An IT project postmortem:

identifying root causes and eliminating rival explanations

Jens Schmidt

IT University of Copenhagen, Rued Langgaards Vej 7, 2300 Copenhagen S, Denmark

ARTICLE HISTORY

Compiled November 27, 2024

ABSTRACT

Information technology (IT) projects often fail. Postmortem analysis is not general practice in IT project management. This is a missed opportunity for IT project management because postmortem analysis is a proven source of practice improvements and preventive actions in other domains. In this paper, the root causes of failure of a major IT project are identified by postmortem analysis, a well-established method for investigating accidents and failure ex post facto to improve practice and performance. The root causes of identified are: a) inadequate planning, b) novelty of a technology to the organisation, and c) inappropriate software development method and process. The postmortem offers insights into risks and challenges that IT projects still face today. Significantly, the postmortem analysis shows how a different approach to project planning could have prevented the failure and termination of the project. This paper also demonstrates how systematic IT project postmortem analysis can be conducted based on leading theory of process tracing and causal modelling in combination with the literature on IT project failure. The demonstration of this approach to IT project postmortems is new and original.

KEYWORDS

Information technology; project management; project failure: project postmortem; process tracing

1. Introduction

Many IT projects fail with respect to budget, schedule, and benefits. IT projects overspend by 73% of budget on average (Flyvbjerg et al., 2022). Almost one in five IT projects spend roughly five times the original budget on average (Flyvbjerg et al., 2022; Flyvbjerg and Gardner, 2023). IT projects have lower budget-performance than other types of projects (Flyvbjerg and Gardner, 2023). Jones (1995) reports that 48% of large software projects get cancelled.The Standish Group (2022) find that 19% of IT projects fail, that 43% are "challenged", and that 43% of IT projects are successful. This level of performance reflects the state of current IT project management practices. There is ample support for the conclusion that improvement of IT project performance is desirable.

Postmortem analysis is a well-established method for investigating accidents and failures *ex post facto* to improve practices and performance in, for example, medicine (Gawande, 2011), aviation (NTSB, 2016), construction, and infrastructure (Petroski, 1992). According to Petroski:

> [...] successes in engineering have tended to arise not out of a steady and incremental accumulation of successful experience, but rather in reaction to the failure of the past Petroski (1992).

1.1. Related research

The point of project postmortems is learning not to repeat past mistakes in future projects. The view that IT project postmortem analysis of failed IT projects is necessary is widely supported (Ahonen and Savolainen, 2010; Birk et al., 2002; Ewusi-Mensah, 2003; Glass, 2001; Nelson, 2005; Reel, 1999; Verner et al., 2005; Williams, 2004). Systematic project retrospectives are recommended by the Project Mangement Institute (PMI, 2021a,b). Postmortem analysis is currently, however, all but absent as a systematic tool in IT project management (Ahonen and Savolainen, 2010; Kasi et al., 2008; Verner et al., 2005; Williams, 2004). The reasons that formal IT project postmortems are rare include that (a) project postmortem is time consuming, (Glass, 2002), (b) postmortem analysis is difficult to do well (Ahonen and Savolainen, 2010;

Verner et al., 2005), (c) project participants may be unavailable or unwilling to contribute (Ahonen and Savolainen, 2010), and that (c) organisations do not command the necessary mechanisms to exploit the postmortem results (Kasi et al., 2008). Additionally, (d) some organisations are reluctant to invest in the analysis of their project failures, especially when the benefits cannot be quantified beforehand (Nelson, 2005).

Current IT project postmortem methods involve combinations of (1) data collection from interviews, surveys, and documents, (2) workshops, (3) data analysis, and (4) reporting (Ahonen and Savolainen, 2010; Boddie, 1987; Collier et al., 1996; Dingsøyr, 2005; Kerth, 2001; Myllyaho et al., 2004; Nelson, 2021; Paté-Cornell, 1993; Tiedeman, 1990).

1.2. A new additional approach to postmortem analysis

In this paper, we present an IT project postmortem analysis that addresses some of the challenges for postmortems by combining well-documented methods in a new and original approach. In the approach followed, (1) the postmortem analysis is based almost exclusively on documents, which makes the postmortem less time consuming and less costly for the project owning organisation. The mainly document-based approach reduces the problem of unavailability or unwillingness of project participants. The reliance of project documents as the main data source also reduces negative effects of hindsight bias (Flyvbjerg, 2021), failing memory, and after-the-fact rationalisations in interviews. (2) The postmortem analysis approach uses current theory of process tracing and causal modelling for data analysis, which are well-documented methodologies for strong causal inference and generalisation. (3) The postmortem approach systematically leverages well-documented causes of failure from the literature on IT project failure for hypothesis generation, and for the elimination of rival explanations. Postmortem analysis is perhaps inherently difficult, but process tracing in combination with the systematic use of the literature on IT project failure facilitates a systematic and methodologically sound approach to causal inference in IT project postmortems. (4) Process tracing enables and facilitates the identification of both contextual and general causal mechanisms (Beach, 2019; Waldner, 2019).

Research has identified and reconfirmed factors that can lead to IT project failure (Ayat et al., 2021; Dwivedi et al., 2015a; Hughes et al., 2016b, 2017; Schmidt, 2023b). According to Sauer (1999) the failure factors in the literature are highly abstract, and therefore they have had limited impact. More insight is needed into the causal mechanisms that link failure factors to failure. Process tracing provides insights into the causal mechanisms that link causes to outcomes (Beach, 2019) in a way that can be both contextual and general (Waldner, 2019). Process tracing based project postmortem analysis can therefore provide knowledge about the causal mechanisms of individual project failures, but also - on certain conditions - identify general causal mechanisms of IT project failure.

1.3. The IT project postmortem

In this paper, we present a postmortem analysis of POLSAG[1] (1995-2012, $86m), a failed IT project undertaken to develop a new digital casefile management system for the Danish Police. The postmortem analysis is conducted using process tracing, a well-established method for causal inference in case studies (Beach and Pedersen, 2013; Bennett, 2008; Bennett and Checkel, 2014; Collier, 2011; George and Bennett, 2005; Mahoney, 2012; Scriven, 1974; Trampusch and Palier, 2016; Van Evera, 1997; Waldner, 2012, 2015) and causal modelling (Pearl, 2009, 2013; Pearl and Mackenzie, 2018) in combination the literature on causes of IT project failure (Schmidt, 2023a). The combination of well-established theory and contextual findings in IT project postmortems are solid foundations for recommending practice changes and for contributing to project management theory.

The POLSAG project is ideal as a case study because a) POLSAG was a sizeable and complex IT project, b) the POLSAG project illustrates challenges that IT projects face still today, and c) all project documents have uniquely been made available for research. As occasional external project reviewers, we find that the learnings from the POLSAG project failure are still not universal knowledge among IT professionals and managers.

[1]POLSAG is the name of the project found in official documents. “POL” for police, and “SAG”, which means “case” or “casefile” in Danish.

2. Methodology, materials, definitions, and approach

In this section, we account for the materials used for the case study, the definitions used, the method and approach to the case study, and the research questions.

Individual project postmortems are ipse facto single case studies. In this paper, a case is "an instance of a class of events" (George and Bennett, 2005), and the event studied is an instance of IT project failure. The case study method followed is process tracing, a well-established and well-documented method for case studies (George and Bennett, 2005; Yin, 2018). Case studies is a valid approach for both explorative, interpretive (Klein and Myers, 1999), and explanatory qualitative research (Sarker, 2021; Yin, 2009) and theory building (Beach, 2019; Eisenhardt, 1989; George and Bennett, 2005). The project postmortem in this paper is an explanatory analysis. The possibility of generalising the findings in case studies is widely accepted (Flyvbjerg, 2006; George and Bennett, 2005; Sarker, 2021; Yin, 2009). Yin (2009, p15-39) calls this "analytical generalisation", and Waldner (2012; 2015; 2019) provides specific criteria for generalisation in process tracing-based studies. See also Section 2.3, Method, and Section 2.5, Specific approach to data analysis by process tracing.

Project postmortem analysis is undertaken to identify the causes of project failure and explain *how* the project failed so that preventive actions against similar failure can be designed. Process tracing is used for analysing causal chains of events between causes and effects in way that (1) allows strong causal inference (Waldner, 2019), and (2) unveils the underlying causal mechanisms that link causes to effects. This is exactly what is needed for postmortem analysis. Causal mechanisms may be event-specific or generally valid ("invariant causal mechanisms"). This makes process tracing ideal for IT project postmortem analysis. Process tracing has a long history in case study theory (Beach, 2019; George and Bennett, 2005; Waldner, 2019; Yin, 2018), it is well-developed and well-documented. Causal mechanisms are theoretical devices that have gained much attention lately in several research domains. The causal mechanisms that link well-known failure factors (Hughes et al., 2016b; Schmidt, 2023b) to failure in actual cases are ideal for devising alternative preventive actions against similar failure. Causal mechanism are less abstract than failure factors (Sauer, 1999), which

make them more useful for devising corrective actions. The generality of findings hinge on the case specific, or invariant, nature of the causal mechanism identified.

2.1. Materials

The data studied includes more than 2,400 project documents, official reports, and interviews with key project participants to fill out gaps in the archival data and to verify assumptions based on the archival data (Brady and Collier, 2010). Project documents in corporations and government agencies meet high standards of authenticity, credibility and clarity as data sources (Bryman, 2016; Scott, 2014, p5-35). Project documents may lack nuances that after-the-fact interviews can provide. But for project postmortems, project data and documents have the advantage of being recorded in real time during the progress of the project. They are therefore not subject to after-the-fact rationalisation or hindsight bias.

The main data analysed are project documents and other official and internal documents pertaining to the POLSAG project. Only three interviews were necessary to verify the account of the project that could be extracted from the documents. One interview with the former Head of IT at the Danish Police to confirm the correctness of our understanding of the project process. A second interview with the Account Director with the contractor to verify our understanding of the project documents, which included the full contract and contract revisions. The project documents analysed are from the project owner's (the Police) archives, so the contractor could potentially have additional and contradicting data. They did not. A third interview was conducted with the CEO and CTO of the subcontractor, who also might have relevant additional data, which they did (cited in the postmortem analysis).

2.2. Definitions

We follow Morris (2011; 2013) in considering *the project* - a combination of a business undertaking and a temporary organisation - as the unit of analysis for the project postmortem. We use the general definition of IT project failure and the theory of IT project termination in Schmidt (2022) and Morris and Hough (1987). Project ter-

mination is a special case of failure that the postmortem analysis must explain: the explanandum.

In this paper, we use Paradies and Busch's (2001; 1988) definition of "root cause" as "the *most basic* cause that can *reasonably* be identified and that management has control to *fix*". We also follow their addition that "any particular event may have several 'root causes' that need correcting to prevent recurrence of the event".

2.3. Method

Important research spanning more than four decades on the study of IT project management and failure is available in the literature (Ackermann and Alexander, 2016; Baker et al., 1983; Brown, 2001; Cerpa and Verner, 2009; Charette, 2005; Chua, 2009; Cole, 1995; Eden and Ackermann, 1992; El-Emam and Koru, 2008; Ewusi-Mensah, 2003; Fowler and Horan, 2007; Glass, 1998; Hughes et al., 2016a,b; Jones, 1995; Kappelman et al., 2006; Keider, 1984; Keil et al., 1998; Kerzner, 2014; McManus and Wood-Harper, 2008; Meier, 2008; Morris and Hough, 1987; Pinto and Mantel, 1990; Pinto and Slevin, 1987; Schmidt, 2022, 2023b; Schmidt et al., 2001; Standish, 2014; Verner et al., 2008; Williams et al., 2001; Yardley, 2002; Yeo, 2002). This literature is the basis for developing root cause hypotheses, and for the comprehensiveness of rival explanations considered. The literature on IT project failure has identified and reconfirmed a set of failure factors (FF), which are "areas of activity" (Pinto and Slevin, 1987) where causes (C) of project failure originate, see Figure 1 and Section 3.4.

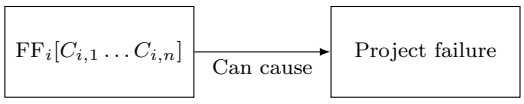

Figure 1. Causal graph for failure factors (FF) and causes (C) in the literature (Schmidt, 2023b).

For example, the known failure causes: "unrealistic objectives", "unclear objectives", and "changing objectives" belong to the "area of activity", or failure factor (Pinto and Slevin, 1987), "Project Objectives" (Schmidt, 2023b). The analytical focus of the postmortem analysis by process tracing is "on understanding the processes whereby causes contribute to outcomes, opening up what is going on in the causal

arrow in-between" (Beach, 2019). The event-history maps in Figures 2, 3, and 4 are "opening up what is going on" (Beach, 2019) between root cause and failure in the the postmortem analysis of the case.

Process tracing is qualitative and non-algorithmic method of analysis that relies in part on the process tracer's background knowledge (Collier, 2011). IT project failure factors embody both theoretical and practical background knowledge (Sauer, 1999). The systematic use of failure factors in the approach presented thus provides background knowledge to support the postmortem analysis.

Process tracing theory is well-established theory of causal inference in social science (Byrne and Uprichard, 2012; Kincaid, 2012), political science (Brady, 2011; Goodin, 2011), and political methodology (Bennett, 2008). Process tracing can be used in conjunction with Pearl's seminal work on causal modelling using directional acyclic graphs (Pearl, 2009, 2013; Pearl and Mackenzie, 2018; Waldner, 2012, 2015).

