

【システム設計書の書き方セミナー】

～第5回 運用コストで差がつく！
設計書の正しい書き方と設計・テストの生産性向上～

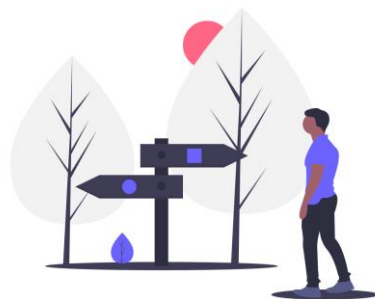
セッション 1

「非機能要件定義書と運用設計書の書き方」



2021年11月24日（水）

10:00～10:40



株式会社システムインテグレータ
代表取締役社長 梅田弘之

社名	株式会社システムインテグレータ
設立/代表	1995年3月14日 代表取締役社長：梅田弘之
所在地	本社：埼玉県さいたま市中央区新都心11-2 東京営業所：東京都渋谷区恵比寿南2-1-10 大阪支社：大阪府大阪市中央区平野町3-6-1 名古屋営業所：名古屋市中村区名駅二丁目41-10 福岡支社準備室：福岡市博多区博多駅前3-4-25
資本金	3億6,771万円
取引所	東京証券取引所第一部（3826）
社員数	231名（役員9名含む）2021年11月1日 現在



せっかくなので、
オンライン名刺交換
しましょう。



SI Object Browser 【データベース開発支援ツール】

- ・GUI操作でオブジェクトを簡単作成
- ・国内主要データベースに対応



GRANDIT 【完全統合型Web-ERP】

- ・日本の企業文化に適合した完全Web-ERP
- ・多彩なアドオン・テンプレートを提供



AISIV-AD 【AIを使った異常検知】

- ・AI画像認識による製造業向け異常検知
- ・エッジコンピュータ、AIクラウドを活用

OBPM Neo 【統合型プロジェクト管理ツール】

- ・プロジェクト管理の標準「PMBOK」に準拠
- ・工数・原価、進捗、要員、品質管理を統合



SI Web Shopping 【ECサイト構築パッケージ】

- ・日本発のECサイト構築パッケージ
- ・大規模EC/ソース公開/グローバル対応



VOICE TICKETS 【カスタマーサクセス支援】

- ・利用者の声(VoC)を直接聴いてチケット管理
- ・利用者と双方向コミュニケーション

SI Object Browser Designer 【ソフト開発のCAD】

- ・設計データをデータベース化
- ・豊富な設計書アウトプット



<TOPS/C> 【プログラミングスキル判定サービス】

- ・エンジニア採用や社員プログラミング教育
- ・プログラミングスキルをリアルタイム判定



IDEA GARDEN 【アイデア創出サービス】

- ・社員発想のアイデアが湧き出す、育てられる
- ・発想会(Gardening)で利用できるツール

設計、開発、プロジェクト管理などソフトウェア開発全般の合理化、生産性向上を実現するツール群



どんな業界でも合理化しないと生き残れません

1. 非機能要件の定義



非機能要件とは

非機能要件(NFR: non-functional requirement)

ソフトウェア設計のうち機能面以外の要件すべてを指す

カバー範囲が広い

(例) 非機能要件が悪くて起きる問題

- ✓ パフォーマンスが悪くて作業効率が悪化した
- ✓ セキュリティに欠陥があり情報漏洩した
- ✓ デザインが悪くて売れない

- ✓ 使い勝手が悪くユーザーから不満が出ている
- ✓ システムが不安定で時々落ちる
- ✓ 拡張性がなくて我慢して使っている

非機能要件の定義項目

JUAS
「非機能要求仕様定義ガイドライン2008」

機能性

効率性

信頼性

使用性

保守性

運用性

移植性

障害抑制性

効果性

技術要件

IPA
「非機能要求グレード2018」

性能・拡張性

セキュリティ

運用・保守性

移行性

可用性

システム環境・
エコロジー

「令和時代の非機能要件」

パフォーマンス

スケーラビリティ

セキュリティ

ユーザビリティ

メンテナンス性

運用性

データ移行

アベイラビリティ

効果目標

技術・環境要件

規制要件

[IPAのホームページからDL可能](#)

非機能要件チェックリスト(1/2)

項目	ファクター	要件	確認・検証方法
アベイラビリティ	<ul style="list-style-type: none"> 稼働率 ダウンタイム RPO/RTO ディザスタリカバリー 	<ul style="list-style-type: none"> 24時間365日稼働。メンテナンスの際も無停止。 完全冗長化によりダウンタイム無くフェールオーバーする RPO6時間、RTO30分（目標復旧時点/目標復旧時間） ディザスタリカバリー対策も講ずる 	対障害性テストを実施する(サーバーやネットワークなど個別に実施)
パフォーマンス	<ul style="list-style-type: none"> 同時アクセス ピーク時 要求速度 	<ul style="list-style-type: none"> 同時アクセス数 1日のピークタイムで1万人/分 バーゲン時などでその5倍 要求速度 フロント：検索1秒、更新2秒以内 バック：検索、更新とも3秒以内 	負荷テストツールで負荷をかける
スケーラビリティ	<ul style="list-style-type: none"> ユーザー数 商品点数 データ容量 	<ul style="list-style-type: none"> 初年度データ容量 ユーザー数10万人、商品点数1万、1日当たり2000人、商品3点購入 年間20%増加で5年間スケールできること 	ダミーデータを作成し、負荷テストツールで負荷をかけた状態で各機能の処理速度計測
セキュリティ	<ul style="list-style-type: none"> 不正アクセス データ盗聴 DoS攻 監査証跡 	<ul style="list-style-type: none"> 管理ページにアクセスできるIPアドレスを制限し、個人情報にアクセスできるユーザーを制限する。 OSやミドルウェアなどの既知の脆弱性はすべて対策する。 ネットワークの侵入検知・防止システム(IPS/IDS)とWAFを導入 セキュリティポリシーを明確にして、担当者に徹底する なりすまし防止のために多要素認証を導入する ユーザーおよび管理者の操作はすべて監査証跡に記録する 	セキュリティ診断ツールでテストする
メンテナンス性	<ul style="list-style-type: none"> 機能追加、変更 バックアップ 	<ul style="list-style-type: none"> 機能変更、追加の際にシステムを停止しない。 バックアップの際にもフロントのパフォーマンス条件を維持 	ダミーでプログラムの追加・変更の実施

