MIS Quarterly

RESEARCH ARTICLE

INTENSIFYING TO CEASE: UNPACKING THE PROCESS OF INFORMATION SYSTEMS DISCONTINUANCE

Mohammad Hosein Rezazade Mehrizi
School of Business and Economics, Vrije University Amsterdam, De Boelelaan 1105,
1081 HV Amsterdam, NETHERLANDS {m.rezazademehrizi@vu.nl}

Joan Rodon Modol
Universitat Ramon Llull, ESADE Business School,
08172 Sant Cugat del Vallès, SPAIN {joan.rodon@esade.edu}

Milad Zafar Nezhad
Industrial and Systems Engineering, Wayne State University,
Detroit, MI 48202 U.S.A. {m.zafarnezhad@wayne.edu}

Appendix A

Coding for the Discontinuance Phases and Their Interactions Across the Cases

Table A1. Realization

Realization	SmallCo	MedCo	LargStrict	LargLenient
Triggers • Acquaintance with new IS • Technological changes • New requests by leading clients	*(Started in early 1993)* The technical designers and programmers and the CEO actively observed the technological changes and saw the trend of MS-Windows becoming the next technology generation	(*Mainly became a hot discussion in 1995*) The acquaintance of a few fresh programmers with the MS-Windows The strategic decision to abandon many old products and only focus on office automation systems	(Mainly started in 1993) The limitations of MS-DOS relative to Windows for producing integrated set of organizational systems and growing demand from their large client base	*(Mainly started in 1993)* The recognition of the limitations of MS-DOS relative to Windows, and their strategy to focus on state of the art technologies
Actions and the roles of embedded actors **Scrutinizing old IS limitations (ceasing indwelling SRM)** • Technical teams and middle managers and later top management and sales and marketing made sense of the limitations of old systems relate to new ones • Technical development teams were searching for the comparable	*(For the entire year 1993 and before March 1994)* Examining the limitations of MS-DOS in terms of graphical functionalities, parallel processing programming; scrutinizing mainly happened to help the valuation of old product for selling them out	*(Frequently observed during 1995 and 1996; and later as an important set of actions until 1999)* It took two years of discussion about the tools, techniques, units, manuals, and expertise developed around MS-DOS to exactly see which features were problematic and which	*(Intensively for the years 1993-1994; and later as occasional actions during 1995-1999 when major limitations of MS-DOS systems was detected or when addressing a major change request was difficult due to basic limitations of MS-DOS)* Scrutinizing MS-DOS	*(From 1993-1994 as the intensive period of realization, yet still often surging in the discussions until 2002)* Extensively done by R&D team and the production team and required formal R&D projects due to the huge investment in developing various technical tools related to