The theoretical and methodological foundations for the POLSAG project postmortem analysis in this paper are process tracing, causal mechanisms, and causal modelling (Bennett, 2008; Collier, 2011; Pearl, 2009, 2013; Pearl and Mackenzie, 2018; Waldner, 2012, 2015, 2019, 2022) combined with the literature on IT project failure factors (Dwivedi et al., 2015b; Schmidt, 2023b) that represent the domain-specific background knowledge needed to develop and assess postmortem hypotheses (Schmidt, 2023a). The literature on failure factors, causes and mechanisms analysed and synthesised in (Schmidt, 2023b) is used to support the systematic identification of root causes. It is also used as a quasi-finite set of rival explanations.

Process tracing is an analytic tool for the systematic examination of diagnostic evidence that can be used to describe social phenomena, evaluate causal claims about sequences of events, and gain insight into causal mechanisms (Bennett, 2010; Collier, 2011; George and Bennett, 2005). Mechanisms can be event-specific or generally valid invariant causal principles (Waldner, 2012, 2015).

We follow Waldner's approach to process tracing which involves articulating causal graphs, event-history maps based on archival data, and checking the correspondence between them by "descriptive inference" (Waldner, 2015, p247-248). Waldner's process tracing method (2023; 2012; 2015; 2016; 2019; 2022) uses DAGs, directed acyclic graphs

(Pearl, 2009; Pearl and Mackenzie, 2018), for symbolising causal graphs and event-history maps. Since we only claim general validity of the root causes identified by the POLSAG postmortem within the scope of similar projects, we have not elaborated separate causal graphs, but let the causal graphs be implied in the event-history maps. We use the term "causal graph" for the combined event-history maps and causal graphs.

Eliminating rival explanations of outcomes is a fundamental challenge for causal claims based on observational (i.e., non-experimental) data, whether they are single case studies or dataset analyses (Collier et al., 2010). In this paper, we have used evaluations of individual causal inferences by analysing inferences as combinations of necessary and sufficient conditions (Collier et al., 2010). For example, a "smoking gun" is a sufficient, but not necessary condition. A "straw in the wind" is a neither sufficient nor necessary condition, but still a potentially relevant indication of a causal relationship. The background knowledge needed for eliminating rival hypotheses is represented by the literature on factors that lead to IT project failure (Schmidt, 2023b).

The transparent and explicit presentation of inferences and evidence step by step is a result in its own right (Dekker, 2014). A concise specification of the causal chain from root cause to failure enables the well-substantiated development of preventive actions. The articulated specification of the causal chain from root cause to failure also supports focused communication between stakeholders that can facilitate consensus about postmortem conclusions. Stakeholder acceptance of postmortem conclusions prepares the ground for the changes needed to improve project management practices for the benefit of future project performance.

2.4. Research questions

The main postmortem question is:

(RQ1): Why did the POLSAG project fail?

Project postmortems are ipso facto single case studies. Well-established foundations for drawing general conclusions from single case studies are widely acknowledged (Brady and Collier, 2010; Collier, 2011; Collier et al., 2010; Flyvbjerg, 2006; Maxwell,

2002, 2004; Waldner, 2012, 2015, 2019; Yin, 2009). The second research question addresses the general implications of the POLSAG postmortem:

(RQ2): How can the POLSAG postmortem inform changes to practice?

2.5. Specific approach to data analysis by process tracing

The specific approach of the POLSAG project postmortem analysis is identification of root causes and rejection of rival hypotheses by the following steps:

(1) Determination of the explanandum from the project data. In the POLSAG postmortem, the explanandum is project failure and termination for six specific reasons given by the Danish government, see Section 3.2.

(2) Chronological account of main project events based on the project data. . Waldner and Pearl's notion of causation is acyclic, i.e., causes precede their effects.

(3) Consideration of each IT project failure factor from the literature as a hypothetical root cause of failure (see Section 3.4):

   (a) For each failure factor (potential root cause): Test if a causal chain of events between failure factor and failure can be identified by analysis of project documents and interviews. For example, if the data support an observation of "unclear objectives" (the failure factor "Project Objectives") then the next question is: Can an unbroken causal chain of events that links "unclear objectives" to failure (the explanandum) be identified? If yes, the hypothesis is promising. A promising hypothesis may be overridden by other hypotheses that are more fundamental root causes, or hypotheses that are more comprehensively supported by data.

   (b) For promising hypoteses, use process tracing (Waldner, 2012, 2015, 2019) to draft causal graph (generally valid causal inference) and event-history map (case specific causal inference). If the event-history map and the causal graph are isomorphic, we consider the causal graph to be implied by the event-history map.

(c) Verify correspondence between event-history map and causal graph by descriptive inference (Waldner, 2012, 2015, 2019) supported by project data from documents, interviews, and background knowledge.

(d) Accept the hypotheses that can be linked to the explanandum by process tracing as contributing root causes or rival explanations.

(e) Reject hypotheses that cannot be linked to the explanandum by process tracing.

(4) Development of preventive actions where there is strong descriptive inference between case specific causal inference and general causal inference, i.e., where invariant causal mechanisms are identified (Waldner, 2012, 2015, 2019).

3. The POLSAG project postmortem

In this section, we present an overview of the POLSAG project, account for what the analysis must explain, the explanandum, and present the postmortem analysis itself.

3.1. The POLSAG project overview

The purpose of the POLSAG project of the Danish Police was to support centralisation and staff savings by providing an updated IT system for case file management and police reports (The Danish Ministry of Finance and PA Consulting Group, 1995). The project feasibility studies conducted in 2002 consisted of multiple reports by external consultants from international management consultant companies. The consultants recommended that the existing system should be replaced by a new IT system implemented in a "big bang"-approach. Subsequent reports by other international consultants assessed that the existing system was "technologically outdated". Another consultant report concluded that a COTS-approach would be feasible. There was no consolidated feasibility assessment of technical, economic, and organisational feasibility, and risk associated with the recommended approach. The project was initiated in 2004.

The project was terminated as a failure in 2012 after multiple budget extensions, delays, and unsatisfactory pilot operation. An external evaluation of the project after

the unsuccessful pilot operation recommended termination of the project because of (1) technical risks, (2) supplier risks, (3) unsatisfactory pilot operation, (4) implementation risk, (5) obsolescence risk, and (6) unsatisfactory business case. See also Section 3.2. The project had spent approximately $86m (National Audit Office, 2013a), see Table 1. An out-of-court settlement for an undisclosed sum was subsequently reached between the Danish Police and the contractors. The settlement is an indication that both the Police and the contractors acknowledged contributing to the failure of the project.

Table 1. POLSAG project, chronological overview.

Year	**Event**
1995	Turnus study (pre-study)
1999	Budget analysis
2002	IT Strategy
2005	Funding granted
2007	Contract signed
2008	System specification approved
2009	Plan revised
2010	Acceptance test approved
2010	Pilot operation starts
2012	Project terminated

The National Auditor made a formal audit of the reasons for terminating the project and of the soundness of the decision to terminate. The formal audit did not systematically investigate the root causes of project failure, and the audit did not put forward recommendations for changes to practice.

3.2. Failure and termination, postmortem explanandum

The POLSAG project was terminated by the Danish government (National Audit Office, 2013b) for the following reasons:

(1) Technical risks: The insufficient quality of the source code, and the lack of documentation of adequate performance (response times) and scalability of the software.

(2) Supplier risks: The cooperation between the contractor and the sub-contractor was unsatisfactory.

(3) Experience from pilot operations at Bornholm: The pilot operation was unsatisfactory, and deliveries were inadequate.

(4) Implementation risks: Moving to digital workflows would require a large change management and training effort.
(5) Obsolescence risk: The POLSAG system was not suitable for future efficiency improvements of police work processes.
(6) Unsatisfactory business case: It would be time-consuming, resource demanding and costly to solve the problems with the system. It would be "throwing good money after bad" (Schmidt, 2022), and the system would provide limited possibilities for introducing more efficient workflows. Given the cost, the net value of the benefits would be very limited.

In the POLSAG project, the six reasons for terminating the project characterise the failure that the postmortem analysis must causally link to root causes by process tracing: the explanandum.

3.3. The project postmortem analysis

The POLSAG postmortem is based is based on more than 2,400 internal project documents, official reports, and interviews with key project participants to fill out gaps and verify clues in the archival data (Waldner, 2012, 2015). The causal graphs and event-history maps were developed simultaneously because the research literature on project failure factors often does not specify the insides of the "causal black boxes" that link general failure factors and causes to failure: the causal mechanisms.

3.4. Hypotheses, root causes and rival explanations

In this section, the causes of failure of the POLSAG project are analysed and evaluated using hypotheses based on known causes of IT project failure from the literature (Schmidt, 2023b).

Research spanning more than fourty years has identified and reconfirmed a limited number of IT project failure factors and causes (Baker et al., 1983; Brown, 2001; Cerpa and Verner, 2009; Charette, 2005; Chua, 2009; Cole, 1995; Dwivedi, 2013; El-Emam and Koru, 2008; Ewusi-Mensah, 2003; Fowler and Horan, 2007; Glass, 1998; Jones, 1995; Kappelman et al., 2006; Keider, 1984; Keil et al., 1998; Kerzner, 2014;

McManus and Wood-Harper, 2008; Meier, 2008; Morris and Hough, 1987; Pinto and Slevin, 1987; Schmidt, 2023b; Schmidt et al., 2001; Standish, 2014; Verner et al., 2008; Yardley, 2002; Yeo, 2002). The failure factors are:

(1) Project objectives
(2) Senior management
(3) Planning
(4) Requirements
(5) Execution and control
(6) Technology
(7) Development method
(8) User involvement
(9) Staff
(10) Contractors
(11) Risk management

Failure factors are areas of activity where causes of project failure may originate (Pinto and Slevin, 1987; Schmidt, 2023b). For example, "unclear objectives", "unrealistic objectives", and "changing objectives" are causes of IT project failure that belong to the area of activity: "Project objectives".

The analysis below in Sections 3.4.1 through 3.4.11 is structured according to general failure factors systematically used as hypotheses of project failure. Promising hypotheses are evaluated by causal diagrams, event-history maps, and descriptive inference according to Waldner's completeness standard for process tracing (2012; 2015; 2022). Hypotheses are promising if they can contribute to explaining the course of events and the observational data generated from project documents, official reports, and interviews.

3.4.1. Project objectives

In this section, we evaluate the hypothesis that unclear, changing, or unrealistic objectives were contributing causes of the termination of the POLSAG project (Baker et al., 1983; Brown, 2001; Charette, 2005; Chua, 2009; Cole, 1995; El-Emam and Koru,

2008; Ewusi-Mensah, 2003; Glass, 1998; Kappelman et al., 2006; Keider, 1984; Keil et al., 1998; Kerzner, 2014; McManus and Wood-Harper, 2008; Morris and Hough, 1987; Pinto and Slevin, 1987; Schmidt, 2023b; Schmidt et al., 2001; Standish, 2014; Verner et al., 2008; Yardley, 2002; Yeo, 2002).

Early analyses in 1995 by the Danish Ministry of Finance and management consultants concluded that the Danish police could make significant staff reductions by increased use of IT (The Danish Ministry of Finance and PA Consulting Group, 1995). The workforce could be reduced by 470 out of 13.700, and administrative work could be removed from police officers corresponding to 380 full-time equivalents (FTEs). A predecessor of POLSAG was then incrementally developed by a combination of existing staff and external consultants, in total, a team of roughly 10 staff.

In 2002, management consultants assessed that the POLSAG predecessor was technologically outdated (although still operational in 2023) and recommended developing a new system, which additionally supported a) digital communications with external parties, including citizens, b) video and sound media, and c) data access across police constabularies.

In the time up to 2006, when the POLSAG contract was signed, six external management consultant companies had been involved in preparing IT strategies, high level visions for the new system, a master plan, a business case, and a risk analysis. The rationale behind the vision and high-level requirements, or how they would support the achievement of desired rationalisation effects expected from the new system, are not clearly described in the project documents. This may imply that this rationale was not deemed important enough to document as part of the project documentation, or that possibly the rationale was non-existent or not thoroughly analysed.

A present-day review of the POLSAG project would not find the documented project objectives clear, stable, and realistic. The POLSAG project objectives did not support effective project management in prioritising resources and managing scope inflation. Unclear, unstable, and unrealistic objectives can lead to scope inflation, which can cause cost overruns and delays, and thereby lead to project failure. However,

(1) The reasons for terminating the project did not include any shortfall on meeting objectives. Neither did the significant descoping of advanced features (e.g., digital communication with citizens and video data) play a role.
(2) The reasons for terminating the project also did not include the significant delays, cost increase and functional scope inflation.
(3) Termination was not caused by a shortfall in functionality. The system featured the functionality wanted and specified by the police. This is documented by the fact that the software system passed the contractual acceptance test. The acceptance test was, as often in comparable contracts, a test of the functionality of the software in a test environment using test data based on testing scenarios agreed between customer and contractor. The positive outcome of the acceptance test demonstrates that the project delivered software that satisfied the Police's needs and requirements despite unclear, changing and unrealistic objectives.

Therefore unclear, changing, and unrealistic objectives were not the root cause of the failure of the POLSAG project. This can be concluded, because the postmortem must determine the causes of the POLSAG project termination, a special case of project failure. When the project was terminated, the POLSAG system had passed the functional acceptance test. This means that the project managed to deliver an IT-system that fulfilled the functional requirements, albeit with inflated scope and significant cost overruns.

3.4.2. Senior Management

In this section, we analyse the hypothesis that the causes of failure and termination of the POLSAG project can be traced back to actions and omissions by senior management (Brown, 2001; Chua, 2009; Cole, 1995; El-Emam and Koru, 2008; Ewusi-Mensah, 2003; Fowler and Horan, 2007; Glass, 1998; Jones, 1995; Kappelman et al., 2006; Keil et al., 1998; Kerzner, 2014; McManus and Wood-Harper, 2008; Meier, 2008; Morris and Hough, 1987; Pinto and Slevin, 1987; Schmidt, 2023b; Schmidt et al., 2001; Standish, 2014; Verner et al., 2008; Yardley, 2002; Yeo, 2002).