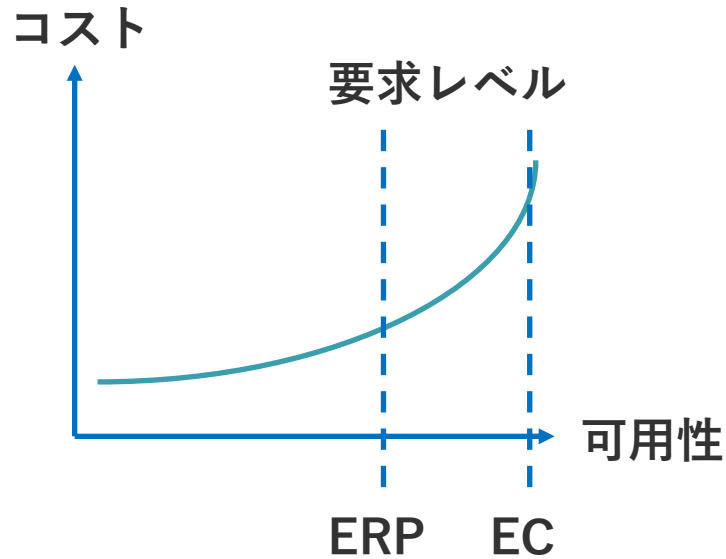
eコマース(オンラインショッピング)の例

非機能要件チェックリスト(2/2)

項目	ファクター	要件	確認・検証方法
運用性	<ul style="list-style-type: none"> ・運用監視 ・出品物監視 ・不正書込み ・バッチ処理 	<ul style="list-style-type: none"> ・正常稼働を自動監視(異常時アラートメール) ・バックアップや更新などの処理を自動化 	異常発生テスト
データ移行	<ul style="list-style-type: none"> ・移行対象データ ・役割分担 ・移行方法、プログラム 	<ul style="list-style-type: none"> ・顧客、商品、購入履歴はデータ移行対象 ・既存データをCSV出力するので、新システムのフォーマットに合わせて取り込む。 	データ移行テスト
ユーザビリティ	<ul style="list-style-type: none"> ・ペルソナ ・UIガイド 	<ul style="list-style-type: none"> ・メインターゲットは20～30代女性でパソコンよりもスマホの購入が多い(7割) ・ユーザビリティ基準を満たすこと 	ターゲットユーザーによるユーザビリティテストを実施する
効果目標	<ul style="list-style-type: none"> ・システムの目的 ・期待する導入効果 	<ul style="list-style-type: none"> ・パフォーマンスやUIを改善することで、売上を現状の月商2億円から3億円にUPする ・リピート率を現状の30%から40%に上げる。 	セッション数やコンバージョン率など、データ分析により検証
技術・環境要件	<ul style="list-style-type: none"> ・プラットフォーム ・採用する技術 ・開発言語 ・RDBMS 	<ul style="list-style-type: none"> ・セキュリティ、アベイラビリティ、スケーラビリティなどの点よりクラウドベースとする。 	
規制要件	<ul style="list-style-type: none"> ・適用される法律 ・知的財産権の抵触 ・輸出禁止技術 	<ul style="list-style-type: none"> ・特定商取引に関する法律 ・個人情報保護法 	受入テストで確認

非機能要件項目

◆アベイラビリティ（可用性）



（よくあるトラブル）
可用性の要求レベルを示していない

（例えば）

EC：24時間365日稼働が減速
完全冗長化構成
障害発生時のダウンタイムも最小

ERP：稼働日・稼働時間のみでもOK
障害発生時はバックアップから戻してもOK

◆パフォーマンスとスケーラビリティ（拡張性）

（よくあるトラブル）

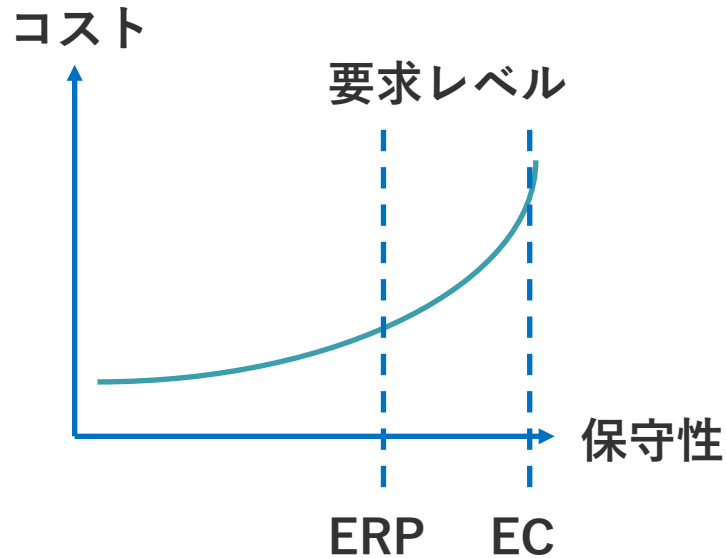
- ・本番稼働→重くて実用に耐えられずにクレーム（東京オリンピックのチケット）
- ・稼働当初はOK→稼働3か月でシステムが遅くなり効率悪化

（原因）

データ量やユーザー数がどれくらい増加するか示していない ➡ 確認・検証方法も定義

非機能要件項目

◆メンテナンス性（保守性）と運用性



（よくあるトラブル）
保守性や運用性の要求レベルを示していない

（例えば）
EC：1年に8時間だけメンテナンスのために停止可能
そのほかはホットスタンバイ
さらに死活監視により24時間監視

◆ユーザビリティ（操作性）

（よくあるトラブル）

- ・ユーザビリティが悪く、客足が遠のいてしまった・・・ECはUIは死活問題
- ・ユーザビリティが悪く、エフォートレスなサービスを提供できず、解約率が高くなった