advantages and capabilities of the old technologies and related products relative to the new ones • Senior experts highly specialized in the old systems were posing novel questions about old systems		ones were still viable and should be kept in the new office automation systems	aspects was extensively done with regard to the implications of moving toward more integrated systems as well as the routines and organizational procedures for designing, producing, and supporting isolated systems	MS-DOS and huge number of interconnected products with sophisticated DOS-related technologies
Discrediting the viability of (parts of) old IS (ceasing legitimization SRM) • Product managers in discussion with technical teams were distinguishing between viable and obsolete aspects of old systems and discussing the boundaries between them • Forward jumpers were discrediting old systems by highlighting the new technological opportunities and deficiencies of old systems • Middle managers had to constantly convince forward jumpers to not disregard the entire old systems as obsolete • Marketing managers and agents were carefully decoupling between internal and external legitimacy of the old systems	Most of the managers (except the financial manager) and technical team (except two seasoned programmers) already considered the old systems as obsolete	Extensively engaging a wide range of managers and technical experts with general tendency to consider many aspects as viable at the beginning, especially the aspects that had been improved through further development of the systems	Discrediting MS-DOS extensively happened at the level of basic product design, as well as the organizational procedures and routines for designing, producing, and supporting products	Initially (during the first half of 1993) it was only done for few aspects of MS-DOS (e.g., parallel processing); yet later it required more effort by the R&D manager due to the dominance of seasoned MS-DOS experts
Scrutinizing ➜ delegitimizing • Paying attention to the detailed, technical characteristics and examining various potential technological features that appeared to be limiting the capacities and functions of the old IS • Reopening the list of shortcomings of the technology that were worked around and adjusted locally in past upgrades and maintenance	*(Through free discussions at the end of the weeks during 1993)* A list of MS-DOS shortcomings was created in order to convince the financial office and their partners that MS-DOS had no future	Scrutinizing was limited to only those aspects that could not be easily improved based on their deep expertise in MS-DOS technology	To expand the list of problematic elements to design, testing, and production techniques and routines	Often happened as systematic tasks to reflect on the limitations of MS-DOS even though many of the shortcomings of the MS-DOS was improved by the technical team
Delegitimization ➜ scrutinizing • The discussion about the boundaries between old-viable and old-obsolete required inspecting some detailed technological features and aspects in more details (to see which aspect is really the problematic aspect)	CEO asked in middle 1993 the head of technical team to create a list of limitations of the product to distinguish between what was problematic about MS-DOS and what were the things that needed to be kept in the new product *(an ongoing task during year 1993)*	It required a lot of effort to articulate detailed reasons for the fundamental limitation of MS-DOS; Technical Manager always stressed: "*let's be careful not to through away the good apples*"	To unpack the range of interdependencies between data, codes, and designs that might not work properly in integrated design	It became relevant when some new features of MS-DOS were further developed, e.g., the possibility of working with graphical peripheral devices such as laser printers and high-resolution monitors
Contextual conditions shaping realization • The longer and deeper experience and specialization in old IS ➜ required more time and efforts to reopen the taken-for-granted details of old IS and carefully distinguish viable and obsolete aspects • The larger number of interconnected technological elements and products related to old IS (relevant in LargLenient and LargStrict) ➜ required more extensive discussions and examinations for distinguishing between old-viable and old-obsolete aspects of old IS • The dominance of forward jumpers with	Small number of clients (10 clients) Their long experience with MS-DOS and up-to-date technical managers and employees made them have the feeling "we are done with MS-DOS"! A simple, stand-alone product that facilitated its sale	Highly specialized skills and technological elements related to MS-DOS systems, with a few seasoned experts on MS-DOS challenged the realization, yet the small number of products facilitated the realization that the products should be sooner or later abandoned	The large customer base and still growing demand for extending the support contracts created tensions between the marketing and production departments regarding the obsolescence of MS-DOS based products	Large number of clients and high interconnectivity of MS-DOS based product, which made it difficult to easily realize that MS-DOS was becoming obsolete; yet the technological strategy of the firm to work on state-of-the art technology facilitated realization

the culture of appeal to new technology, in comparison with seasoned specialized experts on old IS ➜ supporting the de-legitimization of old IS, though increasing the risk of immaturely de-legitimizing the entire old IS path too early				