Lack of management involvement, commitment, and support by senior management can lead to project failure. IT projects often require actions by persons and organisa-

tions that are not within the management scope of the temporary project organisation itself, and therefore IT projects often require active support, involvement, and commitment by senior management. See, for example, Glass (1998). Additionally, senior management involvement, commitment, and support are needed when an IT project can no longer progress within agreed parameters, and therefore changes to budget, scope, or schedule are needed. This is frequent in IT projects (Flyvbjerg et al., 2022).

There was no lack of management commitment and support to the POLSAG project. The Chief of Police participated in most of the steering committee meetings, supported requests for additional funding for the project, and offered to participate in more meetings, if needed.

It would have been beneficial for the project if the steering committee had engaged more in the project when the schedule and budget were at risk. A more actively engaged steering committee with the right project governance competences, especially in the planning phase of the project, would likely have made a difference. However, there is no foundation in the data about the course of events for concluding that lack of top management commitment, involvement or support was a contributing cause of failure. A compelling causal chain linking lack of top management commitment, involvement, or support to one or more of the six reasons for closing the project cannot be identified based on the data available. Additionally, claims in hindsight that management could have taken different actions that might have led to project success rather than failure are circular, uninformative, and unhelpful for improving future practice (Reason, 1990).

3.4.3. Planning

In this section, we evaluate the hypothesis that inadequate planning was a contributing cause of the POLSAG failure and termination (Baker et al., 1983; Brown, 2001; Cerpa and Verner, 2009; Charette, 2005; Chua, 2009; Cole, 1995; El-Emam and Koru, 2008; Ewusi-Mensah, 2003; Glass, 1998; Jones, 1995; Kappelman et al., 2006; Keider, 1984; Kerzner, 2014; McManus and Wood-Harper, 2008; Morris and Hough, 1987; Pinto and Slevin, 1987; Schmidt, 2023b; Standish, 2014; Verner et al., 2008; Yardley, 2002; Yeo, 2002). In the context of the postmortem analysis presented here, "planning"

covers activities that take place before the final commitment to build. This means that planning also includes feasibility analyses. For additional detail about the analysis of project data regarding planning, see Appendix A.

One of the six explicit reasons for terminating the POLSAG project was an unsatisfactory business case, so it is relevant to evaluate this hypothesis by trying to identify a causal chain of events from inadequate planning to an unsatisfactory business case. Planning is much more than budgeting. But in this paper, we focus mainly on budgets and economic performance because they expose problems of estimation, incompleteness of project plans, escalation of scope, escalation of costs, and escalation of recurring costs that can lead to an unsatisfactory business case.

Four global management consultant companies and one local were involved in the planning of the POLSAG project. In 2002, consultants estimated the project cost would be in the range from $21m to $28m. When the project was terminated, the approved budget was $59m, and cost projections had exceeded $105m. For comparison, Flyvbjerg and Gardner (2023) report average IT project cost overruns of 73%, and that 18% of projects on average cost 457% of the original budget.

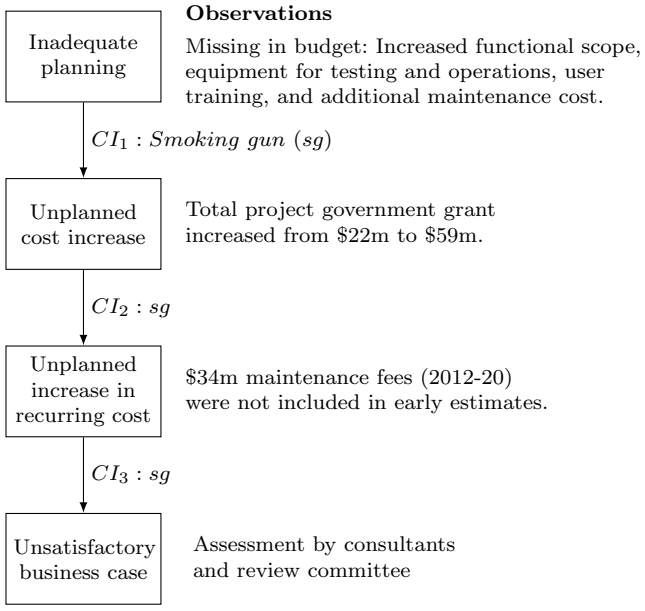

Figure 2. Event-history map and implied causal graph, planning, POLSAG. *CI*: Causal Inference

Figure 2 shows how incomplete and inadequate planning led to an unsatisfactory business case for POLSAG. The original project plan had insufficient budget for the

increased functional scope of the POLSAG software. In addition, several costly items were simply missing from the budget. Missing items included testing equipment, hardware and software for operating the POLSAG system, and the necessary training of thousands of employees in the use of the complex system. The additional cost of constructing the POLSAG software system also caused additional recurring costs. As an example, the additional $10m allocated for additional software features in 2007 would double in four years because of the annual maintenance fee for the software extensions. Added testing and operations equipment would also generate additional recurring maintenance costs. The effects of recurring costs on the business case were not documented until the reconstruction of the business case in 2011 by management consultants. The management consultants that reconstructed the business case were different from the ones involved in the original project planning. See also Appendix A.

The causal inferences, c_1, c_2, c_3 in Figure 2 are not complex, once they are identified: Some indispensable elements were not included in the budget, so spending exceeded the budget. Most of the new elements came with indispensable recurring maintenance costs, adding to the cost overrun. Calling the inference c_1 a "Smoking gun" in Figure 2 means that the first node is a sufficient, but not necessary condition for the second node (Bennett, 2008; Bennett and Checkel, 2014; Collier, 2011). In other words, there may be other causes of unplanned cost increase than inadequate planning. However, in our assessment, inadequate planning is the best explanation for the unplanned cost increase, given the evidence (see Figure 2) and the research literature on IT project failure (Schmidt, 2023b). We therefore identify inadequate planning as a contributing root cause.

To demonstrate the importance of IT project planning and preparation, the business case was reconstructed (simulated) for the point in the time of the contract signature (Schmidt, 2020), see Appendix A. The reconstruction was based on the benefit calculations used by management consultants prior to the project termination.

The business case reconstruction shows that the project was probably unattractive already when the contract was signed, and that benefits most likely could have been achieved more cost-effectively by a less ambitious project. This shows that completeness in planning can be important, not just to avoid cost overruns and project failure,

but also for project design and the selection of the most attractive design alternative. For the POLSAG postmortem analysis, the doubtful attractiveness of the project when taking planning omissions into account also weakens rival hypotheses that place root causes of failure in the area of execution and control. See Section 3.4.5.

3.4.4. Requirements

In this section, the hypothesis to be evaluated is that contributing root causes of the POLSAG project failure and termination originate in the area of requirements (Brown, 2001; Cerpa and Verner, 2009; Charette, 2005; Chua, 2009; El-Emam and Koru, 2008; Ewusi-Mensah, 2003; Kappelman et al., 2006; Keil et al., 1998; Kerzner, 2014; McManus and Wood-Harper, 2008; Meier, 2008; Morris and Hough, 1987; Schmidt, 2023b; Schmidt et al., 2001; Standish, 2014; Verner et al., 2008; Yeo, 2002).

Requirements elicitation and requirements management are difficult disciplines in software engineering. According to Jones (1995), a good requirement specification is initially 80% correct, and requirements change from 1% to 3% per month in the analysis and development phases of software development projects. If project plans do not take scope inflation into account, unclear and changing requirements can lead to cost overruns, delays and project failure.

The high-level requirements for the predecessor of POLSAG were IT support for case file management and the 24-hour police report produced daily in all constabularies. The IT system developed was intended to support the reduction of manual work for these functions significantly.

For POLSAG, the main high level requirement was a "1:1" requirement, meaning reproducing the functionality of the predecessor system, a 50-100 staff year IT system developed incrementally by a small team. In addition to the 1:1 requirement, 1,600 new requirements were elicited with extensive user involvement.

The POLSAG project suffered significant scope inflation that caused sizeable budget overruns and delays. However, the POLSAG software passed the contractual acceptance test, so the termination of the POLSAG project was not caused by unclear or changing requirements. The acceptance test was, as often in comparable contracts, a test of the functionality of the software in a test environment using test data based

on testing scenarios agreed between customer and contractor. The positive outcome of the acceptance test demonstrates that the project delivered software that satisfied the Police's needs and functional requirements. Furthermore, the reasons given for terminating the project make no mention of shortfall in functionality. Additionally, it is possible that POLSAG could have passed a non-functional requirements test (response times, capacity etc.) if the project had not been terminated.

This conclusion shows the importance of conducting systematic postmortem reviews of failed IT projects. Some observers have commented that scope inflation and poor requirements management were the cause of the POLSAG project failure. Systematic postmortem analysis shows that this was not the case, or at least that requirements management was not a root cause of termination, because we cannot convincingly identify a chain of events from inadequacies in the area of requirements to project termination that matches the project data. In this postmortem analysis, a root cause is defined as "the most basic cause that can reasonably be identified...". The inflation of requirements and the excessive modifications of the core of the COTS system had other and more basic causes than the area of "requirements", please see the Sections 3.4.6, Technology, and 3.4.7, Development method.

3.4.5. Execution and control

IT project execution and control can almost always be improved in hindsight. In this section, the task is not to identify what could have been done differently and better, but to test the hypothesis that root causes of failure and termination of POLSAG can be found in the area of execution and control (Baker et al., 1983; Brown, 2001; Cerpa and Verner, 2009; Chua, 2009; Cole, 1995; El-Emam and Koru, 2008; Ewusi-Mensah, 2003; Fowler and Horan, 2007; Glass, 1998; Jones, 1995; Kappelman et al., 2006; Keider, 1984; Kerzner, 2014; McManus and Wood-Harper, 2008; Meier, 2008; Morris and Hough, 1987; Pinto and Slevin, 1987; Schmidt, 2023b; Schmidt et al., 2001; Standish, 2014; Verner et al., 2008; Yardley, 2002; Yeo, 2002).

Execution is about more than management and control of the essentials: scope, budget, and schedule. However, meticulously tracking how the project managed changes in scope, budget and schedule can give an indication of the quality of the execution and

control that can be based on project documents alone, without assuming any particular project management methodology. For the POLSAG project, we cannot use adherence or deviation from any specific project management methodology as an indication of the quality of execution and control, since no such methodology is specified in the project documentation, and none was mandatory for Danish government projects at the time. We have identified 32 relevant project management meetings from mid 2007 to the end of 2008 and tracked the discussions and decisions made about changes to scope, budget and schedule, which actions were taken, and how they were documented. This detailed study of how the POLSAG project documents track the management of scope, cost, schedule, and changes does not support a conclusion that inadequate execution and control, inadequate change management, or inappropriate method significantly contributed to the six reasons for terminating the project. The POLSAG software passed the acceptance test, albeit with significant changes to scope and extensions of budget and schedule. Additionally, the inflation of scope and budget was not driven by inadequate execution and control, but by root causes found in planning (Section 3.4.3), technology (Section 3.4.6), and software development method (3.4.7). Given the high attention often given to the execution and control of IT projects, it may seem surprising that the root causes of project failure were not found in this phase of the project. This postmortem finding supports the general consideration that, in the words of the late Peter Morris, "projects fail at the front-end" (personal communication, 2018).

3.4.6. Technology

In this section, we explore the hypothesis that the root causes of the POLSAG project failure and termination are to be found in the area of technology (Chua, 2009; Cole, 1995; El-Emam and Koru, 2008; Ewusi-Mensah, 2003; Glass, 1998; Keil et al., 1998; Kerzner, 2014; McManus and Wood-Harper, 2008; Meier, 2008; Morris and Hough, 1987; Schmidt, 2023b; Schmidt et al., 2001; Verner et al., 2008; Yardley, 2002; Yeo, 2002), including process technology. Process technology, in this context, means competence and knowledge of how to construct a large-scale IT system according to best practice and textbook engineering heuristics.

POLSAG contained many technology elements, including process technology, that were new to the organisations involved, including 1) COTS system implementation, 2) Service Oriented Architecture (SOA) integration layer, 3) Client/Server application over wide area network (WAN) connections, and 4) Reuse software process (process technology for COTS), see e.g., (Sommerville, 2016). The COTS software turned out to be unsuited for satisfactory operation with more than 10,000 users accessing a single database via a low-capacity network. The project also involved excessive customisation, which led to more than 1,000 detailed software modification requirements and the addition of 500 new database tables to the 300 tables of COTS system in its standard configuration.

Multiple external consultants were involved in the project preparation, but the project documents include no in-depth system engineering or solution design, but fragmented assessments of individual system elements. During project execution, multiple significant changes were made to the system design.

The immature combination of technologies, rather than a clear and proven solution design, meant that the responsibility for the functioning of the combined solution, including end-to-end response times, was not clearly allocated to the contractors. This entailed a significant technical risk for the Police.

The COTS software was excessively customised. 140,000 hours were spent on modifying the system, far more than any other project by the contractors, and far beyond the experience of the software developers involved. This had a serious negative impact on the software quality, which was assessed by three external management consulting companies.

The high degree of customisation also led to a situation where regular upgrades of the standard version COTS system - normally an advantage of COTS systems - would be difficult, expensive, and maybe even impossible. This entailed a risk of obsolescence of the POLSAG system, as assessed by external consultants. Obsolescence risk was one of the six reasons for terminating the POLSAG projects.

This leads to the conclusion that technology - its immaturity, its newness to the organisation, and excessive customisation - were contributing causes of the failure and termination of the project, see Figure 3 and Table 2.

