P-29

国立研究開発法人 放射線医学総合研究所 情報基盤部 科学情報課
National Institute of Radiological Sciences Dept. of Information Technology Research Information Section

昔誰かが作ったシステムを何とかする

○下村岳夫A)
A) 国立研究開発法人 放射線医学総合研究所 研究基盤センター 情報基盤部 科学情報課

直近20年に渡るPCの普及・インターネット活用範囲の拡大に伴い、多くの組織において"システム"と呼ばれる物は爆発的に多様化・複雑化し続けてきた。今日"システム"に全く関わらず社会生活を送る事は極めて難しい、と思える程である。これだけ様々なシステムに長年に渡って囲まれていると、必然的に『システムが古くなる』事による弊害が発生するようになる。消費税率変更など社会的な環境変化によるケース、OSやミドルウェアのバージョンアップに伴う仕様変更に追従できなくなるケース、業務が変化して実態にそぐわなくなるケース、ハードウェアが老朽化して故障の危機に瀕するケースなど要因は様々であるが、『古くなったシステム』問題は研究現場においても次々に発生しており、我々組織内情報部門には多くの相談が寄せられている。本報告では、近年我々が対処した「昔誰かが作ったシステム」の事例紹介を通じて、これらを「何とかする」にはどうしたら良いのか、将来に向けこのような負担を少しでも軽減するにはどうすべきなのか、新しいシステムの作成・管理サイドの視点で考察する。

【事例1】 公開Webシステム群に広がる脆弱性

2015年5月頃

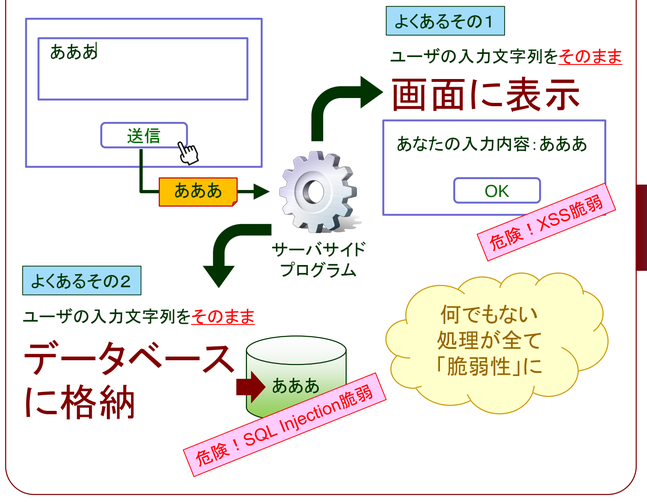
（概要） 所外公開している各種Webシステムには、研究者や事務職員が自作した物も多い。2015年6月、我々情報部門主導で脆弱性の一斉点検を実施した所、13件が要対処項目として検出された。各システムの運用部門に連絡し、急ぎ対処する事となった。

- 仮想環境上に全公開サイトを複製し、Web脆弱性検出ツールを使用
- Web脆弱性検出ツール：ZAP（OWASP社製品）・Skipfish（Google社製品）など
- 主な要対処箇所はPerlやPHPを用いて作成されたサーバサイドプログラム

（問題点）
- 13件中6件は作成者が既に退職しており、対処が難しい状況
- 専門性の高いデータを扱うシステムも多く、「正しい動作」を判断し辛い
- 手作りのソースコードは作成者の癖やポリシーも様々で、解読が困難

（対処）
- 各システムの運用部門と協議し、廃止or改修を検討
- 継続運用が必要と判断された物は、手分けして改修を実施
- 改修内容は基本的な処理が中心だが、網羅性が要求される

手作りプログラムによくあるパターン



対処手段 *1,*2

ソース可読性の向上
- 改行位置・インデント・区切文字・括弧等、記述ルールに一貫性を
- 不必要な省略記述を排除
- 多方面に分散するインクルードファイルを極力まとめる