（原因）

「20～30代のスマホで購入する女性が多い」などのペルソナを示していない
ユーザビリティに対して言及しておらずベンダまかせ、チェックもおろそかだった

◆効果目標

投資対効果…（通常）委託者側の責任

ただし…期待する効果を提示することは効果がある（定量効果/定性効果）
（例えば）お役所のサービスで「高齢者でも迷わずに利用できる」

◆技術・環境要件と規制要件

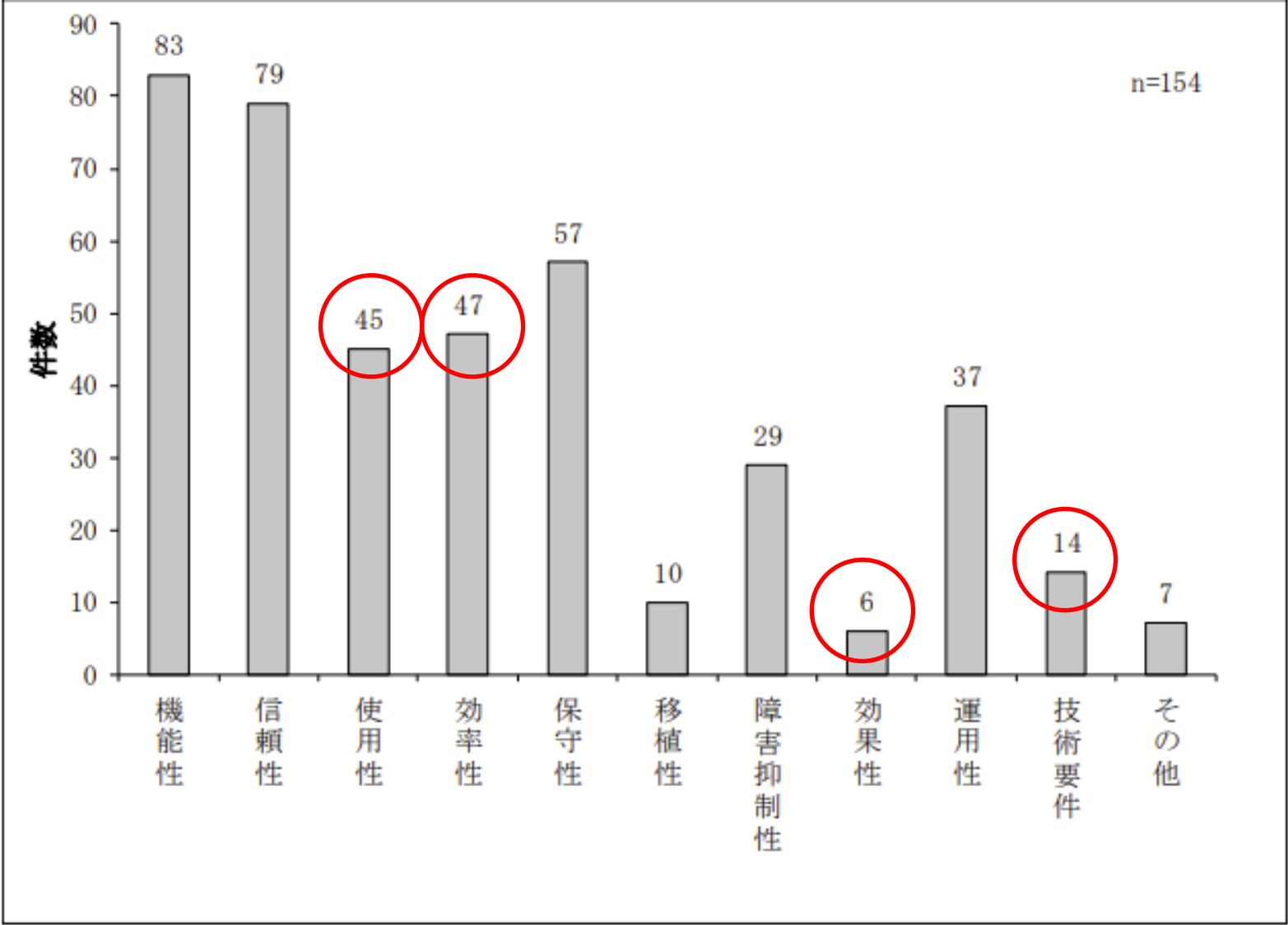
✓技術・環境要件は通常書かれている？

- ・クラウドかオンプレミスか
- ・利用したいツール
- ・既存システムとの関連
- ・RDBMS
- ・開発言語など

✓規制要件は、通常は書かなくても大丈夫？

- ・特定商取引に関する法律
- ・個人情報保護法

非機能要件の重視項目



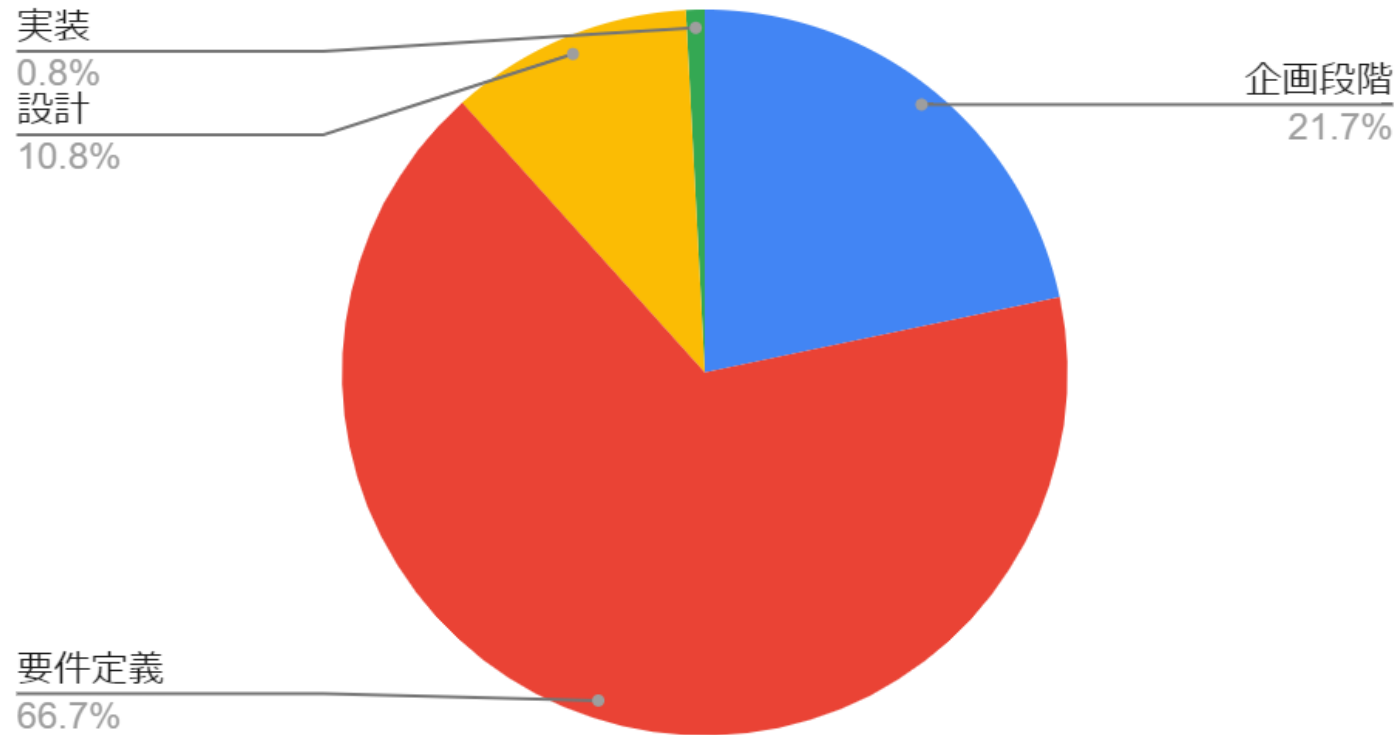
出典：JUAS ソフトウェアメトリックス調査2020年版【システム開発・保守調査報告書】