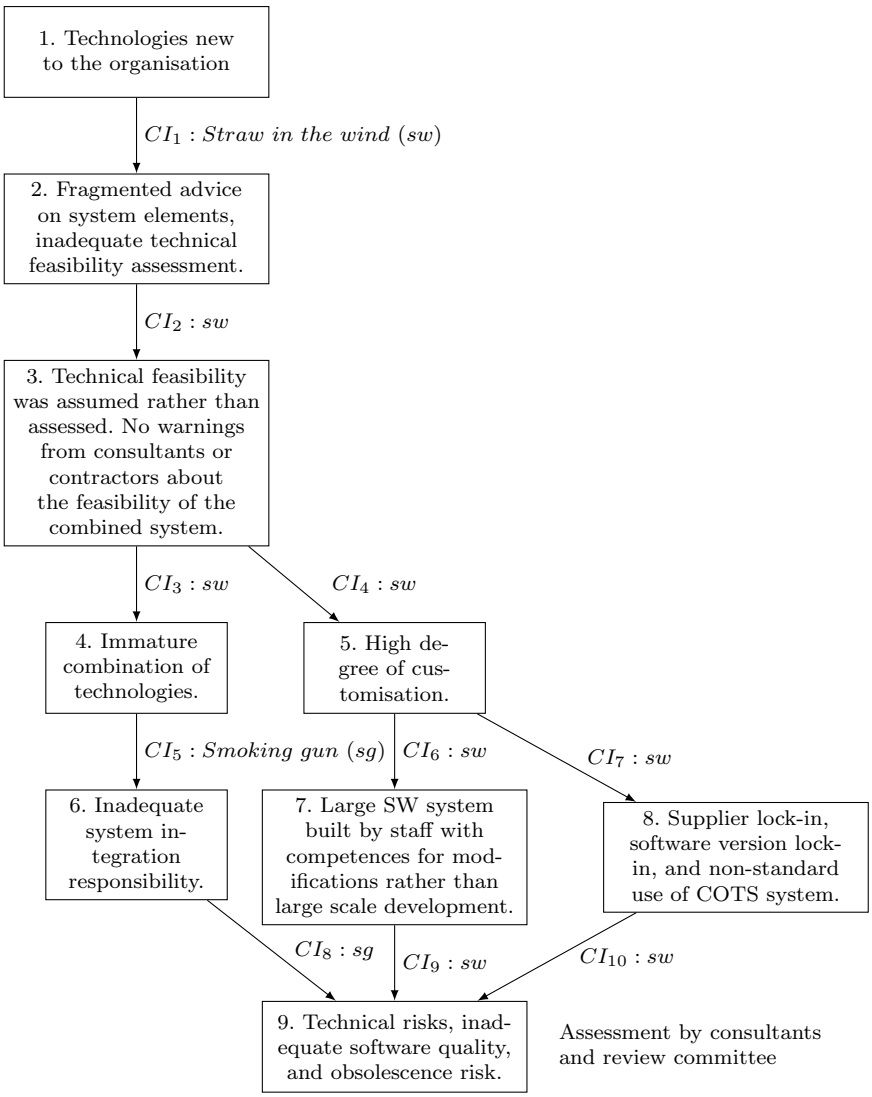

Figure 3. Event-history map and implied causal graph: technology, POLSAG. *CI*: Causal Inference

3.4.7. Development method

In this section, we explore the hypothesis that the software development method followed was a contributing root cause of the failure and termination of the POLSAG project (Brown, 2001; Cerpa and Verner, 2009; McManus and Wood-Harper, 2008; Meier, 2008; Morris and Hough, 1987; Schmidt, 2023b; Verner et al., 2008).

POLSAG was originally conceived as a greenfield, or bespoke, software development project. It was strongly suggested to the Police that using an existing, customisable case file management system, a COTS system, would be more advantageous. The Police had no, or very little, experience with implementing COTS systems.

Table 2. Observations, novelty of technology to the organisation.

	Node in Figure 3	**Observations**
1,	Technologies new to the organisation	COTS replacement of legacy system and COTS integration with mainframe legacy systems. Both new to the Police.
2,	Fragmented advice on system elements, inadequate technical feasibility assessment.	Advice documented in project documents did not address all relevant technical risks.
3.	Technical feasibility was assumed rather than assessed. No warnings from consultants or contractors about the technical feasibility of the combined system.	No project documents with explicit assessment of technical feasibility at system level. Guidance given was not followed, e.g., advice to show restraint with COTS modifications.
4.	Immature combination of technologies.	Immature combination of technologies: COTS, service oriented architecture (SOA), mainframe integration, Client/Server over low capacity wide area network (WAN).
5.	High degree of customisation.	High degree of customisation: Requirement to implement legacy functionality 1:1 in new COTS system. In addition, more than 1,000 detailed requirements. COTS database increased from 300 to 800 tables.
6,	Inadequate system integration responsibility.	Inadequate contractual remedies, including no contractual system commitments by contractors for end-to-end performance of system.
7.	Large SW system built by staff with competences for modifications rather than large scale development.	Large software system built by staff only experienced in moderate modification. Planning of software development did not match the size of the system. The system was not an adapted COTS system, but in effect a new, complex, and sizeable software system (140,000 hours).
8.	Supplier lock-in, software version lock-in, and non-standard use of COTS system.	Supplier lock-in, non-standard use of COTS, and version lock-in. It would not be possible to upgrade the system to later versions of the COTS system.
9.	Technical risks, inadequate software quality, and obsolescence risk.	Assessment by consultants and review committee

Advantages of COTS systems include: a) the system can be tested in advance, b) fewer specification errors than greenfield software systems, c) proven value in operation at other customers, d) the cost of future development of the system is shared with other customers, and typically paid by a subscription type maintenance fee that includes upgrades, e) the cost of support and maintenance is to some extent shared with other customers, f) COTS systems often contain functionality and features that customers find valuable in the future, g) faster organisational implementation and no, or limited,

need for developing new software, h) in the best cases, the software can be adapted by configuration rather than by modifying the source code, and i) COTS projects can have lower risk than greenfield projects. COTS projects are, for example, less sensitive to risks associated with specification errors, estimation errors, and technical risks. See e.g., Sommerville (2016).

Disadvantages and risks of COTS systems include a) the systems and source codes are normally proprietary to the supplier, which entails supplier lock-in and reduced control of the software, b) COTS systems may not have a perfect fit with the customer requirements, c) COTS systems can normally only be modified, configured and adapted by the supplier, or by companies appointed by the supplier, d) extensive customisation and modification can make upgrades impossible or very expensive, e) COTS systems have track records, and therefore history. This means that COTS systems are often not based on, or compatible with, the latest technology.

Good practice, or good software engineering heuristics, for software development based on COTS (e.g., Sommerville, 2016) is to first define requirements and second, find the COTS system that best matches the requirement. The crucial third step: *adapting requirements to fit the selected COTS system*, was not taken in the POLSAG project. Figure 4 shows the cost effects of inappropriate software process in the POLSAG project. Figure 3 shows the related problems of version lock-in, quality issues, technical risks, and obsolescence risks, which can be the consequences of inadequate system engineering and excessive scale.

Good practice for COTS-based system development can be found in software development textbooks at least as early as 2004 (e.g., Sommerville), but it was not common knowledge among IT executives, consultants, and project managers. The project documents, including consultants' reports, show no record of any discussions of the software development methodology implications of the decision to use a COTS system for POLSAG. Second, the textbooks do not unfold the possible consequences of disregarding good practice in the way that the POLSAG project postmortem does. The purpose of the POLSAG postmortem is not to assign responsibility to individuals in hindsight, but to present knowledge in context that can be used to improve future practices based on systematic analysis. Experience from our occasional assignments

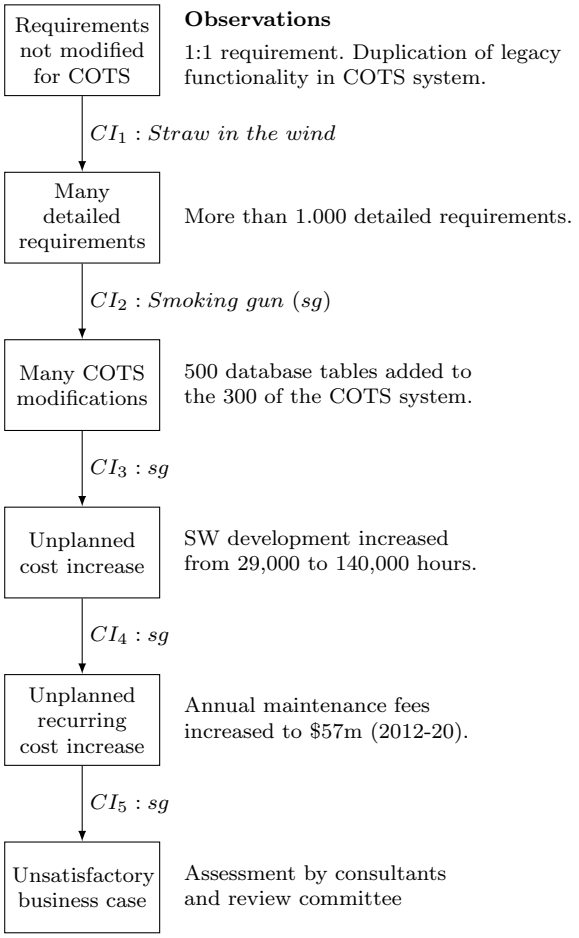

Figure 4. Event-history map and implied causal graph, development method. *CI*: Causal Inference

as external reviewers of large IT projects tells us that good engineering heuristics for COTS-based software development are still not common knowledge.

3.4.8. User involvement

In this section, we assess the hypothesis that the causes of the POLSAG project failure were related to lack of appropriate user involvement (Brown, 2001; Cerpa and Verner, 2009; Chua, 2009; El-Emam and Koru, 2008; Fowler and Horan, 2007; Keider, 1984; Keil et al., 1998; Kerzner, 2014; McManus and Wood-Harper, 2008; Pinto and Slevin, 1987; Schmidt, 2023b; Schmidt et al., 2001; Standish, 2014; Verner et al., 2008; Yardley, 2002; Yeo, 2002). Possible causes of IT project failure related to user involvement include lack of user input and user involvement, lack of user training, and failure to manage user expectations.

The POLSAG project involved users extensively in workshops, requirement elicitation, and testing. Good practice for COTS projects includes adapting requirements to the standard functionality of the COTS system to minimise bespoke modifications. There is no evidence that there was a dialogue with users to align expectations about using the new COTS system "as-is" as much as possible. The highest level of management stressed that the new system must include the elements requested by users and that the new system must not be "heavier" to use than the old one (steering committee meeting minutes from December 2006, and August 2008). This was a potentially risky approach, and higher awareness of good practices for COTS software development might have led to a different approach to user involvement. However, the POLSAG project involved users extensively, and in our assessment, a lack of user involvement was not a root cause of project failure. This conclusion is additionally supported by the fact that POLSAG software passed the contractual acceptance test of functionality.

3.4.9. Staff

In this section, the hypothesis evaluated is that staff skills, motivation, or insufficient staff would have played a role in the failure and termination of the POLSAG project (Baker et al., 1983; Cerpa and Verner, 2009; El-Emam and Koru, 2008; Ewusi-Mensah, 2003; Fowler and Horan, 2007; Jones, 1995; Kappelman et al., 2006; Keider, 1984; Keil et al., 1998; Kerzner, 2014; McManus and Wood-Harper, 2008; Meier, 2008; Morris and Hough, 1987; Pinto and Slevin, 1987; Schmidt, 2023b; Schmidt et al., 2001; Standish, 2014).

There seems to be a propensity to initially attribute causes of failure to individuals (Pan and Flynn, 2003; Reason, 1990), which, of course, sometimes may be correct. But arguments that more competent staff would have done things differently, and therefore would not have failed, are circular and not useful for developing recommendations for improving general practice. In connection with the POLSAG postmortem the CVs of some of the key project management staff were assessed, and it was concluded that the levels of skill and experience did not deviate from those of comparable individuals in comparable settings. Based on this, and the lack of evidence to the contrary in the

project documentation, our analysis does not show that insufficient staff, unmotivated staff, or lack of skills and experience of individual staff members were contributing root causes of project failure and termination of the POLSAG project.

3.4.10. Contractors

In this section, we assess the hypothesis that the root causes of the POLSAG project failure originate in the area of contractors and contracting (Brown, 2001; Chua, 2009; Glass, 1998; Jones, 1995; Keider, 1984; Kerzner, 2014; McManus and Wood-Harper, 2008; Meier, 2008; Morris and Hough, 1987; Pinto and Slevin, 1987; Schmidt, 2023b; Yeo, 2002).

Contracting and contract management are extensive topics in their own right (Cropper et al., 2008; Maurer, 2010; Suprapto et al., 2016). For the POLSAG postmortem, we have looked specifically at contractor performance, underestimation by contractors and consultants, and lack of experience in contractor management. These aspects of contracting are potential causes of failure, according to the literature (Schmidt, 2023b).

There is no doubt that the contractors performed unsatisfactorily, and that contractors and consultants seriously underestimated the project. This conclusion is supported by the fact that the terms of the contract termination were settled out of court. Furthermore, the POLSAG project was loss-making for the contractors according to their financial statements (see Figures 5 and 6). It is also likely that the Police lost confidence in the contractor's ability to deliver the project, indicated by one of the six reasons for terminating the project, which was "supplier risk: the cooperation between the contractor and the sub-contractor was unsatisfactory". However, the POLSAG system did pass the contractual acceptance test, a functional test, and the POLSAG contractors had delivered COTS projects elsewhere that did not fail. So, even though inadequate contractor performance may have contributed to the project failure, it does not fit the analysis as a root cause of its own. The argument that better contractor performance would have prevented project failure is circular and uninformative for improving practice.