Table A2. Marginalization

Marginalization	SmallCo	MedCo	LargStrict	LargLenient
Triggers • The fear that the old systems might never be discontinued although the new systems were developed • Realizing the high cost of keeping and maintaining the old systems and especially the gateways (cutting gateways) • The usage of the new systems in a stable way (initial bugs and problems were solved and it was routinized)	(In early 1994), The decision and agreement to sell the document management product and all the support contracts to the financial officer to start a new business	The development of the new Windows-based office automation as the core product to focus on, in late 2000 and early 2001	*(Early 2002)* when the new MS-Windows products became stable and passed various tests	*(Started in 2002 for some clients)* The usage of new systems in a stable way and the huge cost of maintaining MS-DOS integrated systems stimulated a gradual marginalization of MS-DOS systems
Actions and the roles of embedded actors **Ceasing learning about old IS (ceasing learning SRM)** • Technical and HR managers gradually stopped R&D projects, training, and the hiring on the domains specific to old systems • IS Product managers in negotiation with support teams were carefully distinguishing between 'major' and 'minor' changes into the systems based on their learning commitments • Production and HR managers selectively isolated old and new systems teams and located them in distinct organizational units • R&D and HR managers proactively prevented narrow specialization in domains that might become obsolete quickly	Naturally happened due to the limited client base and limited requests for update and maintenance; but was sometimes suddenly reversed when an influential client urged the company to fix a major bug or add a major feature in the old products	Mainly by short-term isolation of the MS-DOS team from the new MS-Windows team during 2000 and 2001 when the new product was designed, and gradually outflow of the MS-DOS experts during 2002 and 2003	*(Mainly started in mid 2002 and mostly done through 2003)* Often by strict limits on not accepting (major) change requests and stopping support contracts and not extending them	*(Officially started in 2002, but continued as an ongoing process until 2005 and even later until 2009)* For each set of clients who were ready to replace the MS-DOS products, marginalization was executed with iterations back with handover (to customize the gateways for them) Often started by stopping formal learning actions such as official trainings and formal R&D projects and gradually extending to informal learning activities such as learning through support and debugging the old systems
Action: De-routinizing old IS (ceasing routinization SRM) • Marketing and Support managers specified clear deadlines for terminating the support of old systems • Support teams paralyzed the old systems in order not to be used by clients that already adopted new systems • HR managers helped some seasoned experts to move to other companies where they could find a relevant position • Production managers and support teams (when recommending to clients) removed the technological objects (e.g., operating systems, programming tools, and test tools) to prevent working on the old systems • Top managers (often CEOs) supported the teams of seasoned experts who were not willing to move to new domains to create their spin-offs • Production managers and the technical team of old IS carefully codified the old systems to allow them move on to new systems teams and for future occasional requests	Naturally and smoothly happened since the few employees were almost all technical experts, enthusiastic in working with MS-Windows; the two seasoned MS-DOD experts moved to the new company created by the financial officer	*(Gradually during 2002 and 2003; but still ongoing for the following five years)* Mainly by reducing the workload on the MS-DOS by stopping the support contracts and selling out the old products	Often by planned interventions to stop clients' access to the old systems and formally banning the internal team to work on the DOS systems (and asking for documenting the important aspects of the DOS systems in case of some urgent future requests) Actively creating spinoffs by MS-DOS experts and redirecting support requests to them	In a gradual process and often with the collaboration of each client for making it less accessible to use the old systems and assigning less work on the upgrading the old systems to the technical team

Ceasing learning ➜ de-routinization • When systems were not developed further, the clients have lower chance to keep using them (terminating support contracts)	*(During 1994 and 1995)* The CEO rejected clients' requests for supporting the MS-DOS systems by arguing that SmallCo no longer had the capabilities to do so	By selling out the old products and outsourcing the support contracts mainly in 2003	Extensively and often strongly forcing the termination of support contracts and rejecting the major changes	Often did not happen immediately, given the bargaining power of clients to ask for changes to the old systems
De-routinization ➜ ceasing learning • The fewer users and clients used the legacy system, the less the need for support and updating activities (thus less learning to do so)	*(Mainly in 1994, and less in 1995)* CEO and the head of technical team were constantly asking their programmers not to fix the bugs related to the old systems; yet this was once every two or three months interrupted due to the major support requests of clients	Having less major improvements on the old systems during 2002 and 2003 and discussing with their clients to use their new product instead of having the old products improved	Often happened for a large part of clients by rejecting their major change requests	Often effective for junior programmers and support team For senior experts, often happened in a gradual way with a lot of back and forth
Contextual conditions shaping realization • The relatively low bargaining power of companies against their influential clients (in all cases except LargStrict) ➜ forced the companies to come back occasionally to the old systems to relearn about the old systems in order to apply major change requests • Deep specialization ➜ made it hard for seasoned experts to stop relying on their deeply rooted expertise • The scope and interdependency of legacy systems (mainly in LargLenient and LargStrict) ➜ posed serious challenges on dissolving legacy resources; and made it difficult to move clients who have been using many inter-connected products on the old systems	Limited number of clients facilitated marginalizing the legacy IS The fact some of the experienced MS-DOS programmers joined the partner company also helped redirecting support requests to them	Low bargaining power against the clients and deep specialization in MS-DOS slowed down marginalization; despite limited number of commitments in terms of the clients and product diversity	The strong bargaining power vis-à-vis their clients and the isolated systems facilitated marginalizing MS-DOS systems Having and forming a strong network of spinoff companies helped redirect support requests to them	The strategy of admitting clients' requests and deep specialization in MS-DOS and the large amount of technical interdependencies between the old products took a lot of time and effort to marginalize the old IS

