

# A Case Study on Software Requirements Engineering using the FBI Virtual Case File Project Report

N. Devadiga

ndeivadig@alumni.cmu.edu

**Abstract**—Virtual Case File (VCF) was a case management software to be developed by the United States Federal Bureau of Investigation (FBI) to replace the existing Automated Case Support (ACS) software system. The goal of the project was to modernize FBI's suite of investigative software applications; the ACS system was developed in-house consisting of several layers of applications that were outdated and difficult to use. Based on the Goldstein's [1] report it was identified that the VCF system did not adhere to the requirements of the project and was fragmented. This case study identifies the critical problems from requirements engineering perspective that contributed to VCF project failure and discusses software engineering methods that would assist in requirements gathering.

**Keywords**— Requirements engineering, software project management, requirement analysis

## I. INTRODUCTION

The primary objective of the Virtual Case File (VCF) was to automate paper-based work environment and allow agents, analysts share vital investigative information and replace the obsolete Automated Case System (ACS). The Virtual Case File project spanned from 2000 to 2005 during which it experienced a series of software engineering failures. The goal of the project was to modernize FBI's suite of investigative software applications; the ACS system was developed in-house consisting of several layers of applications that were outdated and difficult to use. Based on the Post [1] and Goldstein's [2] report it was identified that the VCF system did not adhere to the requirements of the project and was fragmented. This case study identifies the critical problems from requirements engineering perspective that contributed to VCF project failure and discusses software engineering methods that would assist in requirements gathering.

## II. REQUIREMENTS PROCESS ISSUES

The initial requirement was to upgrade the bureau's existing Automated Case Support system. The ACS

system built by the bureau enabled the agents to search and analyze material between different cases, the system was deemed legacy as it was constructed using old tools like Natural [6], ADABAS [7], IBM terminals [8] from the 70's. Due to the limitations and legacy dependency of the ACS, the requirements were changed to create an entirely new application with a new database and graphical user interface. As per Goldstein [2] report, product requirements were discussed with more than forty domain experts rather than involving few crucially required domain experts, architects, developers, business analysts and the management team. There was no clear distinction between the project's stakeholder's, business analysts and developers. Ideas proposed by independent members in the meetings were added to the requirement list, and requirements were frequently modified without the focus on defining the mandatory core functionalities. Short term goals, schedules, strategies, milestones, model to be adopted were not defined in the meetings. The project team focused on achieving the end goal, rather than identifying project milestones and clarifying/refining the requirements to meet the milestone. The herculean task of building the entire project first time around without clear milestones lead to vague requirements and ever-changing requirements. The VCF project adopted the Ad-hoc (Hobbyist) model [11] with new additions and modifications to the requirements, and there was no defined structure. It is stated that the lack of robust technical architecture is one of the leading reasons for the failure of the project [9]. The design document consisted of more than 800 pages specifying every detail of the project rather than portraying just the high-level design for better comprehension.

When a certain portion of the requirement was developed, stakeholders identified new issues or thought of new ideas and a new modified requirement was proposed. There was no final structure on what is to be delivered and what process model, and framework is to be adopted.

The entire project was to be deployed at once, and the old ACS system was to be discontinued immediately. No

business risk management process was carried out; no backup business continuity plan was decided.

A well-documented requirement specification document is essential for building a detailed technical architecture. Below are two reasons that contributed to incomplete requirements and technical architecture:

1. Lack of planning and requirement analysis - As per SWEBOK [10], to design and build a product, it is essential to understand the requirements of the product, its functionalities and the users of the system. It is of primary importance to understand the requirements from the clients who will be using the system. With appropriate requirement elicitation techniques, a concise and clear requirement specification aids in the development of technical architecture and precise estimations related to time and resources. In the Virtual Case File project, requirements were gathered through group discussions from a wide array of members rather than elicitation from the customers alone. The lack of formal software engineering training among its members impacted the requirements gathering process. As the project adopted, ad-hoc software process requirements were routinely modified leading to additional un-expected downstream changes in other phases of the software development lifecycle. Changes in requirements led to constant changes in the product architecture and development, and with the lack of milestones and expectation to cutover to the new system without transition added to the projects agony.
2. Lack of Responsibility and Accountability - Lack of governance played a significant role in the failure of the VCF project as the management team lacked training in software project management, information technology, and computer science. In the project, the program managers did not duly evaluate the scope of project, schedule, effort, project plan, assign appropriate roles keeping in line with the goals of the project.

a. A responsibility assignment matrix [4] (RACI matrix) is useful in identifying roles and responsibilities for a project. In short, the RACI matrix provides the following insight:

- R – Responsibility: Who is responsible?
- A – Accountability: Who is approver?
- C – Consulted: Whose opinions are sought?
- I – Informed: Who is to be updated about the activities?