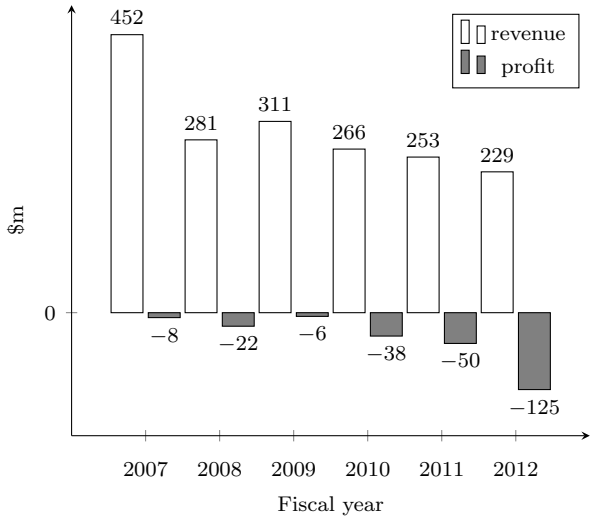

Figure 5. Contractor financial performance during project (financial statements)

3.4.11. Risk management

In this section, we assess inadequate risk management as a potential root cause of failure (Cerpa and Verner, 2009; Kerzner, 2014; Schmidt, 2023b; Yeo, 2002). Risk management in the POLSAG project was rudimentary. This was not uncommon for comparable IT projects at the time. But inadequate risk management in itself was not a root cause of project failure and termination in our analysis. The project ran very small risk contingency budgets, but significant additional funding was granted to the project more than once. The technical problems could possibly have been mitigated better with appropriate risk management. But the technical problems that occurred in the project execution phase could even more appropriately have been eliminated in the project planning phase by testing the COTS system performance on the available network infrastructure, and by testing the functionality of the COTS systems against relevant use cases. However, before termination the project delivered a system that passed the contractual acceptance test, albeit at inflated costs, with significant delays and in a quality that was unproven in operations. Still, the postmortem analysis does not support the conclusion that inadequate risk management was a contributing root cause of project failure and termination of its own.

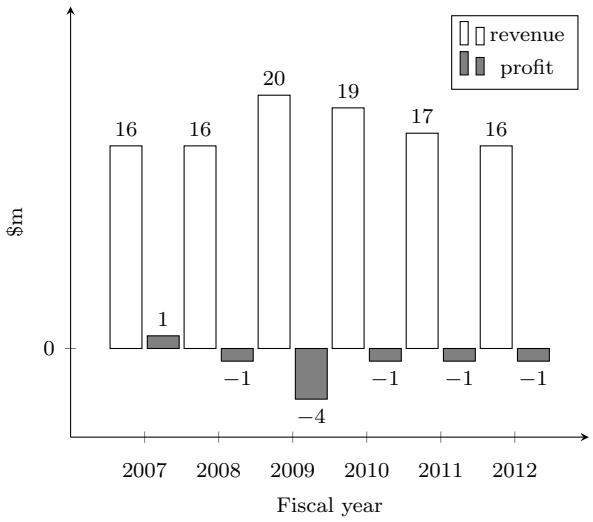

Figure 6. Sub-contractor financial performance during project (financial statements)

3.5. Hypothesis testing conclusions

We have evaluated the known failure factors 1 through 11 in Section 3.4 as root cause hypotheses for the POLSAG project failure.

The project documents show that the POLSAG system passed the functional acceptance test in the contract. This helps block possible causal paths from the failure factors (see section 3.4): project objectives (1), senior management (2), requirements (4), execution and control (5), user involvement (8), staff (9), contractors (10) and risk (11) to the reasons 1 through 6 in Section 3.2 given by the Danish government for terminating the project.

The reasons for terminating the project turn out to be related to the full scale organisational implementation, uncertainty about the technical quality of the system in operation, and to the feasibility of the system in the operations and maintenance phase. The hypothesis testing shows that the causal paths to the reasons for terminating the project originate in the failure factors from the literature: planning (3), technology (6), and development method (7). This conclusion is well supported by the project data.

The three identified root causes explain different aspects of the reasons for terminating the project. they explain different subsets of the project data, and they are

compatible. Therefore the tree identified root causes are considered as contributing root causes (cf. Section 2.2, Definitions) rather than rival explanations.

3.6. Postmortem findings

In conclusion, the postmortem shows *(RQ1)* that the failure factors and causes from the literature that can be most convincingly causally linked to the reasons for terminating the project are:

(1) Inadequate planning. The postmortem shows that critical and costly elements were missing in the project plan, including testing equipment, testing activities, and user training. Other elements were included, but underestimated.
(2) The technology was new to the organisations involved. The postmortem shows how the project relied on faulty assumptions about the capacity and topology of the technical infrastructure and the functionality of major components. This could have been discovered in the planning phase by bringing in the appropriate technical competences at the front-end of the project.
(3) Inappropriate software development method and process. More specifically, software process mismatch for projects based on commercially off-the-shelf (COTS) software (Sommerville, 2016). The postmortem analysis shows why, and how, a software process mismatch led to massive scope inflation, delays, significant cost escalation, obsolescence risk, and unsustainable maintenance costs.

The POLSAG postmortem findings lead to two additional conclusions: First, the causes of project failure can be traced back to the preparation phase, and the problems could have been eliminated or mitigated during project preparation (Ahonen and Savolainen, 2010). This supports the general consideration that "projects fail at the front-end" (Peter Morris, personal communication, 2018). Second, the necessary heuristics and theory to avoid the problems that led to the POLSAG project failure are available in the research literature, in textbooks, and through experienced practitioners. This points to the desirability of competence development for senior managers and other contributors to IT projects.

3.7. Results, recommendations and preventive actions

Recommendations for practice improvements derived from the POLSAG postmortem *(RQ2)* include:

(1) The completeness of plans, including budgets, must be validated as part of the project preparation and planning. The postmortem shows that even experienced and professional IT organisations may launch projects with incomplete plans and budgets. There may be strategic reasons for this (Flyvbjerg, 2007, 2009a), or implicit assumptions that certain planning items are covered outside the project plans and budgets. Practice improvements should include thorough reviews of the completeness of plans and budgets before committing to build. Assumptions about e.g., user training, testing and operations equipment being covered and planned outside the scope of the project should be verified in the preparation phase, not discovered in the execution phase, causing inflation in the scope of work and cost. If no suitable reference class (Flyvbjerg, 2007; Kahneman and Tversky, 1977) for software project scope inflation is available, industrial averages can be used instead for validating the adequacy of contingency allocation. For example, Jones (1995) found that initial requirement specifications are 80% correct and change from 1% to 3% per month during the analysis and development phases of software development projects.

(2) Technology assumptions must be verified before the launch of the execution phase. The capacity and performance of COTS systems can be verified in the preparation phase by testing, technical analysis, and calculations, and by consulting reference sites. The capacity and performance of servers and network infrastructure can be verified in the same way. It is important to verify and validate assumptions to ensure the feasibility of project designs. In the POLSAG case, for example, one database per constabulary would likely have been more feasible performance-wise than one centralised national database, which the capacity of the network infrastructure could not support. In the preparation phase, this discovery would have been a solvable problem. Finding out in the execution phase was detrimental.

(3) Process technology, including software development methods, should be part of the assessment of organisational feasibility for IT projects.
(4) Competence development for individuals with senior management responsibilities and key expert roles in major IT projects should be a requirement prior to the preparation of major IT projects.

4. Discussion and limitations

The postmortem analysis of the POLSAG project failure using process tracing and the project as the unit of analysis shows that it is possible to maintain a focus on practices rather than on actions and omissions of individuals. This is a limitation if placing responsibility for errors on individuals is what is needed. However, it is also a strength, because it focuses the analysis on causes and effects at project level, which can be helpful for devising improvements of project management practices.

The use of factors and causes from the literature as generic hypotheses for systematic analysis limits the analysis to factors that are "managerially controllable" (Pinto and Slevin, 1987; Schmidt, 2023b). On the one hand, this is a limitation if a structural or institutional perspective on project failure is taken. On the other hand, if input for practice changes and contributions to project management theory that is applicable in practice are needed, this feature is a strength.

The focus on strictly causal inference in the approach entails a lack of consideration of motives, purposes and intentions behind actions taken, as in Flyvbjerg's "strategic misrepresentation" (Flyvbjerg, 2007), for example. Strategic misrepresentation in the POLSAG project could - as a purely hypothetical example - be deliberate omission of costly items from the initial budget in order to improve the likelihood of government approval of the project. Research by other methodologies into possible strategic representation in the POLSAG project would, however, find a strong foundation in the postmortem analysis presented.

The focus on root causes that are within the scope of management to fix (Atkins, 2001; Paradies and Busch, 1988) and the exclusive consideration of managerially controllable factors (Pinto and Slevin, 1987; Schmidt, 2023b) will not provide *structural*

answers to Cobb's paradox: "We know why software projects fail; we know how to prevent their failure, so why do they still fail?" (Maddison, 2011). Such structural answers might, for example, include general IT competence development for senior managers, different ways of engaging external experts and consultants, and stipulated planning requirements prior to the funding of IT projects. Postmortem analyses, as the one presented here, will, however, offer contributions to contextual improvements by proposing preventive actions within corporate or agency boundaries.

In hindsight, there is no doubt that different efforts from consultants could have led to a better result for the project. The same can be said for the customer (the Danish Police), who had one of the largest and most experienced IT departments in a country with a top ranking in digitalisation (United Nations, 2022). The same can be said about the contractor, an experienced international provider of software and services. The same can also be said about the subcontractor, who at the time had a significant share of the local market for case-file software.

The consultants, the customer's IT department, the contractor, and the subcontractor all could have - and should have - done better than they did. They all failed.

However, from a learning perspective, and from a project postmortem perspective, the question is: *What* should they have done differently and better? And what should be done differently and better next time?

The root causes found in postmortem analysis are: (1) Inadequate planning, including feasibility assessment, (2) use of technology that was new to the organisation, and (3) inappropriate software development method; COTS implementation is different from bespoke software development. These root causes can all be addressed by various choices of corrective actions, including (a) competence development of own IT management and staff, (b) the use of more specialised consultants, (c) different contractor selection criteria, (d) more thorough feasibility studies, especially for technologies that are new to the organisation. The list can be extended. The list of possible corrective actions may also be different for other organisations, who wish to take advantage of the POLSAG postmortem conclusions.

An advantage of the causal diagrams is that changes can be counterfactually "modelled into" the actual events, so that alternative preventive actions can be analysed.

For example, should we introduce systematic reviews of feasibility studies?, or should we invest in competence development in performing feasibility studies?. Should we find more specialised consultants?, or should we decide to make prototyping mandatory for new technologies?, or both?

5. Conclusions

The risks of inadequate planning, use of new technology, and inappropriate software development method have been demonstrated in specific terms in the context of a major IT project using systematic IT project postmortem analysis in this paper. The general conclusion that projects fail at the front-end is supported by the postmortem analysis in this paper because the three main causes of the POLSAG failure could all have been eliminated or mitigated in the project preparation phase. These findings can be used directly in IT project risk management, project preparation, and competence development. Additionally, the findings can be used for theory development regarding the importance of the front-end in project performance. The combination of the case-specific findings and general causes of IT project failure from the literature strengthens the general validity of the postmortem conclusions. The systematic consideration and refutation of rival hypotheses based on the literature add to the strength of the postmortem conclusions.

5.1. Contributions to practice

The empirical specificity of the postmortem analysis is informative for devising practice changes to prevent similar failures in the future, and for understanding early warning signs and defining red flags. The specific root causes of the POLSAG failure identified in the postmortem analysis are directly applicable by practitioners for improving IT project management practices, particularly in connection with IT projects that are based on COTS systems. The postmortem analysis method itself can also be used directly by practice: (1) Using the known failure factors of IT project systematically for identifying root causes, (2) using process tracing for identifying the causal mecha-

nisms behind the actual failure, and (3) designing preventive actions that could have prevented the failure - for the benefit of future projects.

Preventive actions proposed based on the POLSAG postmortem include: a) completeness review of plans and budgets in the project planning phase, b) the use of reference classes in IT project planning, c) verification of all technological assumptions in the planning phase, d) inclusion of organisational competences in software development methods in organisational feasibility assessments, e) competence development in IT project process technology for managers and participants prior to the planning and preparation of major IT projects.

5.2. Contributions to theory

The postmortem analysis presented is an "explanatory" case study (Sarker, 2021; Yin, 2018). The postmortem analysis tests the failure factor literature's theoretical claims (Sarker, 2021), and the analysis adds to theory by exposing contextual and general causal mechanisms that link causes of failure to project failure.

The demonstration of the systematic approach to IT project postmortems that combines leading theory of process tracing and causal modelling with the literature on IT project failure factors in this paper is new and original. The demonstration of the postmortem approach shows how project postmortems, which are ipso facto single-case studies, can support generally valid conclusions about the causes of IT project failure. Additionally, the paper is a contribution to potential theory development (Sutton and Staw, 1995; Weick, 1995) regarding the importance of the front-end of projects, because it demonstrates not only why, but also how, IT projects can fail at the front-end.

The extensive literature on failure factors has been critiqued for being too abstract (Sauer, 1999). For systematic postmortem analysis, the failure factor literature (Hughes et al., 2016b; Schmidt, 2023b) is a valuable foundation when combined with process tracing and causal mechanism for identifying root causes of failure and eliminating rival explanations.

5.3. Recommendations for future research

Further research recommendations include additional postmortem analyses of failed IT projects. A corpus of IT project failure postmortems will constitute additional knowledge of the causes of IT project failure and a strong foundation for recommending additional practice improvements. A corpus of systematic IT project postmortems will also be a relevant curriculum for competence development.