非機能要件の提示時期

RFP提示段階が理想だが・・・

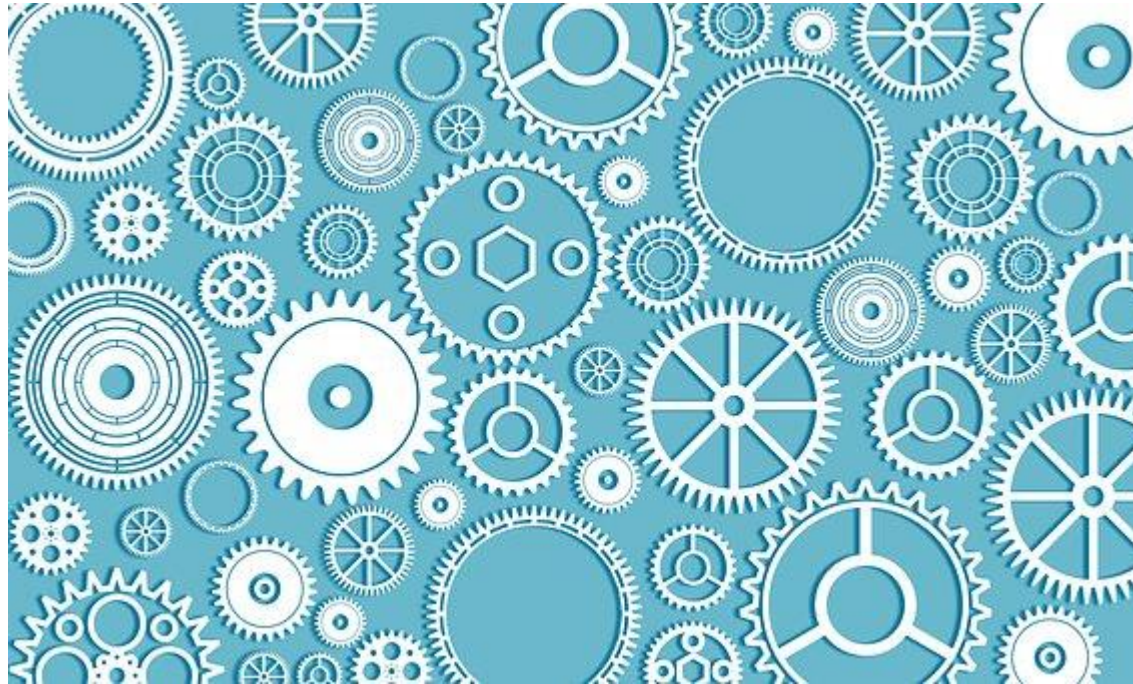
発注者の意識改革

件数 n=120



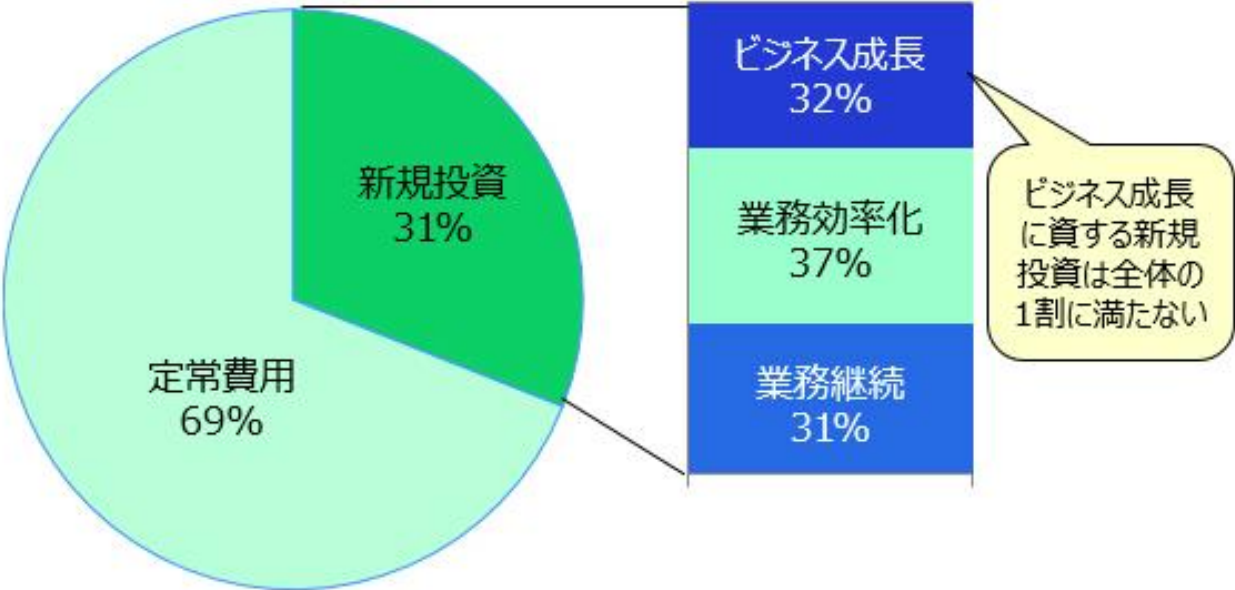
出典：JUAS ソフトウェアメトリックス調査2020年版【システム開発・保守調査報告書】

