

スキルセットシート

システム運用コンサルティング

株式会社シェイクソウル



深海

Ver.2.2

はじめに

■ ごあいさつ

弊社ShakeSoulはスピード、クオリティ、スキルの3つの要素を重視し、お客様の問題・課題解決し、運用における現場力の向上に努めます。

結果・効果を得るための努力を惜しまず、その成果をアウトプットすることをお約束します。

経験としてISPでのバックボーン運用の実績と meet-me にて短期間でサービスの立ち上げを行っております。meet-me では3ヶ月での運用体制(モニタリング、障害対応体制含む)の構築を行っております。

サービス立ち上げには運用に必要な要素の把握とその実現力が求められます。

これを短期間で行うには要素の把握が明確にできていること、実現力が高いことが必要でした。まさしく現場力が高められる環境で、様々な実践的ノウハウを得ることができました。

お客様のシステム運用がより楽しくなるように、現場からの新しい発想、チャレンジが行えるように経験とノウハウを生かしてお手伝いさせていただきたいと思っております。

■ 内容について

運用コンサルティングの経験してきた項目を概要的に示しております。

また、各フェーズでの手法を説明させていただいております。

最後に運用面における具体的経験例を提示しております。これは概念的にとどまらない実践的な提案させていただくために、まず弊社として経験してきた実践的タスクを示すための資料となります。

■ 対応スピード

短期的なものであれば2週間で把握、3週間目で実装に着手、効果が体感できるのは1ヶ月目くらい

のスピード対応を行えるよう想定しております

(ただし、お客様環境に依存いたしますのであくまで目安となります)

システム運用コンサルティング

■コンサルティング項目

主にシステム運用に関わるコンサルティング項目をご紹介します

経験範囲項目	経験項目例
•業務 分析/設計/実装	<ul style="list-style-type: none">• 各人のタスクレベルと役割把握、業務効率性の分析• 質とスピード向上のための改善ポイント、改善方法の提案• ツール導入、ルール化などの手段の実施、効果測定
•コスト 分析/提案/実装	<ul style="list-style-type: none">• コスト発生項目の洗い出し、コスト変動傾向の把握• 削減コスト算出、コスト削減可能箇所の提示• ベンダ、IDCなどに対するコスト削減交渉の実施
•システムスペック 分析/提案/実装	<ul style="list-style-type: none">• トラフィック分析、ネットワーク・サーバリソース負荷分析• 改善ポイント、改善方法の提案、拡張計画の策定と提案• 拡張作業、リプレイス作業の実施
•運用効率 分析/提案/実装	<ul style="list-style-type: none">• 運用に関わる作業内容の把握、運用フローの把握• 効率化のための改善ポイント、改善方法の提案• 改善運用フローの実施サポート、効果測定
•障害 分析/提案/実装	<ul style="list-style-type: none">• 障害箇所、原因、頻度、内容、傾向の把握• 障害発生率軽減のための改善ポイント、改善方法の提案• 改善作業実施、効果測定

手法の説明

■分析手法

- コスト、リソース、スペック、スキルの多角的観点からバランスをとりつつ分析する
 - ひとつの観点に偏ると相対的なメリットにならない場合もある
 - 例えばコスト分析においてはコスト削減のみ実施した場合、カットした部分のフォローを考えどのようなインパクトが発生するか含めて分析する
- 実際に分析に使用するデータはシステムログ、モニタリング、コストなど数値化されたもの
- 数値化が難しい傾向、スキルに対しては経験とある程度の時間をかけ把握する

■設計手法

- 短期的な観点、中長期的な観点を両方もって行う
- 最終的なあるべき姿を定義し、同意と共有されたゴールを定める
 - そのアプローチとして短期、中長期タスクを位置づける
- 即効性のあるもの、影響範囲の小さなものは即日反映できるようスピードを重視
- 中長期的アプローチはステップを策定し、計画性を持たせる
- 導入のインパクトを緩めスムーズに移行することを重視

■実装アプローチ

- 日常的な小さな改善の積み重ねが最終的に中長期的な改善につながるようタスクを設定する
- 概念的な目標の提示のみは行わない
 - どのようにすればよいか。への答えを具体的に提示する
 - 具体的・技術的な実施方法、アプローチ方法を提示し、実施することを重視
- 進捗を確認しつつ、必要な修正、調整を行う
 - あくまでアプローチはゴールを達成するものと位置づける

具体的経験例

具体的に経験してきた実績例を示させていただきます。

- ◆ 運用
 - ◆ 運用設計
 - ◆ 運用ツール構築
 - ◆ 運用フロー
 - ◆ 運用体制
 - ◆ ToDo管理
 - ◆ リソース管理
- ◆ 障害対応
 - ◆ 障害対応経験
 - ◆ 障害対応フロー
- ◆ 情報共有
 - ◆ プロジェクトメンバー間コミュニケーション/連絡手段
 - ◆ 情報共有ツール構築/管理
 - ◆ ToDo進捗管理ツール管理

運用

■ 運用設計

□ 運用ツール構築

- ◆ リソースモニタリング
 - ◆ ツール構築 : Cacti
 - ◆ モニタリング項目策定
 - ◆ サーバタイプ別にテンプレートを用意、サーバ増設時の効率化を図る
 - ◆ カスタマイズ
 - ◆ トラフィック合算グラフ作り : 本番系サービストラフィックを把握
 - ◆ 閾値以上の場合のアラートメール配信など : トラフィック輻輳を未然に防ぐため
- ◆ 監視
 - ◆ ツール構築 : Nagios
 - ◆ 監視内容/項目策定
 - ◆ サーバの役割によって内容をテンプレート化、作業効率を図る
 - ◆ MSP業者への障害対応マニュアル策定
 - ◆ 独自プロセスの扱い(起動、停止、確認方法などの手順)
 - ◆ 監視項目のアップデート方法ルール化
 - ◆ 共通の管理ファイルを持ち、バージョン管理する
 - ◆ 各メンバとMSP業者との共有方法ルール化
 - ◆ メンテナンス前にMLへ通知など
- ◆ アラートの通知者、非通知者の組織間のすみわけ
 - ◆ 障害対応者、開発者、マネージメント層や役割による情報把握レベルを変えた
 - ◆ 非通知者への報告タイミングと方法の策定 : 障害継続時の経営判断をあおぐ際の仕組み