Table A3. Reversion

Reversion	SmallCo	MedCo	LargStrict	LargLenient
Triggers • Ongoing requests from the clients on the old system • Lack of readiness and instability of the new system • Risk of too early departure from incumbent systems	It almost never happened since the old systems were sold to partners and the new product was completely different from the old one; very occasional requests of the legacy clients that the partner company could not handle (almost a couple of times in the years between 1993 and 1995)	*(Mainly in 1996 and continued until 1999)* The deep, specialized technical capabilities triggered some efforts to keep MD-DOS and several of its related tools	*(Started in 1995 but was completed in 2000)* Long delay in developing the new integrated products and making them stable and some problems in the new products kept many clients asking for reverting back to MS-DOS	*(Mainly started in 1994)* Long delay (more than ten years) to develop new alternative systems and the ongoing demand for the MS-DOS systems triggered an extensive, long reversion
Actions and the roles of embedded actors • Re-legitimizing old IS (intensifying legitimization SRM) • Product managers were promoting old IS as still viable solutions in short-term • Product managers and marketing agents were constantly highlighting the relative advantages of old systems against the immature versions of new technologies		*(As an ongoing attempt during 1996-1999)* Often focused on internal attempts by senior MS-DOS experts who were arguing that many of the DOS-related technological capabilities were viable; the Technical Manager had to carefully ensure that even internal discussion regarding the problems of MS-DOS did not affect their credibility for clients who were still actively using the legacy products	*(For the entire period of 1995-2000 intensively and then as one of the side-production lines during 2000-2002 for urgent change requests)* A major effort to convince technical teams to keep the development and upgrade of the old systems especially when some other teams were working on the new systems (a feeling of being left behind)	*(Started mainly in early 1995 but continued actively until 2002 as a formal department and later as additional projects and later occasional projects until 2005)* Seriously pursued by production manager and support manager internally and by marketing team externally
Learning more about the old IS to maintain it (intensifying learning SRM) • Seasoned, specialized experts were actively identifying the shortcomings of old systems and improve them in an effective way • Senior experts in old IS rediscovered several untapped potentials of old systems and developed them • Product managers and development teams launched new development projects to improve the functionalities of old systems		Mainly through update and support activities to improve the old systems and therefore keep clients satisfied; Technical Manager had to motivate the young programmers to still see working on MS-DOS as an important job	Often through a selected list of change requests after being approved by the production manager to exclude major design changes and limit to really urgent requests	Often in the form of R&D projects on the MS-DOS and linking them with the major improvements in the old products
Re-legitimizing ➜ learning to improve • Knowing which aspects of the old IS were considered as viable and thus could and should be further improved meanwhile		*(Mainly from 1996-1999, for every major request from clients and every month for the small change requests)* Through the approval of most of change and support requests by the technical manager and the head of support team	Reapproving some of the change requests on the DOS systems that were initially rejected (and later accepted because the alternative systems were not yet ready and stable)	Approving many clients' major requests to update DOS-based products
Learning to improve ➜ re-legitimizing • The improvement of old IS turn it into a viable solution to be presented to the clients		*(Mainly from 1996-1999, often when a major limitation was discovered that could not be easily addressed in the current systems)* To highlight the relevant basic aspects of the product design and production and support routines	Often in a careful way to minimize major changes in old products	Through extending the support contracts and promoting the internal MS-DOS technical team by the managers
Contextual conditions shaping	Given the small	Being open to clients'	Large number of clients	Extensive client base and