2. 運用設計書



前回のセミナーの質問：IT関連投資において新規開発と運用保守はどちらが大きい？

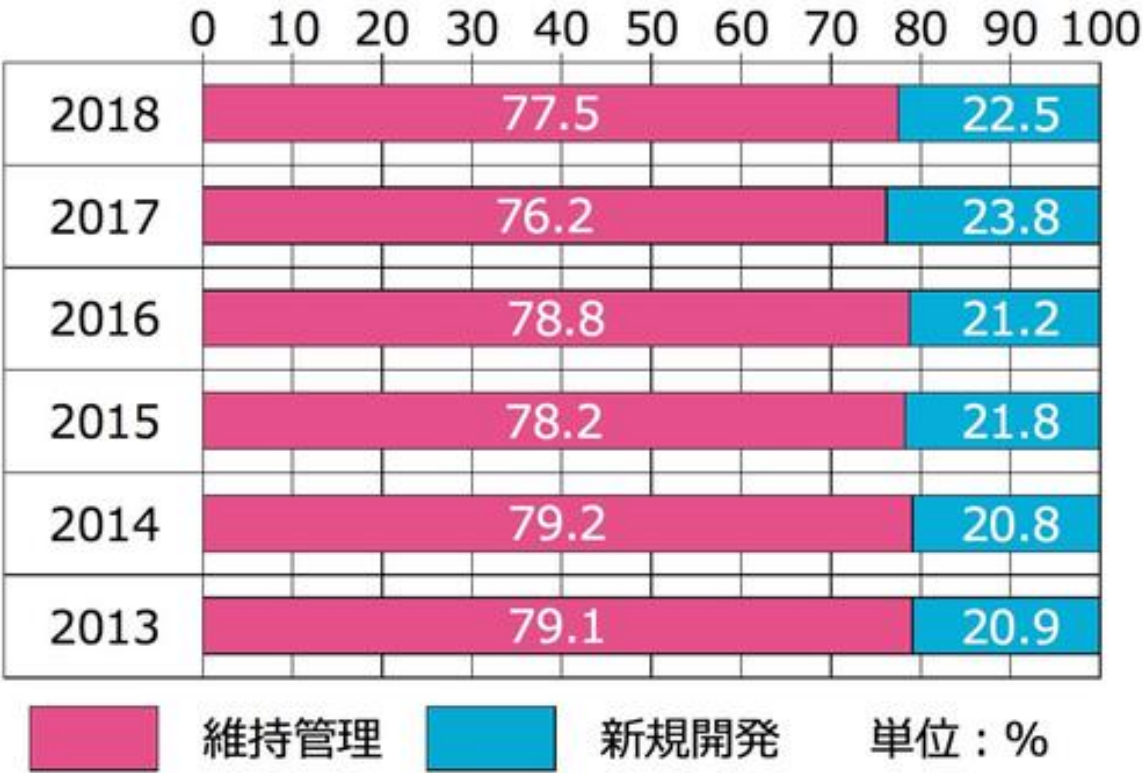




出典：ITR「IT投資動向調査2017」（回答数：2,685件）

<https://diamond.jp/articles/-/141935>

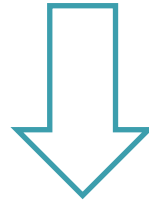
IT投資に占めるビジネス成長に資する投資
（DIAMOND onlineの記事）



<https://it.impressbm.co.jp/articles/-/18036>

ユーザー企業のIT予算の配分比率（IT Leadersの記事）

【目的】 システムを安定稼働して利用者が使える状態を保つ



通常業務を円滑に進める

システム障害を未然に防ぐ

障害を迅速に復旧する

定義内容

- ✓ 運用ルール
- ✓ 運用の仕組み
- ✓ 運用手順
- ✓ 運用体制



(1) 運用の基本方針**(2) システムの業務概要、特性**

- ・システムの概要、特性
- ・システムの利用者

(3) システム構成

- ・システム構成図
- ・ネットワーク構成図
- ・各設備、機器の仕様、仮想化、冗長化
- ・ツールやソフトウェア、ライセンス情報

(4) 非機能要件

- ・パフォーマンス
- ・拡張性
- ・セキュリティ
- ・可用性

(5) 運用体制

- ・運用体制図
- ・ヘルプデスク
- ・緊急連絡先

(6) セキュリティ管理

- ・データ漏洩
- ・アクセスコントロール
- ・VPN、IP制限、端末制限
- ・ウィルス対策、セキュリティパッチ

(7) 運用スケジュール

- ・定期作業項目、随意作業項目の洗い出し
- ・日次、週次、月次、年次の作業予定

(8) 稼働監視

- ・監視項目、監視ツール、監視方式
- ・監視対象ログ
- ・外形監視

(9) 障害対応（可用性管理）

- ・障害の想定と防止策
- ・冗長化
- ・障害発生時の手順
- ・障害管理/インシデント管理
- ・障害分析、KPI管理

(10) 災害対応

- ・災害の想定と対策

(11) バックアップ

- ・バックアップツール、方式
- ・バックアップ対象データ、容量
- ・スケジュール

(12) ジョブ管理

- ・ジョブ管理ツール
- ・ジョブ一覧
- ・ジョブフロー

【ドキュメントの意義】 他のメンバーや将来加わるメンバー、将来の自分と情報を共有すること
運用方針、運用の目的、システム停止の影響の大きさなど、同じ認識を持って運用を行う

(e コマースの例)

【システム特性の共有】

- ✓ 24時間365日稼働なので停止させない運用となる
- ✓ 売上UPのためにシステムの改良が頻繁
- ✓ 表示速度が非常に重要でパフォーマンス低下に要注意
- ✓ サイトが2時間停止した際の販売機会損失や信用失墜の影響度も明記



【運用の基本方針】

- ✓ 冗長化によりダウンタイムを最小限にする
- ✓ DevOpsにより開発から運用までの自動化を徹底する
- ✓ 外形監視によりシステムのパフォーマンスを常時チェックする

サービス提供側の責任範囲

SLA(Service Level Agreement)

OLA(Operation Level Agreement)



- ✓ サービス提供の範囲
- ✓ サービス時間
- ✓ 可用性 など

ITML(IT Infrastructure Library)

1989年に英国が発行したIT「サービスマネジメント(ITSM)のガイドライン、最新版は2019年のITML4

サービスライフサイクル

サービス戦略

サービスの設計

サービスの移行

サービスの運用

継続的なサービスの改善

内容

サービスカタログ管理

サービス資産構成管理

可用性管理

インシデント管理

キャパシティ管理

問題管理

ITサービス継続性管理

サービスデスク

サービスレベル管理

IT運用管理

...

