

目次

監訳者まえがき

はじめに vii

第1章 イン트라ネットの衝撃 1

1.1 概要 2

1.2 イン트라ネットとは何か 2

根幹はクライアント/サーバー 5

イン트라ネットの土台 5

イン트라ネットのアプリケーション 8

1.3 なぜイン트라ネットを構築するのか 13

コミュニケーション 14

コラボレーション（協働） 14

知識管理 17

イン트라ネットの利点 18

1.4 イン트라ネットのコストについて 21

1.5 イン트라ネットの構築をいつ開始すべきか 22

初期の採用者 23

主流期の購入者 24

後期の採用者 24

第2章 12ステップのプログラム 25

2.1 概要 26

2.2 基本ステップの判断ツリー 26

意思決定支援ツール 28

2.3 各ステップの概要 28

ステップ1：導入の適否 28

ステップ2：手助けが必要ですか？ 30

ステップ3：ユーザーからの支援をリストアップ 31

ステップ4：組織の諸問題 33

ステップ5：技術的なアーキテクチャの決定 33

ステップ6：アプリケーションとプロトタイプの決定 34

ステップ7：登録およびIPアドレス指定 34

ステップ8：最終プラン作成 35

ステップ9：アーキテクチャの導入 36

ステップ10：アプリケーションの開発 36

ステップ11：イン트라ネットの方針および手続きの作成 37

ステップ12：イン트라ネットの管理 38

2.4 すべて失敗に終わった場合 38

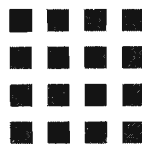
第3章 最初のステップ 41

3.1 概要 42

3.2 決定を行う 42

3.3 ステップ1：イン트라ネット導入の適否 42

ビジネスを評価する 43



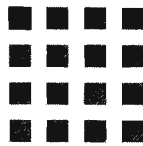
情報の必要性の評価	44
I-TIPS	46
3.4 ステップ2：手助けが必要ですか	46
技能査定	47
技能の取得	48
外部リソース	50
パートナーとの共同作業対外注	51
3.5 ステップ3：ユーザーからの支援を得る	53
経営幹部の支援を得る	54
草の根の支援	55
3.6 アドレスとドメイン名の取得	59
IPアドレスの登録	59

第4章 組織上の問題 **61**

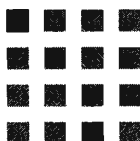
4.1 概要	62
4.2 企業構造を進化させる	62
4.3 文化的な問題	63
敏捷性	63
Just-In-Time文化	64
開放性とセキュリティ	64
変化の管理	65
イントラネット利用規定の役割	66
4.4 イン트라ネット導入プロジェクトの組織化	68
イントラネットのリソースと開発の差異	68
導入プロジェクトの管理	71
4.5 イン트라ネットとIS組織構造との統合	73
Web情報の枠組み	73
イントラネットの組織構造	75
4.6 法的な問題	78
情報アクセスおよび発行方針	79
電子メール	79
知的財産権	79
ドメイン名	80
商標	81
著作権	81
業務上の機密	81
物理的資産および知的財産の保護	81
まとめ	82

第5章 イン트라ネットのアーキテクチャ **83**

5.1 概要	84
5.2 Webを基本とする システムの各要素	84
抽象化	86
5.3 Webの枠組み	88
5.4 一般的なアーキテクチャ	89
5.5 アーキテクチャの選択	91
5.6 構成要素の選択	92