変数チェックの徹底
- ユーザ入力に限らず、取扱う変数の内容は全てチェック（想定していない文字や記号の無害化or排除orエラー処理）

パラメータ受渡し方法の制限
- セッション管理を厳重に
- フォームリクエストは原則POSTに統一、GETの利用は最小限に

利用関数の見直し
- システム関数の直接利用は原則禁止
- 廃止・非推奨となった古い関数を、推奨される新関数にリプレース
- プリペアドステートメントの利用

全件の調査・改修に要した期間は約3ヶ月間。その後の再検査により、Mediumレベル以上の脆弱性が0件となった事が確認できた。

※Mediumレベル未満の脆弱性として、「Low」や「Information」が多数検出されるが、いずれも一般的な「脆弱性」には当たらない 些細な内容である事が確認できている。
※2016年2月、公開サイトの一部が海外からの執拗な攻撃に晒されたが、深刻な侵入・破壊・改ざんには至らなかった。この一斉点検で基本的な対策を施した事が、密かに功を奏した形と考えられる。

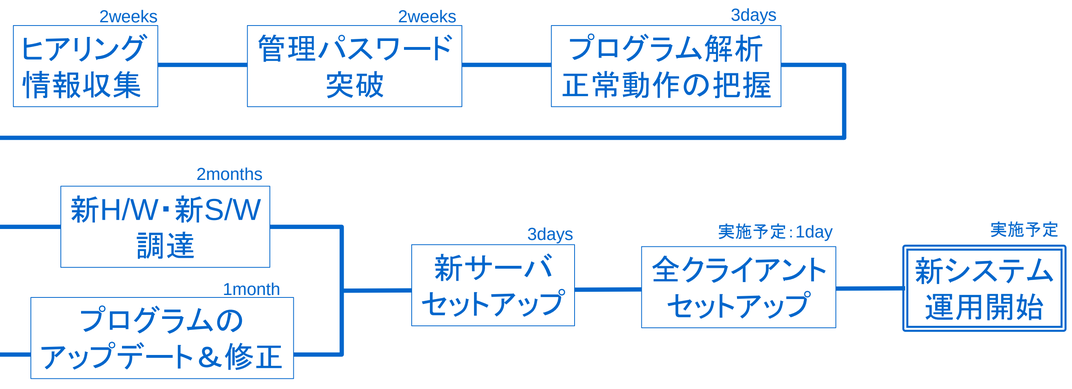
【事例2】 診断データベースの消滅危機

2015年7月頃

（システム概要）
- 用途：患者個人情報や診断履歴・検査予約の一元管理
- 構成：DBサーバ1台（iMac）+クライアント（WinXP,WinVista等）10台
- インターネットから切り離された所内病院専用LANに接続
- 2005年頃運用開始、現在も診断業務上重要な役割を果たしている
- FileMaker 8を用いて当時の職員が作成
- ユーザ：所内病院の医師・看護師・技師10名程度

（問題点）
- 当時の開発者は既に退職、引継ぎは行われず内部構造を知る人が居ない
- 管理パスワード不明・ドキュメントが一切存在しない・バックアップ無し
- ハードウェア老朽化に伴い異音も発生、いつ故障してもおかしくない状況
- ユーザは忙しく、積極的な協力は期待できない

（対処）



- ヒアリング：現場の協力が得辛く困難を極めたが、しつこく現場に通い続ける事でクリア
- 新サーバ：1UラックマウントタイプのHPサーバ機にWindows8.1・FileMaker Pro14をセットアップ
- バックアップ：仕組みはVB Scriptで新規作成し、タスクスケジューラで平日夜間実行（予定）

※全クライアントセットアップ作業中に、現場側都合により一旦中断。新システムの準備が整った所で、本支援作業は完了となった。現在も旧システムで運用継続中だが、データバックアップだけは週1回手動で取ってもらう形とした。現H/Wが故障した時点で新システムに移行する予定。故障が発生した場合、一定の停止期間や最大1週間分のデータ喪失が発生する事は了解いただいている。