運用

□運用フロー

- ◆ 日常的な意思決定フロー策定
 - ◆ オペレータの権限を決め、それ以上の判断はチームリーダにあおぐなど
- ◆ 日常的なルーチンワーク内容策定
 - ◆ ルーチンワーク実施ルール決め
 - ◆ 毎朝10時に行うことなど：前回からの経過時間を一定にするため
 - ◆ 報告フロー決め
 - ◆ ルーチンワークの詳細手順と判断基準の明示

運用

■ 運用体制

- ◆ 必要な人員数の算出
- ◆ 各人のスキルセットを考慮したペア/チーム構成決め
 - ◆ 各人の経験値、スキルの見極め
 - ◆ スキルに適したタスクアサイン
 - ◆ 障害対応を想定したチームづくり
- ◆ 事象ケースに応じた報告フローの策定
 - ◆ 役割による判断権限を守るため
 - ◆ 設計変更が生じるような作業内容はチームリーダーとディスカッションするなど
- ◆ MSP業者などのリソース活用判断
 - ◆ 人的リソースがまかなえる業務範囲の見極め

■ ToDo管理

- ◆ 日常的な業務ToDo管理
 - ◆ tracを用いたチケットングティングシステムを利用
 - ◆ trac 構築
 - ◆ プラグイン設定、ユーザ管理(権限含む)を行う
 - ◆ 各チーム(web, ツールアプリ開発など)ごとにtrac構築実施
 - ◆ ToDo報告方法のルール化
 - ◆ trac 上で進捗状況をログとして残す
 - ◆ trac を見ればすべての ToDo の進捗状況が分かるように
 - ◆ メンバ全員が回覧できる状況にする
 - ◆ クリティカルな ToDo 終了時には一度アサイン担当者(チームリーダー)へアサインを変えて判断をあおぐなど

運用

■ リソース管理

- ◆ インフラ系リソースの管理方法策定
 - ◆ ネットワーク機器：スイッチならばポートアサイン、vlan 管理など
 - ◆ サーバ：ホスト名など
 - ◆ アプリケーション
 - ◆ IPアドレス
 - ◆ IDC内ラック：ラック配置情報、電源供給先機器など
- ◆ 各機器における設定ファイルの管理
 - ◆ バージョン管理：subversionを使用
 - ◆ 管理方法のルール化：必ず変更点をコメントに残すことなど
- ◆ 管理ファイルの作成
- ◆ 管理ファイルアップデート方法ルール化

障害対応

■ 障害対応経験

- ◆ サーバ障害
 - ◆ ハード障害
 - ◆ ブレード障害 : ファームウェア、モジュールの不具合など
 - ◆ OS 不具合 : ハングアップなど
 - ◆ Apache, MySQL まわり
 - ◆ 独自ゲームサーバ不具合動作
 - ◆ バグによる不具合動作対応
 - ◆ 独自ゲームプロセスの挙動とlogファイルの把握を事前に行っている
- ◆ ISP におけるインターネットバックボーン障害対応
 - ◆ BGPオペレーション 経路広報に関する不具合対応
 - ◆ 自AS、他AS 依存
 - ◆ ルーティング障害
 - ◆ レイヤー1,2,3 における様々な障害
 - ◆ ルータ/スイッチなどの不具合動作、ハード障害対応

障害対応

■ 障害対応フロー

- ◆ フロー設計
 - ◆ MSP業者との役割分担とエスカレーションしてもらうタイミング、判断基準などを決める
- ◆ 障害対応ポリシー策定
 - ◆ ダブルオペレーションはしないなど
- ◆ 障害レベルによる報告フロー策定
 - ◆ 開発者への連絡、経営への報告のタイミング策定など
- ◆ レポートテンプレート作成
 - ◆ 障害報告書
 - ◆ 一次報告：発生後
 - ◆ 途中経過報告：発生後30分以上継続の場合など
 - ◆ 最終報告書：原因究明後
- ◆ レポーティングルールの策定
- ◆ 一次報告、途中経過報告、最終報告を提出するタイミングを決める

情報共有

■プロジェクトメンバー間コミュニケーション/連絡手段

□情報共有ツール構築、管理

- ◆ メーリングリストメンバー設計、管理
 - ◆ プロジェクト全体のメーリングリスト管理
 - ◆ 必要なメーリングリストの策定、メンバー選定
- ◆ wiki 構築
- ◆ wiki 運用ルール策定
 - ◆ ページごとのアクセス権設定
 - ◆ ページ構成を考慮した記述ルール策定

□ToDo進捗管理ツール管理

- ◆ trac 構築
 - ◆ プラグイン導入、ユーザ管理(権限含む)を行う
 - ◆ メールでのノーティファイ、稼働時間管理機能を追加
 - ◆ 各チーム(web, ツールアプリ開発など)ごとに trac 構築、管理
- ◆ trac 運用ルール策定
 - ◆ 報告フロー、チケットアサイン変更のタイミングなど策定
- ◆ 日常業務 メンバスキルに応じたチケットアサイン
- ◆ 日常業務 チケット進捗状況チェック