Additionally, we recommend postmortem analyses of successful IT projects to develop knowledge of how successful projects have mitigated risks of failure. This will require additional conceptual work, since defining IT project success is not as simple as defining that a terminated IT project is a failure.

6. Acknowledgements

We are grateful to Bent Flyvbjerg, Peter Sestoft, and to the Editor-in-Chief, Gaurav Bansal, and the anonymous reviewers of *Journal of Information Technology Case and Application Research* for their comments and suggested improvements to previous versions of this paper.

7. Full disclosure

The author was from 2010 to 2013 employed in a senior executive role by the main contractor for the project included in the study, and therefore directly involved in the project. The research behind this paper was originally funded partially by the Danish Ministry of Finance. The Ministry has provided no input for the research, nor imposed any restrictions. These facts are stated as a formality, and they do not constitute any conflict of interest.

References

Ackermann, F. and Alexander, J. (2016). Researching complex projects: Using causal mapping to take a systems perspective. *International Journal of Project Management*, 34(6):891–901.

Ahonen, J. J. and Savolainen, P. (2010). Software engineering projects may fail before they are started: Post-mortem analysis of five cancelled projects. *Journal of Systems and Software*, 83(11):2175–2187.

Atkins, W. (2001). Root causes analysis: Literature review. Contract Research Report 300/2001, Health and Safety Executive, UK (WS Atkins Consultants).

Ayat, M., Imran, M., Ullah, A., and Kang, C. W. (2021). Current trends analysis and prioritization of success factors: a systematic literature review of ict projects. *International journal of managing projects in business*, 14(3):652–679.

Baker, B. N., Murphy, D. C., and Fisher, D. (1983). *Factors affecting project success*. Van Nostrand Reinhold.

BCG (2009). Eksternt review af polsag-programmet. Review, Rigpolitiet.

BCG (2011). Review af polsag. Technical report, Udvalget vedrørende en undersøgelse af POLSAG.

Beach, D. (2019). *Process tracing methods: Foundations and Guidelines*. Springer.

Beach, D. and Pedersen, R. B. (2013). *Process-Tracing Methods: Foundations and Guidelines*. The University of Michigan Press.

Bennett, A. (2008). Process tracing: A bayesean perspective. In Box-Steffensmeier, J. M., Brady, H. E., and Collier, D., editors, *The Oxford Handbook of Political Methodology*, chapter 30, pages 702–721. Oxford University Press.

Bennett, A. (2010). Process tracing and causal inference. In Brady, H. E. and Collier, D., editors, *Rethinking Social Inquiry: Diverse Tools, Shared Standards*, pages 2017–19. Rowman & Littlefield.

Bennett, A. and Checkel, J. T., editors (2014). *Process Tracing: From Metaphor to Analytic Tool*. Cambridge University Press, november edition.

Birk, A., Dingsøyr, T., and Stålhane, T. (2002). Postmortem: Never leave a project without it. *IEEE Software*, 19(3):43–45.

Boddie, J. (1987). The project postmortem. *Computerworld*, 21(49):77–81.

Brady, H. E. (2011). Causation and explanation in social science. In Goodin, R. E., editor, *Oxford Handbook of Political Science*, pages 1054–1107. Oxford University Press.

Brady, H. E. and Collier, D. (2010). *Rethinking Social Inquiry: Diverse Tools, Shared Standards*. Rowman & Littlefield, 2nd edition.

Brown, T. (2001). Modernisation or failure? IT development projects in the UK public sector. *Financial Accountability & Management*, 17(4):363–381.

Bryman, A. (2016). *Social Research Methods*. Oxford University Press, 5th edition.

Byrne, D. and Uprichard, E. (2012). Useful complex causality. In Kincaid, H., editor, *The Oxford Handbook of Philosophy of Social Science*, chapter 6, pages 109–129. Oxford University Press.

Cerpa, N. and Verner, J. M. (2009). Why did your project fail? *Communications of the ACM*, 52(12).

Charette, R. N. (2005). Why software fails. *IEEE Spectrum*, 42(9):42–49.

Chua, A. (2009). Exhuming it projects from their graves: An analysis of eight failure cases and their risk factors. *Journal of Computer Information Systems*, 49(3):31–39.

Cole, A. (1995). Runaway projects - cause and effects. *Software World (UK)*, 26(3).

Collier, B., DeMarco, T., and Fearey, P. (1996). A defined process for project post mortem review. *IEEE Software*, 13(4):65–72.

Collier, D. (2011). Understanding process tracing. *Political Science and Politics*, 44(4):823–30.

Collier, D., Brady, H. E., and Seawright, J. (2010). Outdated views of qualitative methods: Time to move on. *Political Analysis*, 18(4):506–513.

Contractors (2006a). Contract, appendix d, prices v.3.10. Technical report, CSC.

Contractors (2006b). Indicative proposal for polsag 1:1 plus additions. Document 153415.

Contractors (2007). Original supply contract. Technical report, Rigspolitiet.

Contractors, C. (2006c). Price estimate, polsag. Letter from the CIO of the police to the main contractor.

Cropper, S., Ebers, M., Huxham, C., and Ring, P. S., editors (2008). *The Oxford Handbook of Inter-Organizational Relations*. Open University Press.

Dekker, S. (2014). *The Field Guide to Understanding 'Human Error'*. Ashgate Publishing Limited, 3rd edition.

Deloitte (2002). It strategi 2002. Technical report, Rigspolitiet.

Dingsøyr, T. (2005). Postmortem reviews: purpose and approaches in software engineering. *Information and Software Technology*, 47:293–303.

Dwivedi, Y., Wastell, D., Henriksen, H., and De', R. (2015a). Guest editorial: Grand successes and failures in it: Private and public sectors. *Information Systems Frontiers*, 17(1):11–14.

Dwivedi, Y., Wastell, D., Laumer, S., Henriksen, H., Myers, M., Bunker, D., Elbanna, A., Ravishankar, M., and Srivastava, S. (2015b). Research on information systems failures and successes: Status update and future directions. *Information Systems Frontiers*, 17(1):143–157.

Dwivedi, Y. K. (2013). *Grand successes and failures in IT, public and private sectors, IFIP WG 8.6 International Working Conference on Transfer and Diffusion of IT, TDIT 2013, Bangalore, India, June 27-29, 2013. Proceedings.* Springer, Heidelberg.

Eden, C. and Ackermann, F. (1992). The analysis of cause maps. *Journal of Management Studies*, 29(3):309–324.

Eisenhardt, K. M. (1989). Building theories from case study research. *The Academy of Managemnet Review*, 14(4):532–550.

El-Emam, K. and Koru, A. G. (2008). A replicated survey of it software project failures. *Software, IEEE*, 25(5).

Ewusi-Mensah, K. (2003). *Software development failures. Anatomy of Abandoned Projects.* The MIT press.

Flyvbjerg, B. (2006). Five misunderstandings about case-study research. *Qualitative Enquiry*, 12(2):219–245.

Flyvbjerg, B. (2007). Curbing optimism bias and strategic misrepresentation in planning: Reference class forecasting in practice. *European Planning Studies*, 16(1):3–21.

Flyvbjerg, B. (2009a). Optimism and misrepresentation in early project development. In Williams, T. M., Sunnevag, K. J., and Samset, K., editors, *Making essential choices with scant information. Front-end decision making in major projects*, pages 147–169. Palgrave Macmillan.

Flyvbjerg, B. (2009b). Survival of the unfittest: why the worst infrastructure gets built—and what we can do about it. *Oxford Review of Economic Policy*, 3:344–367.

Flyvbjerg, B. (2021). Top ten behavioral biases in project management: An overview. *Project Management Journal*, 52(6):531–546.

Flyvbjerg, B., Budzier, A., Lee, J. S., Keil, M., Lunn, D., and Bester, D. W. (2022). The empirical reality of it project cost overruns: Discovering a power-law distribution. *Journal of Management Information Systems*, 39(3):607–639.

Flyvbjerg, B. and Gardner, D. (2023). *How big things get done.* Random House.

Fowler, J. and Horan, P. (2007). Are information systems' success and failure factors related?: An exploratory study. *Journal of Organizational and End User Computing*, 19(2):1–22.

Gartner (2004). Analyse af politiets it-strategi. Technical report, Finansministeriet.

Gartner (2007). Evaluering af budget for nyt sagsstyringssystem, document 139862. Technical report, Rigspolitiet.

Gawande, A. (2011). *The Checklist Manifesto: How to Get Things Right*. Profile.

George, A. L. and Bennett, A. (2005). *Case Studies and Theory Development in the Social Sciences*. MIT Press.

Glass, R. L. (1998). *Software Runaways: Lessons Learned from Massive Software Project Failures*. Prentice Hall, 1 edition.

Glass, R. L. (2001). Frequently forgotten fundamental facts about software engineering. *IEEE Software*, 18(3):112–111.

Glass, R. L. (2002). Project retrospectives, and why they never happen. *IEEE Software*, 19(5):112–111.

Goodin, R. E., editor (2011). *The Oxford Handbook of Political Science*. Oxford University Press.

Hughes, D. L., Dwivedi, Y. K., Rana, N. P., and Simintiras, A. C. (2016a). Information systems project failure – analysis of causal links using interpretive structural modelling. *Production Planning & Control*, 27(16):1313–1333.

Hughes, D. L., Dwivendi, Y. K., Simintiras, A. C., and Rana, N. P. (2016b). *Success and Failure of IS/IT Projects - A State of the Art Analysis and Future Directions*. Springer.

Hughes, D. L., Rana, N. P., and Simintiras, A. C. (2017). The changing landscape of is project failure: an examination of the key factors. *Journal of Enterprise Information Management*, 30(1):142–165.

Jones, C. (1995). Patterns of large software systems: failure and success. *IEEE Computer*, 28(3):86–87.

Jones, C. (1996). *Software Systems Failure and Success*. Thomson Computer Press.

Jones, C. (2010). *Software Engineering Best Practices. Lessons from Successful Projects in the Top Companies*. McGraw-Hill.

Jones, C. (2017). *Software Methodologies: A Quantitative Guide*. Auerbach Publications.

Kahneman, D. and Tversky, A. (1977). Intuitive prediction: Biases and corrective procedures. Technical report, Defence Advanced Research Projects Agency.

Kappelman, L. A., McKeenman, R., and Zhang, L. (2006). Early warning signs of it project failure: The dominant dozen. *Information Systems Management*, 23(4):31–36.

Kasi, V., Keil, M., Mathiassen, L., and Pedersen, K. (2008). The post mortem paradox: a delphi study of it specialist perceptions. *European Journal of Information Systems*, 17(1):62–78.

Keider, S. P. (1984). Why systems development projects fail. *Journal of Information Systems Management*, 1(3):33–38.

Keil, M., Cule, P., Lyytinen, K., and Schmidt, R. C. (1998). A framework for identifying software project risks. *Communications of the ACM*, 41(11):76–83.

Kerth, H. (2001). *Project Retrospectives: A handbook for Team Reviews*. Dorset House Publishing.

Kerzner, H. R. (2014). *Project Management Best Practices Achieving Global Excellence*. Wiley, 3rh edition.

Kincaid, H., editor (2012). *The Oxford Handbook of Philosophy of Social Science*. Oxford University Press.

Klein, H. K. and Myers, M. D. (1999). A set of principles for conducting and evaluating interpretive field studies in information systems. *MIS Quarterly*, 23:67.

Lawler, J. and Waldner, D. (2023). Interpretivism versus positivism in an age of causal inference. In Kincaid, H. and Bouwel, J. V., editors, *Oxford Handbook of Philosophy of Political Science*, chapter 11, pages 221–242. Oxford University Press.

Maddison, A. (2011). *The impact of critical success factors on government IT projects: A case study of the defence information infrastructure programme*. Phd thesis, Cranfield University, College Rd, Cranfield MK43 0AL, UK.

Mahoney, J. (2012). The logic of process tracing tests in the social sciences. *Sociological Methods & Research*, 41(4):570–597.

Maurer, I. (2010). How to build trust in inter-organizational projects: The impact of project staffing and project rewards on the formation of trust, knowledge acquisition and product innovation. *International Journal of Project Management*, 28:629–637.

Maxwell, J. A. (2002). Realism and the role of the researcher in qualitative psychology. In Kiegelmann, M., editor, *The role of the researcher in qualitative psychology*, pages 11–30. Ingeborg Huber.

Maxwell, J. A. (2004). Using qualitative methods for causal explanation. *Field Methods*, 16(3):243–264.

McKinsey (2010). Budgetanalyse af politiet 2009-2010. Technical report, Styregruppen for budgetanalyse af politiet.

McManus, J. and Wood-Harper, T. (2008). A study in project failure. Technical report, BSC, The Chartered Institute for IT.

Meier, S. R. (2008). Best project management and systems engineering practices in pre-acquisition practices in the federal intelligence and defence agencies. *Project Management Journal*, 39(1):59–71.

Morris, P. (2011). Managing the front-end: Back to the beginning. *Project Perspectives*, 33:4–9.

Morris, P. (2013). *Reconstructing Project Management*. Wiley-Blackwell.

Morris, P. and Hough, G. (1987). *The Anatomy of Major Projects*. Chichester: Wiley and Sons.

Myllyaho, M., Salo, O., Kääriäinen, J., Hyysalo, J., and juha Koskela (2004). A review of small and large post-mortem analysis methods. Technical report, ICSSEA.

National Audit Office (2013a). Beretning til statsrevisorerne om politiets it-system polsag. Technical report, Rigsrevisionen (National Audit Office).

National Audit Office (2013b). Beretning til statsrevisorerne om politiets it-system polsag. Technical report, Rigsrevisionen (National Audit Office).