realization • Legacy commitments to old IS (for LargLenient and MedCo) and the dominance of public, large clients (for MedCo and SmallCo) and increasing demand for support contracts ➜ forced the companies to keep learning on the old systems and improve them by dedicating teams to learn on and develop the old systems • The larger number of interconnected technological elements and products related to old IS (relevant in LargLenient and LargStrict) ➜ heightened the reversion since improving one part requires subsequent improvement in other parts • Deeper specialization in old IS (mainly in LargLenient and LargStrict) ➜ enabled the companies to dedicate further resources and capabilities to improve old systems	client base, the fact that there was only a separate product developed on MS-DOS, the reversion was not relevant; Some seasoned DOS experts helped making local, occasional improvements by moving to the other partner company that bought the old product and the support contracts	requests required extensive reversion for around four years Deep specialization on MS-DOS helped reversion to the MS-DOS and the related technological capabilities; especially by seasoned experts	using the multiple legacy systems and the fact that the new systems were not stable required considerable reversion efforts; yet it was challenging due to the dominance of the forward jumpers	large number of interconnected products combined with the strategy of the company to address the support requests of the clients resulted in extensive reversion activities; deep specialization on MS-DOS provided enough capability for reversion

Table A4. Handover

Handover	SmallCo	MedCo	LargLenient	LargStrict
Triggers • The development of the new systems and making sure they are stable • The shrinking market on the old systems (especially for the cases that they had major delay in developing new systems in comparison to the other competitors)	Given that the old product was completely sold out and the new document management system was completely different in terms of the design and technology, handover was very limited; using design ideas and transferring the brand (e.g., similar name of the product); for the clients who used the old product, the data entry to the new systems was done anew	Since the new product was a completely different product and did not use many of the aspects of the old products, it required very limited migration of legacy data for many clients who were using completely different products; in many cases, the clients (often with the help of MedCo) started entering new data to their systems	*(Started during 2000 and gradually expanded until the end of 2002)* The development of Windows-based systems in which the basic design of the products were also significantly improved comparing with the MS-DOS	*(Mainly at the end of 2001 for the core products and until the end of 2002 for other peripheral products)* When new systems became stable
Actions and the roles of embedded actors • Learning more about old IS to connect old and new IS (intensifying learning SRM) • Production managers and a selected team of highly skilled developers who mastered both old and new IS domains launched new development projects to connect old and new systems by developing and deploying gateways • Support teams were setting deadlines on using gateways for clients • Technical developers at the support team limited the functionalities of gateways (e.g., making them one-way ; that is, from the old to the new system)			*(From 2000-2002 as a formal line of production, later as extension projects until 2005 and afterward as occasional tasks when a major change to old products was done)* Extensively engaged in creating ways of converting the data from MS-DOS to MS-Windows especially because the basic product designs were changed (gateways created a major line of R&D and production projects)	*(During 2000-2002 in parallel with designing the new systems)* Mainly for converting the legacy data from systems to new integrated database structures (often for large clients); highly challenging because of the many interdependent products, data elements, and business processes that had to be integrated in the new systems and still support legacy data and functions
Reallocating viable resources from old to new IS path (ceasing resource complementarity SRM) • HR managers integrated old IS teams into new IS teams to leverage their common, basic knowledge • HR managers in collaboration with technical managers defined transitional tasks for seasoned experts of old IS to gradually move to new IS teams			Often the designers and support team were reallocated, but less programmers; transferring legacy data and the functionalities of the old systems became a major technical challenge; tendency to abandon many technological tools	Mainly focused on retraining the technical team and support teams to be able develop and support the new systems (less concern about the transfer of technological components and configurations since the entire product design was changed)
Learning to connect old and new ➜ reallocating from old to the new • Convertors allow for transferring legacy data from old to new IS • Gateways allow that many customers start working with the new systems through and with the help of the old IS (e.g., using old IS legacy data but through the applications of the new IS) • Connecting old and new IS makes the two systems both support the historical routines, roles, functions (backward compatibility) which then allows organizations and their clients be able to keep the same legacy of business processes and routines and use them in the new systems (thus for them not everything should be changed)			Often was extensively done for transferring data; also happened for transferring the configurations and settings between the various products in the entire integrated system (e.g., the compatibility between HR and finance modules)	Extensive projects and teams for developing gateways (in 2001 it became comparable with the projects dedicated to design new systems)
Reallocating from old to the new ➜ learning to connect old and new • Once the same users and same business processes, and same functions and roles were transferred to the new IS, it requires that old and new IS be more connected to bring all the related legacy data and features (more need to develop gateways and convertors)			Through creating dedicated R&D projects and teams to focus on learning more about MS-DOS for improving the performance of gateways	Dedicated teams for ensuring the compatibility of old and new system