システム構成図

機器やネットワークが運用管理対象…通常のシステム構成図よりブレイクダウン

- ✓ Webサーバー／APサーバー／DBサーバー／ロードバランサー
- ✓ RDBMS/ストレージ
- ✓ 仮想サーバー／仮想ネットワーク／HUB/ルーター
- ✓ OS／監視ソフト／バックアップツール／ファイアウォール

IT資産管理ツール

ブレイクダウンするほど構成図のメンテナンスが大変



IT資産管理ツール活用

IaC (Infrastructure as Code)

インフラの設定手順をコード化…インストールやOSの設定、ネットワーク設定などを自動化



IaCツール活用

クラウドの構成管理

サービスをインスタンスとして管理…追加や削除が簡単



クラウドの構成管理活用

セキュリティの6つの基本原則…ガートナーセキュリティ&リスク・マネジメントサミットより 【これからの企業のセキュリティ対策に求められる6つの原則】

1. 「チェックリスト点検」から「リスクベース思考」へ転換する
2. 「インフラ」ではなく「ビジネス」を守る
3. セキュリティリーダーは、「ディフェンダー」ではなく「ファシリテーター」になる
4. 情報フローを「制御」するのではなく「理解」する
5. セキュリティ技術の限界を理解する
6. セキュリティ被害は避けられないと認める

セキュリティリスクと対策

セキュリティリスク	主な対策	
✓ データ漏洩 ✓ 不正アクセス ✓ 改ざん ✓ なりすまし ✓ 盗聴 ✓ ウイルス ✓ セキュリティホール ✓ 人為ミス	・ アクセスコントロール ・ リモートアクセス対応 ・ パスワード管理 ・ 多要素認証 ・ VPN利用 ・ IPアドレス制限 ・ ゼロトラスト	・ 端末制限 ・ 暗号化 ・ のぞき見防止フィルタ ・ SIEM ・ セキュリティパッチ ・ 脆弱性診断

アクセスコントロール

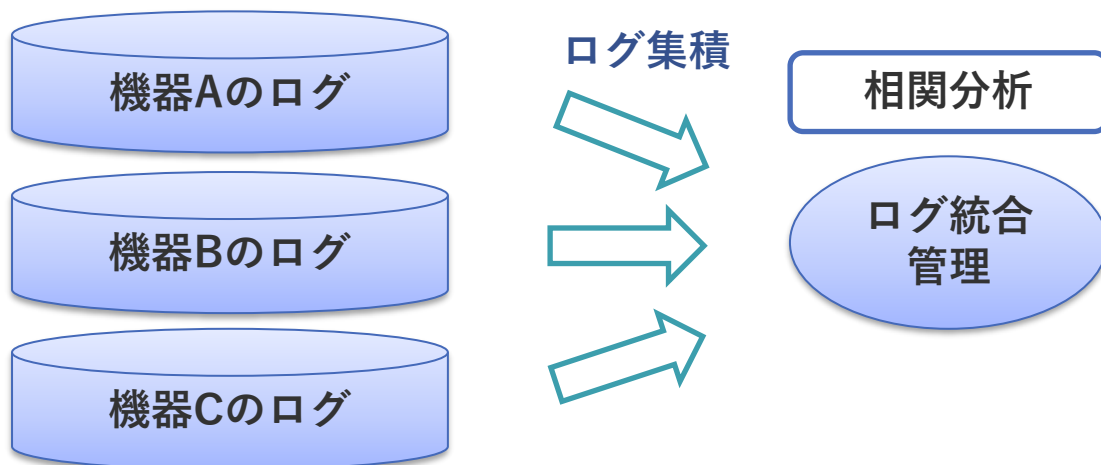
アプリケーションに比べ、運用管理者に対する権限コントロールが大雑把
堅牢なコンピューターーム、厳格な入出退管理ではリスクが減らない

情報漏洩事件の多くは内部犯行

1. サーバーやデータベースの権限を不用意に権限を与えすぎない
2. アプリケーションの権限はさらに慎重に付与する
3. アカウントを共有しない
4. ログで監査証跡をきちんと監視、チェックする
5. それらのポリシーを明記しておく

SIEM(Security Information and Event Management)