【事例3】 図書館業務を圧迫する報告フロー

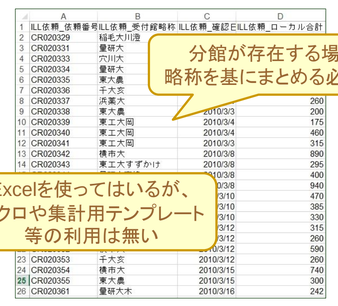
2013年8月頃

（概要） ➢ 所内図書館では月間数十件に上る文献複写の依頼・受付の処理実績を「対応館」・「依頼/受付の別」毎に集計・印刷し、毎月末に経理部門へ報告する必要がある

（問題点）
- 使用している図書館専用ソフトには文献複写実績の集計機能が無い
- 出力したリストを基に集計表を作成・印刷する作業が自動化されておらず、Excelでの煩雑な手作業を強いる報告フローが事実上ルール化されていた

昔誰かが定めた作業の流れ：これも古くなったシステムの一つ

図書館専用ソフトから期間指定でデータを抽出し、Excel出力



Excelで並べ替え・対応館毎に抽出

労力大

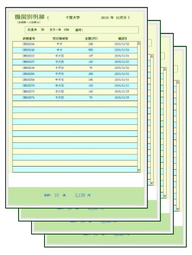
Excelで帳票化・印刷

（対処） ➢ 1clickでデータ読込＆集計、1clickで全帳票印刷できるサブシステムを作成※・提供
- 長年守られて来たルールも変更

図書館専用ソフトから期間指定でデータを抽出し、csv出力

1click

1click



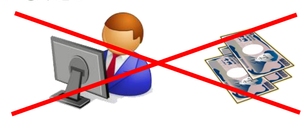
※FileMaker Pro 12 Advancedで作成（開発期間：2weeks）

【事例4】 開発者も予算も無いがWebサイトは重要

2014年7月頃

（概要）
- 某外部協力団体のWebサイト：10年以上前に当時のスタッフにより作成された
- デザイン性に乏しく、静的コンテンツのみで情報発信力に欠ける
- 団体設立20周年を記念して、Webサイトをリニューアルしたい

（問題点）
- 現スタッフにはWeb開発の経験者が居ない
- Webサイト開発にかけられる十分な予算が無い

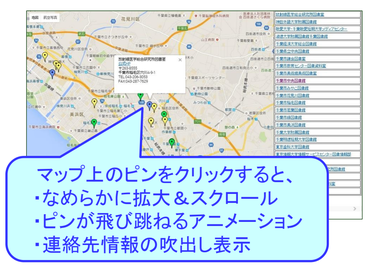


（対処）
- 利用中のレンタルサーバを調査し、バックアップ・CMS環境の導入可否を検討
- WordPressを用いて、現スタッフの手による新サイト構築を提案*3
- 環境整備は代行、Webサイト構築に際しては適宜技術的アドバイスを実施
- Google Map APIを用いた協力組織の動的アクセスマップ作成を代行

WordPress

Web開発未経験のスタッフ1名が、業務の合間に構築作業を実施

2ヶ月でリリース



結果と考察

各事例はいずれも力技で「何とかできた」物である。対処を通じて『古くなった』・『将来古くなる』問題に関して鍵を握る重要ポイントと感じた点は下記の通り。

調査分析（要件定義）	現場をよく観察し、実業務の流れを正確に掴む 真に実現したい事は何かを明確にし、ユーザと合意	新規システム開発時と同様 トライ＆エラーの為にもユーザ側のキーマンを味方に付けるのが近道
方針検討（基本・詳細設計）	既に不要となった処理を見極め、「思い切って捨てる」	
実装（プログラミング）	移植性の高さはそのまま老朽化に対する強さとなる	➢ ベタでも良いから丁寧に可読性の良いソースコード記述を ➢ 汎用性が高く長寿命と思われる言語・環境の選択を ➢ 網羅性に十分注意を
運用支援（ユーザサポート）	慣習や先入観念を排除してルール検討を	➢ 管理情報を文書で残す事や、バックアップの重要性をユーザに正しく理解してもらう ➢ もっと改善できる点は無いか？

参考文献

*1 速攻！図解プログラミング PHP+MySQL Hershe 2006（株）毎日コミュニケーションズ P12
*2 PHPサイバーテロの技法-攻撃と防御の実際- GIJOE 2006（株）デジカル P49-112
*3 基礎からのWordPress 高橋のり 2014 SBクリエイティブ（株） P1-48