Contextual conditions shaping realization • The technological differences between old and new systems posed challenges for connecting them via gateways, and created a knowledge gap between old and new IS teams, which challenged their migration to new team • The business process continuity in clients' side and the instability of the new systems (in all cases) increased the period during which gateways had to be deployed • The scope and interdependency of legacy systems (mainly in LargLenient and LargStrict) required the development of complex gateways in order to deal with many interconnections between old and new IS paths	The completely different product design in the new systems to comply with backward compatibility; limited commitment to clients to support and transfer their data	The completely different product design in the new systems to comply with backward compatibility	Because of the major technological differences between MS-DOS and Windows-based products, the extensive interdependencies between different products and technological components, and the need to guarantee business process continuity, the handover involved a lot of effort and expertise	The big difference between the design and production of products in MS-DOS and integrated systems in Windows and the many interdependencies in the legacy products in terms of the various data elements and business processes

Table A5. Iterations Between the Phases

Interactions Between the Phases	SmallCo	MedCo	LargStrict	LargLenient
Between Realization and Reversion **Discrediting old IS ⟷ re-legitimizing old IS** • Knowing which aspects are old-obsolete helps relegitimizing the other aspects that are old-viable • Tension between discrediting some aspects, but not delegitimizing the entire legacy IS **Learning to maintain the old ➜ discrediting old IS** • Discovering new shortcomings and realizing how fundamental are some of the limitations of the old IS and how costly to fix them	Not observed	A long set of back and forth discussions over around two years (1996-1998) to internally discredit MS-DOS but externally still keep it as a viable solution for their clients	*(For two years since 1993 as a continues discussion among different groups of designers, programmers, and gradually in late 1994 among the middle and top management)* Major tensions regarding the prestige of working on new systems and the feeling of being discriminated if asked to work on the old systems; by realizing the cost and efforts needed for making such change and improvement in the old IS	*(For around four years from 1994 to 2000, iteratively)* A lot of discussions due to the complexity of the products and their interdependencies Often happened at the very late phases that the cost of marinating MS-DOS systems appeared to become too much
Between Reversion and Handover **Learning to maintain the old ➜ learning to connect old and new** • Deeper knowledge that requires creating gateways (more at the fundamental levels of data structure compatibility and the consistency with the different operating systems, and the capability to support some specific business processes) **Learning to connect old and new ➜ learning to maintain the old** • Keeping legacy IS next to the new working and thus the need for maintenance continues • Deeper learning for creating gateways enhances the capability to find ways to improve the legacy and thus maintain it	Not observed	Not relevant since the new MS-Windows system was complete different from the old system and the new technical team was newly hired	*(Intensively happening during 2001 and 2002 in parallel with designing the new products)* To ensure that the changes to the legacy products are considered in (re)designing and implementing gateways and the new systems The complexity of the many interdependent products often required a lot of efforts to systematically examine the parts that can be affected by a change in the legacy IS and therefore be considered in designing gateways and new products	*(Frequently during 2000-2002 in parallel with redesigning new products)* Through formal, dedicated R&D projects on gateways and appointing the senior MS-DOS experts to them The senior MS-DOS experts acting as brokers between the gateways teams and MS-DOS support teams