Nelson, R. R. (2005). Project retrospectives: Evaluating project success, failure, and everything in between. *MIS Quarterly Executive*, 4(3):361–372.

Nelson, R. R. (2021). *IT Project Management: Lessons Learned from Project Retrospectives 1999–2020*, volume 4. Now Publishers, Inc.

NTSB (2016). Mission statement of the national transportation safety board. Technical report, National Transportation Safety Board.

Pan, G. and Flynn, D. (2003). Why information systems project postmortems fail: An attribution perspective based on a case study analysis. In *ICIS Proceedings*, volume Twenty-Fourth International Conference on Information Systems of *AIS Electronic Library*, pages 366–377. Association for Information Systems.

Paradies, M. and Busch, D. (1988). Root cause analysis at savannah river plant. *IEEE Conference on Human Factors and Power Plants*, pages 479–483.

Parliament (2006). Aftale om gennmførelse af politireformen og styrkelse af politiindsatsen. Technical report, The Danish Parliament.

Paté-Cornell, E. (1993). Learning from the piper alpha accident: A postmortem analysis of technical and organizational factors. *Risk Analysis*, 13(2):215–232.

Pearl, J. (2009). *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2nd edition.

Pearl, J. (2013). Structural counterfactuals: A brief introduction. *Cognitive Science*, 37:977–985.

Pearl, J. and Mackenzie, D. (2018). *The Book of Why*. Allen Lane, Penguin Books.

Petroski, H. (1992). History and failure. *The American Scientist*, 80(6):523–526.

Pinto, J. K. and Mantel, S. J. (1990). The causes of project failure. *IEEE Transactions on Engineering Management*, 37(4):269–276.

Pinto, J. K. and Slevin, D. P. (1987). Critical factors in successful project implementation. *IEEE Transactions on Engineering Management*, EM-34(1).

PMI (2021a). *A Guide to the Project Management Body of Knowledge*. Project Management Institute, 7th edition.

PMI (2021b). *Software Extension to the PMBOK Guide Seventh Edition*. Project Management Institute.

Rambøll (2005). Nyt polsas, document, 139707, 153443. Technical report, Rigspolitiet.

Reason, J. T. (1990). *Human Error*. Cambridge University Press.

Reel, J. S. (1999). Critical success factors in software projects. *IEEE Software*, 16(3):18–23.

Rigspolitiet (2002). Polsas review, estimering ifm. udvikling af nyt sagsbehandlingssystem. Technical report, Rigpolitiet.

Rigspolitiet (2006). Project board meeting 8 december 2006. Technical report, Rigpolitiet.

Sarker, S. (2021). Ten statements related to qualitative research in is: true or false? *Journal of Information Technology Case and Application Research*, 23(4):251–256.

Sauer, C. (1999). Deciding the future for is failures not the choice you might think. In Currie, W. and Galliers, B., editors, *Rethinking Management Information Systems*, pages 279–309. Oxford University Press.

Schmidt, J. (2020). The polsag project: Root cause analysis. Technical report, IT University of Copenhagen.

Schmidt, J. (2022). It project failure, termination, and the marginal cost trap. *Journal of Modern Project Management*, 10(2):255–275.

Schmidt, J. (2023a). It project postmortems: Theoretical foundations. *Unpublished*.

Schmidt, J. (2023b). Mitigating risk of failure in information technology projects: Causes and mechanisms. *Project Leadership and Society*, 4:100097.

Schmidt, R., Lyytinen, K., Keil, M., and Cule, P. (2001). Identifying software project risks: An international delphi study. *Journal of Management Information Systems*, 17(4):5–36.

Scott, J. (2014). *A matter of record: Documentary sources in social research.* John Wiley & Sons.

Scriven, M. (1974). Maximizing the power of causal investigations: The modus operandi method. In Popham, W. J., editor, *Evaluation in Education: Current Applications*, pages 68–84. Mccutchan Publication Corporation.

Sommerville, I. (2004). *Software Engineering.* Addison Wesley, Boston, MA., 7th edition.

Sommerville, I. (2016). *Software Engineering.* Addison Wesley, 10th edition.

Standish (2014). The standish group report. Technical report, The Standish Group.

Standish (2022). The chaos report (2022). Technical report, The Standish Group.

Suprapto, M., Bakker, H. L., Mooi, H. G., and Hertogh, M. J. (2016). How do contract types and incentives matter to project performance? *International Journal of Project Management*, 34(6):1071 – 1087.

Sutton, R. I. and Staw, B. M. (1995). What theory is not. *Administrative Science Quarterly*, 40(3):371–384.

The Danish Ministry of Finance and PA Consulting Group (1995). Turnusundersøgelse af politiet: Politiets administrative arbejdstirettelæggelse og teknologianvendelse. Technical report, The Danish Ministry of Finance.

Tiedeman, M. (1990). Post-mortems - methodology and experiences. *IEEE Journal on Selected Areas in Communications*, 8(2):176–180.

Trampusch, C. and Palier, B. (2016). Between x and y: how process tracing contributes to opening the black box of causality. *New Political Economy*, 21(5):437–454.

United Nations (2022). United nations e-government development index. Technical report, United Nations.

Van Evera, S. (1997). *Guide to Methods for Students of Political Science.* Cornell University Press, reprint edition.

Verner, J., Cox, K., Bleistein, S., and Cerpa, N. (2005). Requirements engineering and software project success: An industrial survey in australia and the u.s. *Australasian Journal of Information Systems*, 13(1):225–238.

Verner, J., Sampson, J., and Cerpa, C. (2008). What factors lead to software project failure? In *Proceedings of the Second International Conference. Research Challenges in Information Science.*, pages 71–80. RCIS, IEEE.

Waldner, D. (2012). Process tracing and causal mechanisms. In Kincaid, H., editor, *The Oxford Handbook of Philosophy of Social Science*, pages 1–15. Oxford University Press.

Waldner, D. (2015). Process tracing and qualitative causal inference. *Security Studies*, 24(2):239–250.

Waldner, D. (2016). Invariant causal mechanisms. *Qualitative & Multi-Method Research*, 14(1-2):28–33.

Waldner, D. (2019). Causal mechanisms and qualitative causal inference in the social sciences. In Nagatsu, M. and Ruzzene, A., editors, *Contemporary Philosophy and Social Science: An Interdisciplinary Dialogue*, chapter 9, pages 275–300. Bloomsbuft Academic.

Waldner, D. (2022). Qualitative causal inference and critical junctures: The problem of back-door paths. In Collier, D. and Munck, G. L., editors, *Critical Junctures and Historical Legacies: Insights and Methods for Comparative Social Science*, pages 159–182. Rowman & Littlefield Publishers.

Weick, K. E. (1995). What theory is not, theorizing is. *Administrative Science Quarterly*, 140(3):385–390.

Williams, T. (2004). Identifying the hard lessons from projects - easily. *International Journal of Project Management*, 22(4):273–279.

Williams, T., Ackermann, F., Eden, C., and Howick, S. (2001). The use of project post-mortems. In *Project Management Institute, Annual Symposium*, Annual Symposium, pages 1–6. Project Management Institute.

Yardley, D. (2002). *Successful IT Project Delivery: Learning the Lessons of Project Failure*. Addison Wesley, 1 edition.

Yeo, K. T. (2002). Critical failure factors in information system projects. *International Journal of Project Management*, 20:241–246.

Yin, R. K. (2009). *Case Study Research Design and Methods*, volume 5 of *Applied Social Research Methods Series*. SAGE Publications Ltd.

Yin, R. K. (2018). *Case study research and applications: Design and methods*. Sage publications Thousand Oaks, CA, 6th edition edition.

Appendix A. Project Planning

This appendix adds more detail to the analysis of the hypothesis that inadequate planning was a contributing root cause of the POLSAG project failure (see Section 3.4.3). Cost overrun, delays and shortfalls of scope and benefits are not uncommon in completed IT projects. Concluding that inadequate planning was a root cause of the POLSAG project termination therefore requires more than observing overruns, delays, and shortfalls in the project execution phase. The analysis in this appendix and in Section 3.4.3 therefore additionally demonstrates that important overruns, delays and shortfalls were foreseeable in the planning phase.

First, we present an overview of the development in the government funding for the project. This shows the willingness of the sponsor to fund significant cost overruns.

Second, we present the items that were necessary for the project, but not included in the budget. This shows that the significant initial and recurring costs were foreseeable in the planning phase, which supports the conclusion that inadequate planning was a contributing root cause of failure.

Third, we present the main schedule delays in the project. Fourth, we present the revisions of cost estimates.

Fifth, we present two business case simulations that show, albeit in hindsight, that the project was unattractive at the time of the contract signature. Simulation 1 is based on contract data and minimal scope extensions estimated by consultants. Simulation 2 is based on reference class forecasting (Flyvbjerg, 2007) using Jones' (1995) database of 6.700 IT projects from different sectors.

A.1. Funding

From 2007 the funding the POLSAG project more than doubled from 2004 to 2010 (National Audit Office, 2013b; Parliament, 2006): see Table A1. Cost values are shown in local currency to make numbers reconcilable with official reports.

Table A1. POLSAG overview of bills presented to Parliament.

Bill	**DKKm**	**Planned Completion**	**Scope**
Akt 57 2004	153 (173 in 2010 prices)		Initiation of development and modernisation program for the IT systems of the Danish Police (later replaced by Akt 74).
Akt 74 2005	153 (173 in 2010 prices)	As soon after 2006 as possible.	Procurement of system, recruiting staff with project competences, external consulting, operations environment (hardware and software), implementation etc. *Maintenance not considered.*
Akt 106 2006	-	-	Confirms 2005 bill.
Akt 119 2007	221 (236 in 2010 prices)	Mid 2010	Additional funding needed for additional cost of procurement and software development of DKK68m. Scope no longer includes operations environment (hardware and software). *Maintenance not considered.*
Akt 66 2010	405.5	June 2012	Additional DKK164.5m needed for additional software development and operations environments (hardware and software for testing, training, and operations). The bill includes of operations of these environments until June 2012. *Maintenance not considered.*
G 1 Feb 2012	313		Confidential bill, mandate to terminate the POLSAG contract and terminate the project. Additional grant: DKK350m less DKK37m cancelled grant, total DKK313m.

A.2. Scope

In the original POLSAG contract, necessary items were missing from the project scope[2]. Missing items included:

(1) Hardware and systems software to run the POLSAG system, including testing, staging, training, and operations of DKK75m (BCG, 2011, p25).

(2) Contingency allocations for necessary additional application software because of underspecification and underestimation. Underspecification and underestimation can be mitigated by considering past experience, or with the help reference classes (Flyvbjerg, 2009a,b), for example Jones (1995; 1996). The need for risk contingency allocations is generally acknowledged, and in the POLSAG project, contingency allocations could have mitigated the impact of DKK17m worth of delays and design changes and DKK63m worth of extensions. (BCG, 2011, p25).

(3) Allocations for foreseeable recurring updates of the application software to accommodate changes of laws and instructions (included in the DKK63m above).

(4) Allocations for updates in the systems predecessor, POLSAS, after the point in time, where the 1:1 reengineering of the POLSAS functionality in POLSAG had been completed.

(5) Allocations for external consultants needed to run the project.

(6) Costs for system software updates: database, word processing, scanning, pdf-file software, etc.

(7) Network equipment upgrades and increased network capacity.

(8) Maintenance cost (including license upgrades, error correction services and telephone support) of foreseeable application software extensions: according to the contract annually 17 - 26% of the cost of the extensions. This was particularly problematic, because maintenance costs are recurring annual costs,

[2]An overview of planned total cost of the project has not been found in the project documents. The main supply contract for POLSAG does not contain such and overview, for a number of reasons including that some of the project cost, for example cost of an operations environment, external consultants, integration layer and required modifications in adjacent system, were not part of the main POLSAG supply contract. Additionally, what can be characterised as "normal extensions" to a complex software system, like POLSAG, were not included in the supply contract, since they could be specified and quantified at the time contract preparation. Nevertheless, experience shows that such changes may be significant over the life time large and complex software system. Jones has suggested a 8% per year level of scope changes during the operations phase of a software system (Jones, 2017).

(9) Hardware for document scanning.

The lack of control over the scope of deliverables contributed to the lack of control over cost and schedule. It is also an indicator of inadequate planning.

A.3. Schedule

The contract signed in April 2007 stated that the POLSAG system would be fully implemented in the organisation by 1 December 2010, which was the planned date for approved operations test. This means that the guidance from top management to have the system fully implemented in 2009 had failed already at the time of signature of the contract.

In October 2009, a re-planning of the project was agreed, according to which the system would be fully implemented by end of December 2011. The duration of the project thus changed from approximately 44 months to approximately 55 months (an extension of 25%). This delay seems not to have been critical to the client. The delay was not listed as an issue in the six reasons for closing the project. *It is remarkable that a factor five increase in size of the software development project (from 27-28,000 to 140,000 hours for software development) was expected to be absorbed in only a 25% delay. If analysed at the time, this could have been an early indicator of problems ahead.*

The reviews and reassessment of the project seems to have been driven more by the cost increase, and the dissatisfaction of users at the pilot operation site than by the schedule delays. It seems that the police would have waited for the system, if they had had confidence in the quality and sustainability of the solution.

A.4. Cost

The cost estimates of the project varied substantially over time. Important components were left out of some of the cost estimates. Most importantly the cost of maintenance and recurring extensions of the software application, POLSAG. See Tables A1 (National Audit Office, 2013b) and A2.

According to the contract, the POLSAG software annual maintenance cost (i.e., excluding cost of the hardware and software of the technical operations environment) was calculated as 26% of software licenses plus extensions, including future extensions.

Table A2. POLSAG projected cost over time. FTE: Full Time Equivalent, yearly.