Description	Responsible	Accountable	Consulted	Informed	Virtual Case File Project
Requirement Analysis	Analysts, Domain Experts	Customers and stakeholders	Management	Customers and Management	RACI - Group discussion, No clear separation of roles C – Attrition, Micromanagement I – Not adequately involved
Design	Analysts, Domain Experts, Developers	Customers	Customers	Customers and Management	R - Lack of precise design specification, Lacked formal training, Poor design choices ACI – Requirement changes
Implementation	Developers	Manager	Domain Experts, analysts	Customers and Management	AC- Ad-hoc Model R - Poor architecture blueprint
Testing	Developers, Testers	Customers and Management	Customers and Management	Customers and Management	RACI - Blame game among teams

It helps in improving governance and identifying problems at the various stages of software development lifecycle. Below is a RACI matrix built to identify issues in steps of the VCF project.

### III. CONCLUSION

Below are two important attributes that can be adopted by similar projects -

1. Understanding requirements require thorough comprehension of the product to be designed; this can be achieved by iteratively discussing the requirement with the clients and defining project

milestones. Clients should be involved in all stages of the software development lifecycle; this helps in gathering the requirements and obtaining immediate feedback at each step [3]. This would ensure that the product specification satisfies the client requirements. Various requirement elicitation techniques as defined in SWEBOK can be adopted.

In case of the VCF project, there was no clear distinction between the stakeholders and rest of the members, and requirements were discussed and added based on personal judgments and group discussions.

Requirement analysis is an iterative process and involving clients in each iteration would update the client with the current state of the project as well help in capturing important feedback.

It is imperative to have domain expertise based on the requirement to extract maximum information from the client, unlike in the VCF project the members lacked formal training software engineering and computer science as managers and engineers [9].

## 2. Adopt a suitable framework

To streamline the software development process SWEBOK [10] suggests the use of software processes as per the requirements of the project. A software process like Microsoft Solution Framework [5] would have been a good fit for the VCF project. Microsoft Solution Framework incorporates Agile practices and functions such as open communication, shared vision, empowering team members, shared responsibility, clear accountability, focus on business value, investment in quality and learning from experience. Phases of MSF such as the envisioning phase explores and identifies the scope of the project, and the planning phase discusses and approves project plans. MSF risk management phase recognizes the risks involved in specific steps of the software development and helps by providing the lessons which were captured in other projects for similar situations, and this would have been helpful as the VCF project did have any back-up for transitioning from the ACS system VCF.

and Science News, IEEE Spectrum, 1 Sept. 2005, [spectrum.ieee.org/computing/software/who-killed-the-virtual-case-file](http://spectrum.ieee.org/computing/software/who-killed-the-virtual-case-file).

- [3] "A Field Study of the Software Design Process for Large Systems." Communications of the ACM, ACM, [dl.acm.org/citation.cfm?id=50089](http://dl.acm.org/citation.cfm?id=50089).
- [4] "9.1.2.1 Organization Charts and Position Descriptions". A Guide to the Project Management Body of Knowledge (PMBOK Guide) (5th ed.). Project Management Institute. 2013. p. 262. ISBN 978-1-935589-67-9.
- [5] Keeton, Marlys (April 2006). Microsoft Solutions Framework (MSF): A Pocket Guide. Van Haren Publishing. p. 15. ISBN 90-77212-16-7.
- [6] "Adabase new products". Computerworld. May 29, 1978. p. 27
- [7] S.M. Deen (1977). Fundamentals of Data Base Systems. p. 174. ISBN 1349158437.
- [8] IBM Corporation (1972). IBM 3270 Information Display System Component Description
- [9] "FBI's Virtual Case File System." FBI, FBI, 3 Feb. 2005, [archives.fbi.gov/archives/news/testimony/fbis-virtual-case-file-system](http://archives.fbi.gov/archives/news/testimony/fbis-virtual-case-file-system).
- [10] "Version 3.0." Guide to the Software Engineering Body of Knowledge (SWEBOK(R)), IEEE Computer Society Press, [dl.acm.org/citation.cfm?id=2616205](http://dl.acm.org/citation.cfm?id=2616205).
- [11] Ad-hoc development model - <https://courses.cs.washington.edu/courses/cse403/07sp/lectures/Lifecycles.pdf>

## REFERENCES

- [1] "The FBI's Upgrade That Wasn't." The Washington Post, WP Company, 18 Aug. 2006, [www.washingtonpost.com/wp-dyn/content/article/2006/08/17/AR2006081701485.html](http://www.washingtonpost.com/wp-dyn/content/article/2006/08/17/AR2006081701485.html).
- [2] Goldstein, Harry. "Who Killed the Virtual Case File?" IEEE Spectrum: Technology, Engineering,