Between Handover and Marginalization **Reallocating from old to new ➜ ceasing learning on the old** • Especially when the legacy data and functionalities is transferred, then the support activities can be stopped • Reallocating the technical teams to new IS development and support naturally reduces their learning on the legacy domains	Not observed	Not relevant since the new MS-Windows system was complete different from the old system and the new technical team was newly hired	*(During 2012 and 2013)* More as one-directional force for the production and support team by formally banning their engagement with the old products	*(Iteratively done from 2002 until 2005 for most of the clients and later for clients who were late in replacing MS-DOS)* Often by reallocating the production and R&D experts to work on the MS-Windows and for the clients when their legacy data and systems' configurations were transferred to the new systems; to customize gateways for clients
Between Marginalization and Reversion **De-routinizing ⟷ learning more about old IS to maintain** • Failing to de-routinize required learn further about the old systems since the customers kept asking changes as long as they used it • Even when legacy IS was de-routinized, some occasional requests will take back and requires relearning to maintain the legacy	Not observed	Several support requests that most of them were handled by MedCo but gradually some of them were outsourced to other companies after 2001	Limitedly observed after 2002; Only occasionally happened for very urgent problems requested by large companies	*(Iteratively, for each set of major change requests from 2002 until 2005 and later as occasional projects and tasks until 2009)* Frequently happening due to the extensive use of MS-DOS systems by large number of influential clients

Appendix B

Validity and Transparency Considerations

Table B1. Strategies for Enhancing the Validity and Transparency

Validity Measure	Strategies for Addressing
Construct validity	• Relying on a theoretically informed conceptual framework based on unlearning literature to operationalize various IS elements, discontinuance practices, and involved roles. • Sticking to interview protocols consisting of three sections: (1) getting to know the background of the company; (2) describing the whole replacement process; and (3) eliciting the discontinuance story by focusing on the various IS elements, practices and roles. • Adopting an open inquiry approach to reduce the bias due to imposing some terms on the informants (e.g., using "old stuff" instead of just saying "obsolete knowledge" (that could be interpreted differently by various actors). This also helped us capture a variety of elements in the discontinuance process. • Triangulation of data sources by asking questions at least from two relevant informants, and in case of inconsistencies, continuing the process to resolve it; covering external informants (e.g., ex-employees) to complement internal informants' opinions. • The main findings were presented to three companies (SMLFC, MedCo, and LargLenient) to identify and fix potential misunderstandings. • In the data analysis process, a second researcher re-coded 20% randomly selected of interviews (coding for IS elements and discontinuance practices). We discussed the new insights emerged in the second round of coding.
Internal validity	• (Through research design and analysis) by selecting extreme cases and through cross-case comparison we could capture the potential conditions that shaped the discontinuance process. • (During data collection) paying specific attention to "who did what," "when," and "why did actors did so" to identify the rationales and causes behind discontinuance practices.
External validity (generalization)	• There is no claim about generalization, since the aim of the paper is understanding deeply the discontinuance process. However, the comparison between companies helped us articulate mechanisms that explain how the process can be differently shaped by legacy IS conditions; thus showing the boundary conditions of the propositions.
Transparency and responsibility	• Relying on case study protocol and interview protocol. • Using a case study report for each company as a compendium of all information that helped us codify and articulate information from various sources. • Using ATLAS.ti for managing codes and memos, helping systematic iterations between data analysis stages, especially when some ambiguities required re-examination of the codes instances.