Date	**Cost mDKK**	**Annual mDKK**	**Internal FTE/DKKm**	**Comment**
2002, Feb. (Rigspolitiet, 2002)	125-175		350-375 FTE	DKK168m for bespoke system, DKK123-173m for COTS system. Both excluding integration layer.
2002, March (Deloitte, 2002)	140-190		375 FTE	Hardware, software, services and consulting, 2002-2008. Roll-out completed mid 2007 (consultant).
2004, June (Gartner, 2004)	95			Using COTS software and a "need to have" approach. Excluding integration layer, estimated at DKK20-25m (consultant).
2005, Sep. (Rambøll, 2005)	52	3		licenses, modifications and development. Does not include integration layer (consultant).
2006, Aug. (Contractors, 2006c)	50			Early estimate by the later to be contractor
2006, Nov. (Contractors, 2006b)	98			Proposal from contractor excluding integration layer.
2006, Dec. (Contractors, 2006a)	88	14		Price appendix to contract (signed March 2007). Excluding integration layer, excluding hw and sw for testing, training, staging and production.
2007, Feb. (Gartner, 2007)	189			"Evaluation of budget" for the new system, including integration layer and some follow-on cost. (The evaluation was made before contract signature, which took place in March 2007.)
2009, August (BCG, 2009)	248	35	70 FTE	FTE includes PM and development, not training (consultant).
2011, August (BCG, 2011)	390	166*	338m*	*) 2004-2012 Business Case (consultant)
2011, August (BCG, 2011)	484*	563*	390m*	*) 2004-2020 Business Case (consultant)
2013, Mar (National Audit Office, 2013b)	422**		145m**	**) 2005-2012 (National Auditor's review).

A.5. Business case simulation of supply contract

To assess a possible result of more extensive cost and benefit planning of the project, two simple planning simulations have been made. The simulations are based on the following assumptions:

(1) Simulation 1 (execution of contract with no changes):
   a) the application software (POLSAG) would - counterfactually - have been delivered at the cost of the original contract,
   b) the maintenance fees follow the original contract: annually from 17% to 26% of license cost and customisation cost, and
   c) the need for recurring extensions follow the forecast in a consultants report from late in the process (BCG, 2011).

(2) Simulation 2 (execution of contract with foreseeable changes):
   d) The same assumptions as in simulation 1 with the following additions based on reference class (Flyvbjerg, 2007; Jones, 1995, 2010):
      (i) initial requirements specifications assumed to be 80% complete, and
      (ii) software requirements assumed to change monthly by 1% during the design and development process. This is at the low end of Jones' indication of a 1% - 3% range.

The simulations show that over a ten year period the cost of the system would be DKK720.5m in simulation 1 (Table A4), and DKK933.35m in simulation 2 (Table A5), including minimal extensions beyond legal requirements, but excluding training, external consultants, internal resources etc. The benefits from the same consultants report (BCG, 2011, p92, p94) for that period would be in the range from DKK603.9m ("low benefits") to DKK942.7m ("high benefits"). The simulations are based on simple extrapolation, since the consultants' report assesses benefits over a nine-year period. This means that the benefit range would be from DKK-116.6m (negative) to DKK222.2m in simulation 1 (Table A4), and in the range of DKK-329.45m (negative) to DKK9.35m in simulation 2 (Table A5). The positive effects from the system mainly stem from improved efficiencies in the physical handling of documents (BCG, 2011). It is very

likely that a less costly solution than POLSAG (DKK80m per year, excluding internal resources, consultants etc.) could have introduced similar efficiency improvements.

For comparison, the IT spending in 2008 by the central IT department of the National Police (i.e., excluding DKK55m IT spending in the constabularies) was DKK272m (McKinsey, 2010), including mainframe operations and all legacy systems. The new POLSAG system would - by a conservative estimate - have accounted for 23% of the IT department's spending for the first ten years of operation. There is no record of this number being discussed prior to signing the contract.

This means that, based on the benefits calculations by consultants in 2011 (BCG, 2011), the POLSAG business case was probably unattractive even at the time the contract was signed. This finding is significant. A possible reason why this problem was not considered earlier in the process is the incompleteness of the original budget, in other words: inadequate planning.

The original budget did not include significant necessary maintenance cost and recurring extensions of the software driven by, for example, changes in legislation, and the maintenance fees for such extensions[3].

A.6. Business case simulation calculations and assumptions

This section contains the sources and assumptions behind the counterfactual reconstruction of a partial project business case that could - in principle - have been made at the time of the proposal for the POLSAG project, i.e., before the project was committed and the contract signed.

Simulation 1 (Table A4) calculates the situation where the project had been delivered on time and within the cost specified in the contract, i.e., with no change of budget, schedule or scope.

Simulation 2 (Table A5) calculates a situation where the project would have been delivered on time and within the cost specified in the contract, but with estimated

[3]A consultants' report from late in the process (BCG, 2011) makes an explicit assumption that the maintenance fees would be renegotiated. It is not clear how realistic that assumption is, given that both contractor and subcontractor had made significant losses on the project. It is realistic to assume that the contractor and the subcontractor would want to recover some of these losses over time from profits on the recurring annual maintenance fees.

changes to scope and cost based on Jones (1995; 2010) used as reference class (Flyvbjerg, 2007). See additional assumptions and sources in Table A3).

(1) The software contract sum is from the contract signed in 2007.
(2) Extensions for legally required changes are as estimated by consultants (BCG, 2011).
(3) Cost of operations and platform maintenance, including integration layer is at consultants estimates (BCG, 2011).
(4) Hardware and software cost are as estimated by consultants (BCG, 2011).
(5) The consultant's estimate of reduced software maintenance fees are excluded (BCG, 2011), since the assumptions behind the reduction are not documented, they were not discussed with the contractors, and their plausibility is not clearly justified.

Cost excluded from the rough business case simulation include:

(1) Internal cost,
(2) cost of user training, according to a consultant, DKK118m (BCG, 2011),
(3) cost of external consultants,
(4) cost of additional hardware and software, e.g. scanning, PC upgrades etc.,
(5) cost of communications infrastructure upgrades, e.g. WAN connections, and
(6) contingency allocations.

A.6.1. Business case simulation 1: contract

This calculation (see Table A4) simulates the counterfactual situation, where the POLSAG software would have been delivered at the cost of the original contract signed in 2007. Legally required extensions estimated by a consultant in 2011 (BCG, 2011) cover necessary modifications of the software to accommodate changes of rules and legislation affecting the case management of the police. The DKK14m cover maintenance fees from the original contract signed in 2007, the 17% is the annual fee for application software extensions (the standard fee is 26%, but extensions have been calculated in the original contract at 17%). The DKK76m for testing and operations environment

Table A3. POLSAG business case simulation assumptions and sources.

#	**Business case simulation assumptions and sources**	**DKKt**	%
1	POLSAG application sw (Contract, (Contractors, 2007)))	88.000	
2	Standard underspecification (20%, i.e. +25%) (Jones, 1995)	22.000	
3	Standard ration of change (1% pr m x approximately 24 mths) (Jones, 1995)	21.120	
4	Additional scope (using Jones 1995 as reference class)	43.120	49 %
5	Extensions, (BCG, 2011, p43), estimated changes due to changes of legislation	10.000	
6	Other extensions and changes, e.g., adaptation for new versions of office automation software, additional functions etc.(rough estimate)	5.000	
7	Annual ops and platform maintenance, (BCG, 2011, p100), "drift af produktionsmiljøer + vedl. af randsystemer og integrationslag (14-5)"	19.000	
8	Hw & sys sw (BCG, 2011, p18,p25) Test env. plus	76.000	
9	Benefits range (high) (BCG, 2011, p12)	857.000	
10	Correction for one additional year (benefits, high)	942.700	
11	Benefits range (low) (BCG, 2011, p12)	549.000	
12	Correction for one additional year (benefits, low)	603.900	
13	Excluded cost: Internal resources, external consultants, user hardware and software, upgrading of communications infrastructure, scanning hardware, risk contingency allocations, user training (DKK118m, consultants' estimate).		

and the DKK19m annual cost of these environments are from the consultant's review in 2011 (BCG, 2011), and the range of quantifiable benefits is from the same report.

A simple ten year projection shows that even disregarding important missing elements (such as internal cost, external consultants, other than legally required extensions of the software, scanning equipment, user training, etc.) the contract entailed a commitment of more than DKK700m. It should be noted that the quantifiable benefits almost exclusively originated in supposed savings from electronic handling of previously physical documents in connection with delivering documents to the courts, and from filing closed cases. Based on this simulation, the project is not attractive from an investment point of view. Even when significant costs, for example the training of 10.000 user, are excluded, the low range of the benefit forecast renders the investment negative. The midtpoint of the net effect based on the consultants' forecasted range of benefits (low to high) is DKK52.8m, but it is plausible that the forecasted benefits could be achieved more cost effectively (Schmidt, 2022), since they are generated with a very limited subset of the functionality of the POLSAG system.

Table A4. Business case simulation 1: minimal scope extensions.

DKKt / Year	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Totals**	**Totals**
Contract, application sw	88.000												
Additional scope	0												
Total application sw	88.000											88.000	
Legal extensions*		10.000	10.000	10.000	10.000	10.000	10.000	10.000	10.000	10.000	10.000		
Other extensions		0	0	0	0	0	0	0	0	0	0	0	
Applications sw total extensions		10.000	10.000	10.000	10.000	10.000	10.000	10.000	10.000	10.000	10.000	100.000	
Annual app sw maint. (17%)	14.000	15.700	17.400	19.100	20.800	22.500	24.200	25.900	27.600	29.300	31.000	247.500	
Hardware & sys sw (test & ops*)	76.000											76.000	
Annual ops & platform maint.	19.000	19.000	19.000	19.000	19.000	19.000	19.000	19.000	19.000	19.000	19.000	209.000	
Total hw & sw (incl. maint.)												**720.500**	**720.500**
Benefits range (BCG, 2011, p12)												603.900	942.700
Net effect (low - high)												**-116.600**	**222.200**

*Consultants' estimate

Table A5. Business case simulation 2: reference class extensions.

DKKt / Year	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Totals**	**Totals**
Contract, application sw	88.000												
Additional scope	43.000												
Total application sw	131.000											131.000	
Legal extensions*		10.000	10.000	10.000	10.000	10.000	10.000	10.000	10.000	10.000	10.000		
Other extensions		5.000	5.000	5.000	5.000	5.000	5.000	5.000	5.000	5.000	5.000		
Applications sw total extensions		10.000	10.000	10.000	10.000	10.000	10.000	10.000	10.000	10.000	10.000	150.000	
Annual app sw maint. (17%)	14.000	16.550	19.100	21.650	24.200	26.750	29.300	31.850	34.400	36.950	39.500	294.250	
Additional scope maint. (17%)		7.310	7.310	7.310	7.310	7.310	7.310	7.310	7.310	7.310	7.310	73.100	
Hardware & sys sw (test & ops*)	76.000											76.000	
Annual ops & platform maint.	19.000	19.000	19.000	19.000	19.000	19.000	19.000	19.000	19.000	19.000	19.000	209.000	
Total hw & sw (incl. maint.)												**933.350**	**933.350**
Benefits range (BCG, 2011, p12)												603.900	942.700
Net effect (low - high)												**-329.450**	**9.350**

*Consultants' estimate

A.6.2. Business case simulation 2: contract and reference class

Simulation 2 (see Table A5) differs from simulation 1 (see Section A.6.1) by including an experience-based level of variation in software projects relative to the initial requirement specification. The deviation is calculated based on a view that the original requirement specification is 80% complete, and that specifications change by monthly 1% during the design and development process (Jones, 1995, 2010). Additionally, a modest inclusion of cost of modifications other than the DKK10m per year estimated to cover changes required strictly by changes of rules and legislation affecting the case management of the police. The DKK3m annually is a rough estimate. These modifications to the business case simulation make the investment in the project seem even less compelling (see Table A5) than in simulation 1.

A.7. Conclusions on inadequate project planning as cause of failure

The purpose of this appendix is to add detail to the process tracing analysis in Section 3.4.3 that leads to the conclusion that incomplete, or inadequate, planning was a contributing cause of the POLSAG IT project failure. In summary, the main observations regarding planning of cost, scope, schedule and business cases are:

(1) Cost and scope
   (a) In 2005, the project cost was estimated at DKK153m, presumably everything included, except maintenance cost (see table A1).
   (b) In 2007, the project cost was estimated at DKK221m, an additional 36% but now explicitly *excluding* hardware and software to operate the system. Maintenance cost is still not considered.
   (c) In 2010, the project cost was estimated at DKK405.5m, an additional 72% to the 2015 estimate, now including hardware and software to operate and test the system.

(2) Schedule and delays
   (a) In 2006 the Chief of Police stressed the importance of the project being fully implemented by 2009 (Rigspolitiet, 2006) The Danish Police reform was implemented in 2007 consolidating 54 constabularies to 12.

(b) It was not until 2010 that the POLSAG software first failed the acceptance test. The functional acceptance test was later passed, and the system was taken into pilot operation with a very poor result. The pilot operation was later given up and the users subsequently migrated back to the old system.

(3) Business case

(a) One of the six reasons for closing the POLSAG project was an "Unsatisfactory business case" (National Audit Office, 2013a), so inadequate planning is a strong candidate for a contributing root cause of project failure.

(b) Business case simulations (A.6.1 and A.6.2) demonstrate that business case was unattractive even at the time of the contract signature.

(c) More comprehensive planning could have led to a redesign of the project and the consideration in the planning phase of more cost effective alternatives for achieving the desired benefits.

(d) The instability of estimates and budgets of the POLSAG project, driven mainly by lack of control over the scope of delivery, is an additional indicator that supports the conclusion that inadequate planning was a contributing root cause of project failure.