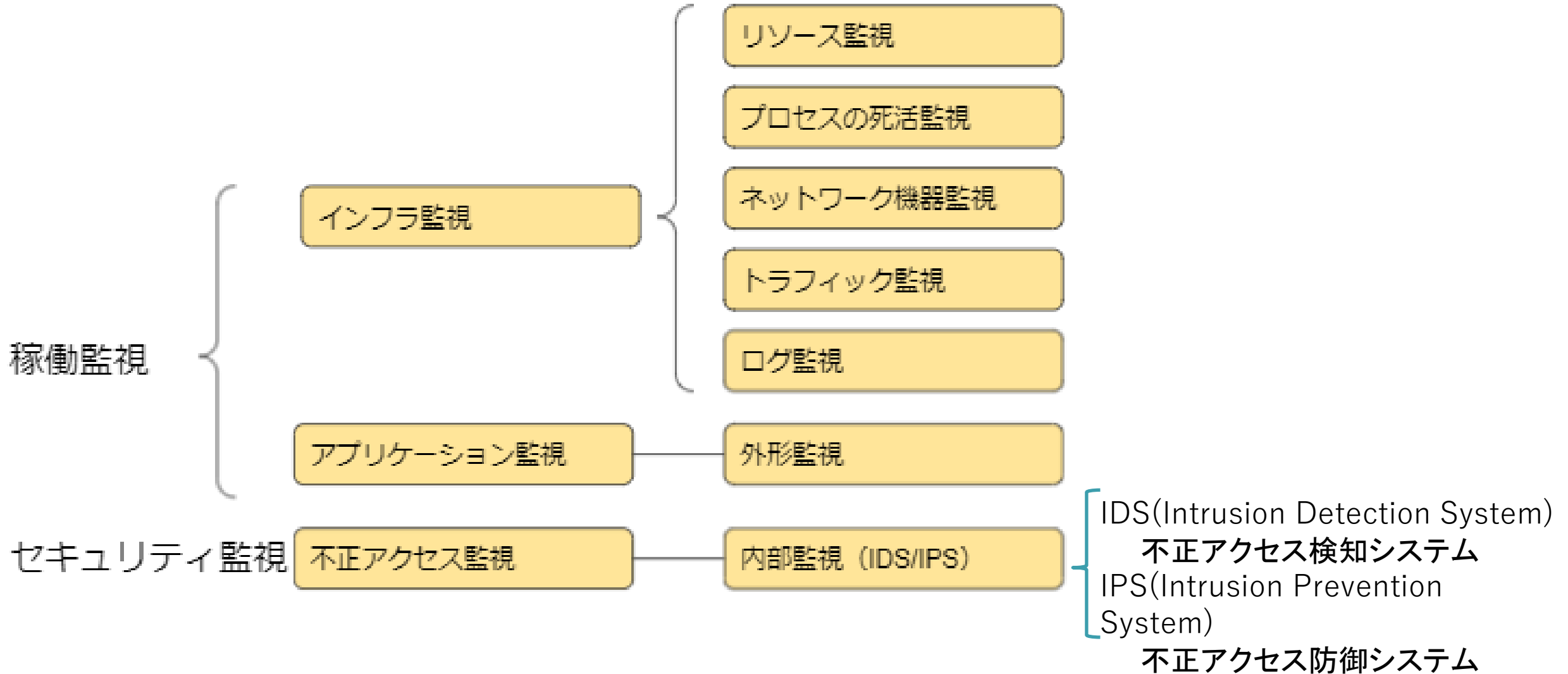
機器やソフトウェアのログを統合管理・分析して脅威を検知するセキュリティソフト



刑事ドラマの「アリバイ」

手作業のログ監視 ⇒ SIEMによる自動監視

稼働監視対象



リソース監視/キャパシティ管理

サーバーやネットワーク機器などのリソース使用状況を監視
(CPU使用率、メモリ使用量、ディスクの空き容量、ネットワークトラフィック…)

アラート自動送信

サーバーやネットワーク機器がダウンしていないかモニタリング
(各機器のプロセスを対象にhttpdプロセスやDBプロセスが正常応答するか確認)

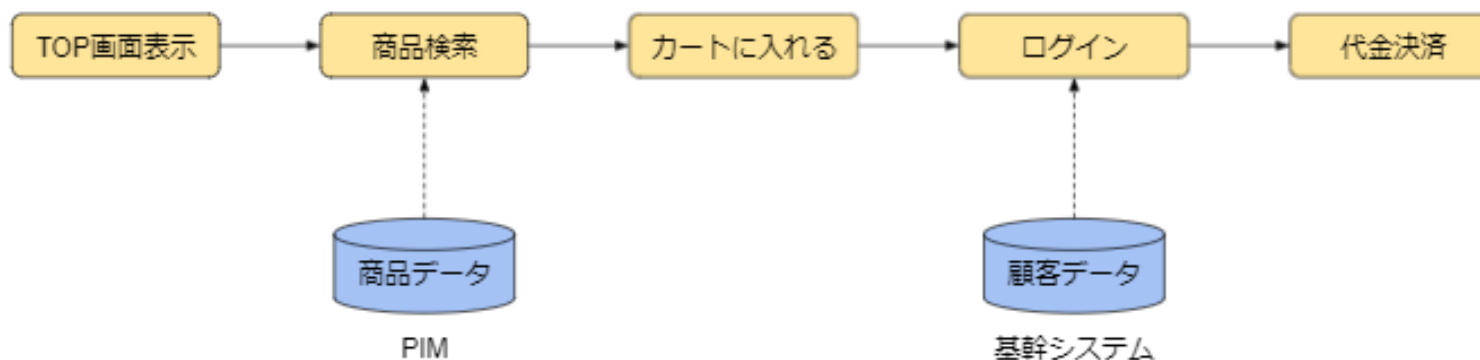
総合監視ツール活用

外形監視

外部からユーザーの視点でアクセスし、応答時間などをモニタリング
URL監視…定期的にURLにアクセスしてページ表示時間を計測
シナリオ監視…シナリオ通りに動作することを監視

アプリケーション監視

(eコマースのシナリオ監視の例)



可用性のKPI…システムがどのくらい使える状態にあるか

$$\text{可用性} = (\text{サービス時間} - \text{停止時間}) / \text{サービス時間} \times 100\%$$

サービス時間 : 運用サービスの提供を取り決めた時間

(例) eコマース 24時間365日対応・・・24H × 365日 × 8760時間

8時間停止の場合・・・可用性 : $(8760-8)/8760 \times 100 = \text{約}99.9\%$

フォーナイン(99.99%)で52.56分、ファイブナインで5.256分の停止時間

保守性のKPI…障害によりシステムが停止してから復旧するまでの時間

$$\text{MTTR} = \text{総停止時間} / \text{システム停止回数} \quad (\text{MTTR: Mean Time to Restore Service})$$

(例) 1年間で3回障害によって停止し、それによる停止時間合計が8時間だった場合

MTTR : $8\text{H} / 3\text{回} = 2.67\text{時間}$

インシデントと障害

Incident : 出来事、事件

3. ツール活用



ツール活用時代（脱「手作業」）

運用管理で使われているツール/サービス

- | | | |
|----------------|---------------------|---------------|
| ✓ IT資産管理ツール | ✓ 運用スケジューラ/ジョブ管理 | ✓ インシデント管理ツール |
| ✓ IaC | ✓ リソース・死活監視ツール | ✓ 障害分析ツール |
| ✓ クラウド構成管理サービス | ✓ ログ監視ツール | ✓ ヘルプデスクツール |
| ✓ SIEM | ✓ 総合監視ツール | ✓ FAQシステム |
| ✓ セキュリティソフト | ✓ 外形監視ツール | ✓ チャットボット |
| ✓ 脆弱性診断ツール | ✓ IDS/IPS(不正アクセス監視) | ... |