表現	92
クライアントブラウザ	93
ネットワークコミュニケーションサービス	93
Webサーバー	94
サーバーアプリケーションインターフェイス	96
アプリケーション	96
サービスインターフェイス	96
業務サービス	96
オペレーティングシステムサービス	97
ハードウェア	98
5.7 サーバーの基本構成要素	100
URL (Universal Resource Locator)	100
サーバーソフトウェアの構成要素	102
5.8 サーバーの論理的な構成	105
スタンドアロンシステム	105
シングルサーバーシステム	107
マルチサーバーシステム	107
分散型システム	110
5.9 メインフレームの統合	110
3270 データストリーム (スクリーンスクレーピング)	111
RPC (Remote Procedure Call) アーキテクチャ	111
メッセージアーキテクチャ	111
TPモニター (Transaction Processing Monitor)	112
ORB (Object Request Broker)	113
発行/購読のアーキテクチャ	113
サービスブローカー	113
第6章 オブジェクトリクエストブローカー	115
6.1 概要	116
6.2 オブジェクト技術とは何か	116
6.3 ORBの定義	116
ORBの標準規格	117
採用の遅れ	119
現在の人気度	120
6.4 分散オブジェクト技術とWeb	120
6.5 分散オブジェクト技術とWebに関連する問題	121
6.6 3つの主要な方式	121
6.7 相互運用性	123
6.8 選択する	123
6.9 将来の傾向	124
第7章 イン트라ネットの設計	125
7.1 概要	126
7.2 イン트라ネットの種類	126
通信イン트라ネット	126
統合イン트라ネット	126
カタログイン트라ネット	127



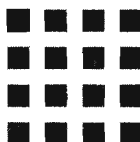
単一ログインイントラネット	127
7.3 パフォーマンス	127
7.4 コスト	128
7.5 セキュリティ	128
おもな脅威	130
主要なニーズ	134
セキュリティプロトコル	135
セキュリティのトレンド	136
7.6 管理のしやすさ	137
命名規則	137
役割の明確化	137
標準規則	138
手続き	138
文書化	138
スタイルガイド	138
サイト管理ツール	141
データの集中管理	141
マルチメディア製作	141
7.7 使い勝手	141
7.8 トレードオフ	143

第8章 イン트라ネットの開発 **145**

8.1 概要	146
8.2 従来の開発方法との違い	146
短期開発、段階的な改良、早期のプロトタイプ作成	147
エンドユーザーの参加	147
場所に依存しない	148
方法論	149
8.3 ステップ9：アーキテクチャの実装	151
通信インフラストラクチャ	151
アーキテクチャのインフラストラクチャ	157
8.4 ステップ10：アプリケーションの開発	159
プロトタイプを選択	159
アプリケーションの対象範囲	161
機能仕様	161
リソースの計画	162
スケジュール	163
標準規則	163
反復	163
テスト	164
ユーザー受入れテスト	164

第9章 イン트라ネットの各種ツール **165**

9.1 概要	166
9.2 ツールの選択	166
ツールを使うための技術	168



技術とツールの評価	168
ベンダー	170
オープンツールかメーカー固有のツールか	171
フリーソフトか市販品か	172
アプリケーションツールのモデル	172
9.3 ブラウザーとサーバーソフトウェア	176
ブラウザー	176
サーバーソフトウェア	179
9.4 おもなツールの種類	179
開発ツール	179
プログラミング言語	180
検索エンジン	183
クライアントツール	185
管理および運営ツール	187
プラグイン	190

第10章 ユーザーインターフェイス 193

10.1 概要	194
10.2 デザイン環境と目標について	194
10.3 ユーザーインターフェイスデザインの基礎	195
一貫性	195
単純化と直感性	196
利用者の意識	197
美的な完成度	198
10.4 全体のルック&フィール	199
背景	199
配色	201
インターフェイスのメタファー	202
サイトの構造	202
10.5 ページ要素	206
グラフィックヘッダー	207
本文	207
ナビゲーションインターフェイス	214
情報フッター	221
10.6 グラフィックのヒント	222
グラフィックファイルのフォーマット	222
グラフィックファイルのサイズ	224
ディザリング	224
10.7 規格決定のタイミングと方法	225

第11章 バックエンドアプリケーション 227

11.1 概要	228
11.2 外部へのアクセス	228
11.3 グループウェア	229
Notesベースのグループウェア	230
Webベースのグループウェア	232
11.4 データ読み出し	234



