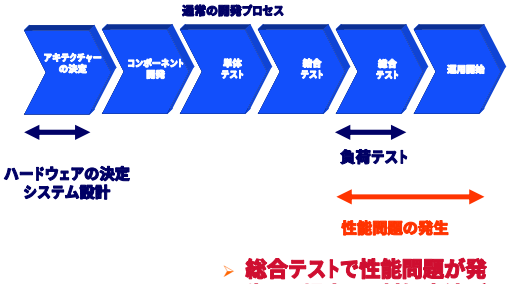
- 本来は、顧客が自ら考え提示するもの
  - 多くの場合は、不適確な要件が多い
    - 例) すべてのユーザーが快適に使用できる
    - 例) すべてのトランザクションが5秒以内に終了する
- 性能要件のレビューを行う
  - 性能要件が適確でないと、不適確な負荷テストとなる
    - 要件を数字化されている
      - 形容詞は不適格
    - 数字は、なるべく平均を使用する
      - 目標をトップ5%のデータにしないため

テストする箇所の決定

- 性能要件のレビューにてテストする箇所を決定する
  - 負荷テストは機能テストとは違う
  - すべてのページに負荷をかける必要はない
  - 性能的に重要なトランザクションをピックアップする

重要)
負荷テストは他のテストに比べ、時間がかかるため、負荷をかけるシナリオは少なめに

単体、結合時のテストの是非



- 総合テストで性能問題が発生した場合は、対処方法が限られる。

テストの合否

- 性能要件とは異なるテストの合否を決定する
  - 開発現場の指標とする
- 性能要件から導く
  - 応答時間 あるページにて平均5秒以内
    - 単体レベル 1秒、 結合レベル 3秒
- 性能要件以外から合否を導く
  - サーバーリソースの合否

単体時における性能テスト

単体
テスト

致命的な性能不具合を解消
性能能力の確認

- プロファイラー等を使用しての性能テスト(1ユーザー)
- 合否判断は、応答時間およびシステムリソースで判断
- 複数ユーザーでのテストを行う
- 外注の場合は、明確な指示を

負荷テストは時間がかかる

- あるチケットサイトでは、計6回の負荷テスト＆チューニングに2ヶ月の期間をかける
- あるクレジット会社では、計8回の負荷テスト＆チューニングに3ヶ月の期間をかける
- あるインターネットショップサイトでは、計3回の負荷テスト＆チューニングに1ヶ月半の期間をかける

テストの回数とスケジュール

- 総合テストにおいては
  - 最低、2回の負荷テストを計画する
  - 期間は、最低2週間
- 単体レベルでの負荷テストが行われている場合のメリット
  - プログラムでのボトルネックはある程度解消されている
  - テストでの資産を再利用できる

負荷テストの予算化

- 負荷テスト全体の工数
  - 最低、0.5人月は抑えよう
- 負荷テストツールの予算化も必要

結合テスト時における負荷テスト詳細計画

結合
テスト

負荷テストの詳細を決定

- テストシナリオの再確認および決定
- 詳細スケジュール
- テスト要員
- 負荷テストツールのハードウェア準備

テストシナリオの決定

- テストシナリオの再確認および再明確化
  - 実行スクリプト
  - 負荷テストのシナリオ(テスト時のスクリプト組み合わせ)
  - 仮想ユーザー数
  - 遅延時間
  - 実行方法
  - 取得データ(サーバーリソースも含め)

詳細スケジュール

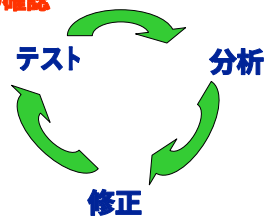
- 負荷テストで考慮すべき工数
  - スクリプト作成 (e-Tester使用)
  - データ準備
  - 負荷テストツールのセットアップ
  - サーバーリソース取得のセットアップ
  - 負荷テスト実施
  - 結果分析
  - テストの回数