システム開発で使われているツール/サービス

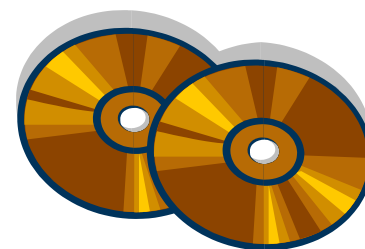
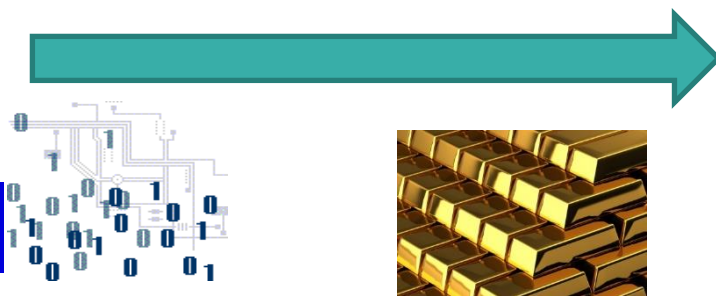
- | | | |
|---------------------|----------------|---------------|
| ✓ 構成・変更管理ツール | ✓ データモデリングツール | ✓ プロジェクト管理ツール |
| ✓ 開発ツール | ✓ チケット管理ツール | ✓ 継続的デリバリーツール |
| ✓ 開発支援ツール | ✓ 静的解析ツール | ✓ DevOpsツール |
| ✓ NoCode/LowCodeツール | ✓ 動的テストツール | ✓ パフォーマンスツール |
| ✓ デバッガ | ✓ テスト自動化ツール | ✓ セキュリティ診断ツール |
| ✓ モックアップツール | ✓ バグトラッキングシステム | ... |

CAD…Computer Aided(Assisted) Design 「コンピュータによる設計支援」
建設業、土木業、家電、機械、自動車、航空機、アパレルなど各業界で活躍

ソフトウェア設計は…WordやExcelなどによる設計を連綿と続けている



コンピュータ設計



プログラムコード生成

自動コードが何十年たっても普及しない理由・・・錬金術？

1. コード生成のため設計作業で詳細な情報を入力しなければならない
2. 通り一遍のアプリしかできず、UIのよいものが作れない



SI Object Browser Designer



追い求めた設計作業の
効率化で特許を取得！

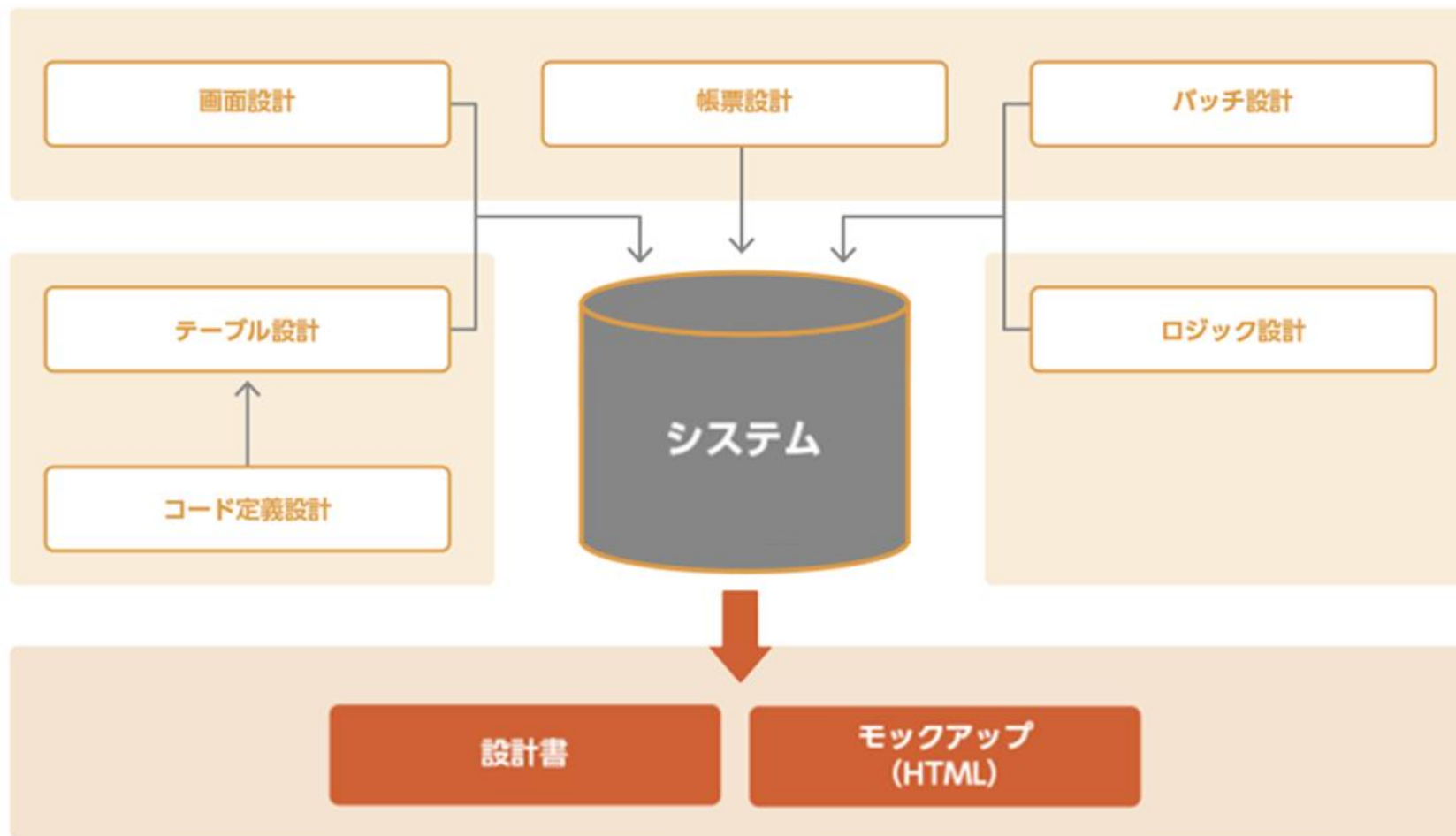
第5702824号

平成27年2月27日取得

設計情報をデータベースで一元管理

SI Object Browser Designer

- ・ 多様なアウトプット・・・機能設計書、モックアップ、CRUD図などを自動生成
- ・ バージョン履歴管理・・・バージョン間の変更箇所比較や仕様変更時の影響範囲を表示





ありがとうございました



時間を奪うのではなく、時間を与えるソフトウェアを創り続ける

System Integrator

※本資料掲載の情報・画像など、すべてのコンテンツの無断複写・転載を禁じます。