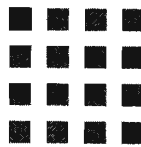
11.5	データベースへの接続	235
	サードパーティ製の接続ツール	235
	データベースベンダーの拡張機能	235
	データベース	236
11.6	ドキュメント読み出し	238
	構造化されていないデータ	238
	ハイエンドの画像管理/ドキュメント管理システム	239
11.7	企業間の通信/ Electronic Data Interchange (EDI)	239
11.8	企業と消費者の間の通信/電子商取引	239
11.9	セキュリティ	241
11.10	過去と未来	241

第12章 イン트라ネットの実現 243

12.1	概要	244
12.2	テストのガイドライン	244
	ステートとコンテキスト	246
	単純なイントラネットアプリケーション	246
	簡単なフォームを利用したイントラネットアプリケーション	248
	複雑なイントラネットアプリケーション	249
12.3	運用	250
	クライアント管理	251
	インフラの管理	252
12.4	ヘルプシステムのコセプ	255
	ハイパーテキストによる索引	255
	ヘルプデスクアプリケーションのサポート	255
	検索能力	255
12.5	アプリケーションの移行	256
12.6	データ変換	256
12.7	イントラネットのインターネットへの接続	257
	ファイアーウォール	257
	サーバーのセキュリティ	258
	トラッキング	258
12.8	パフォーマンスについて	259
	負荷とボリュームについて	260
	不測の事態への備え	260
	まとめ	261

第13章 イン트라ネットの管理 263

13.1	概要	264
13.2	そこはジャングル	264
13.3	イントラネット管理の枠組み	266
13.4	イントラネットシステムの管理	269
	推奨する方針と手順	269
13.5	イントラネット特有の問題	273
	プランニング	273
	運用	274
	テクニカルサポート: ヘルプデスクの改良	278



13.6	イントラネット管理の自動化	280
	Tivoli Systems	280
	Hewlett-Packard	282
	Computer Associates	282

第14章 ヒント、技、落とし穴 285

14.1	概要	286
14.2	計画性の欠如	286
	落とし穴#1：長期計画の欠如	286
	落とし穴#2：説得力の欠如	287
	落とし穴#3：利用度の過小評価	288
	落とし穴#4a：複雑さの過小評価	288
	落とし穴#4b：不適当なアーキテクチャの採用	289
	落とし穴#5：非グラフィカルブラウザの使用	290
	落とし穴#6：ハードコードによる「メンテナンスお化け」	290
	落とし穴#7：流行マニア	291
	落とし穴#8：プロトタイプの手短	291
	落とし穴#9：リソースの補強	292
14.3	専門家による支援の欠如	292
	落とし穴#10：専門知識の欠如	293
	落とし穴#11：古いアプリケーションの取り込み	293
	落とし穴#12：スキルを融合すること	294
	落とし穴#13：デザインの失敗	295
	落とし穴#14：先進機能の濫用	295
	落とし穴#15：法律違反	296
	落とし穴#16：データベース変換プロジェクトでのセキュリティ不足	296
14.4	管理の枠組みの欠如	297
	落とし穴#17：バージョンの整合	297
	落とし穴#18：テスト不足	298
	落とし穴#19：古い	298
	落とし穴#20a：技術の袋小路	299
	落とし穴#20b：成長に対応しきれない	299
14.5	アプリケーション開発上の落とし穴	300
	落とし穴#21：パフォーマンス	300
	落とし穴#22：分散開発	301
	落とし穴#23：中枢機構の作り直し	301
	落とし穴#24：危険な綱渡り	302
	落とし穴#25：クライアントマシンの品質	302
	落とし穴#26a：CGIスクリプト対API呼び出し	303
	落とし穴#26b：開発しやすさよりも使いやすさ	303
	落とし穴#26c：コンテンツ制作とプログラミングの融合	304
付録A	変化による影響の判定	305
付録B	イントラネットプロジェクト作業計画	313
付録C	インターネットサービスプロバイダー	319
付録D	Request for Comment (RFC) リスト	323