上記項目に最低1週間は確保

システムテストにおける負荷テスト

システム
テスト

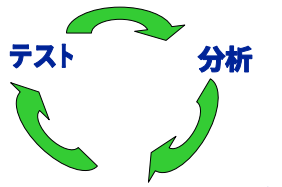
性能不具合の発見および修正
性能能力の確認



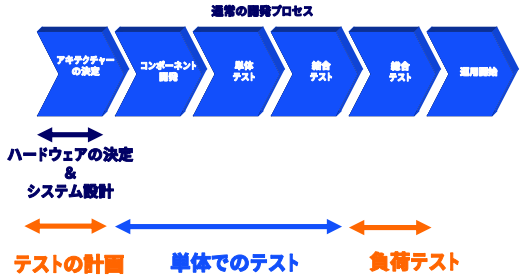
総合テストにおける負荷テスト

総合
テスト

システム性能の最適化



開発工程で性能管理を行う。



運用時の性能監視

運用

性能監視

- 性能不具合の検出
- SLAの維持
- 性能データの蓄積→キャパシティプラン

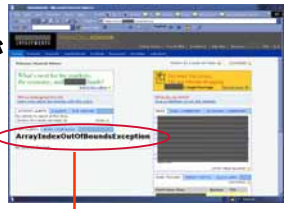
性能不具合の検出

システム監視ソリューションで
問題なし

OK ☑ ネットワーク
OK ☑ サーバー
OK ☑ ルータ & スイッチ

ところが

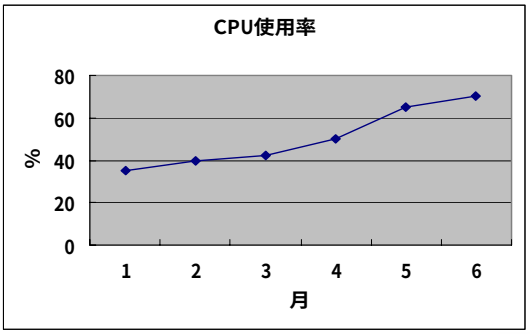
エンドユーザーが体験するのは



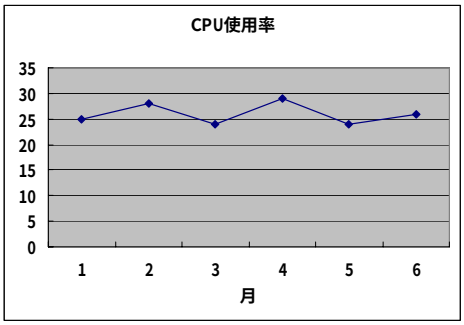
アプリケーション層のエラー

ゲージは全て緑色だが、顧客は大いに不満!

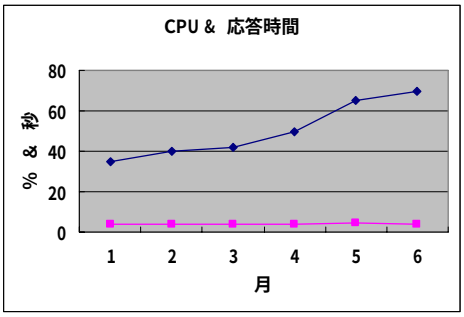
演習



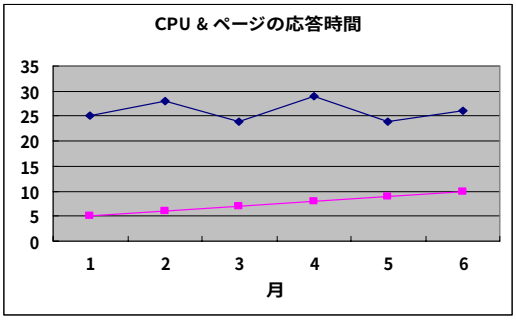
演習



演習



演習



設計時のサイジング

Customers. Performance. Loyalty.

事前にサイジングするためには

- 性能目標が決定している
- 運用時のアクセスパターンを理解している

性能目標が決定している

- 同時ユーザー数
  - メモリー相関
- スループット
  - CPU相関
- ページの応答時間
  - サーバースペック＆台数相関

運用時のアクセスパターンを理解している

- どのページがどれくらいの割合でコールされるか。
  - トランザクション、メソッドレベル

開発を促す性能目標値

- 多くの場合は、事前にサイジングはできません。
  - 通常行っているサイジングとは？
- 性能目標が細部まで決定していると？

システム設計時の性能テスト計画

システム設計

性能テスト計画の概要が決定

- 性能要件のレビューおよび決定
- テストすべき箇所
- 単体レベルでのテスト計画およびその合否
- 負荷テストのスケジュール
- テスト回数
- 予算

上記項目が決定していない時は、サイジングは失敗します

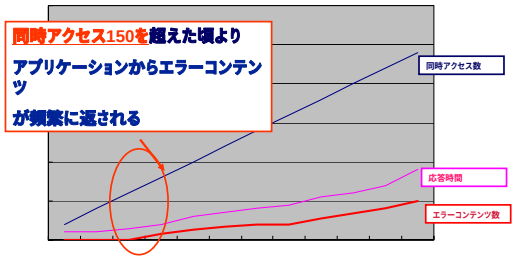
性能設計での失敗事例

Customers. Performance. Loyalty.

性能設計での失敗事例

- 大手エンドユーザーからの要件定義は、人事管理、勤怠管理、決済管理の機能が入ったシステム要件を提示、 中堅システムインテグレーターが受注
- 機能面で開発工程が遅れ、単体、結合レベルで性能に関して気にせずにアプリケーションを構築
- システムテストの際ににエンドユーザーからの依頼で性能要件を満たしているかをエンピレックスでテストを行う

負荷試験結果



- 応答時間は、目標の3倍
- アプリサーバーは負荷途中で停止

テスト後

- アプリサーバーの不具合修正に5ヶ月
- 応答時間の性能は、基本的に修正不可と判断
- Webサーバーを5台から10台へ、APサーバーも5台から10台

Empirix社製品構成