Chapter 4 Process Domain Risks

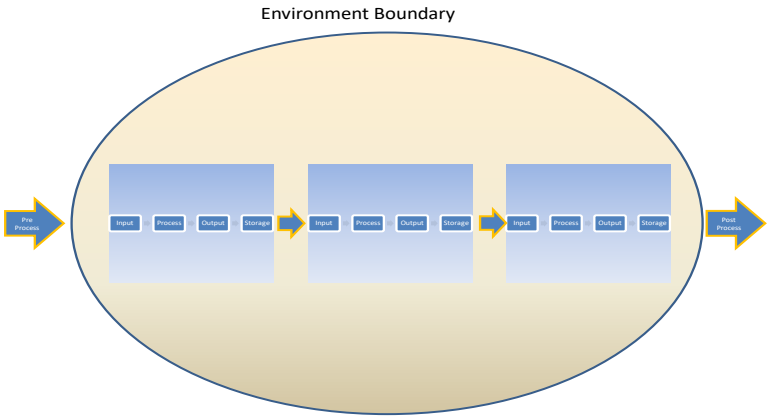
Overview

Example risk statement – The process fails to deliver only information content that is complete, current, accurate and valid.

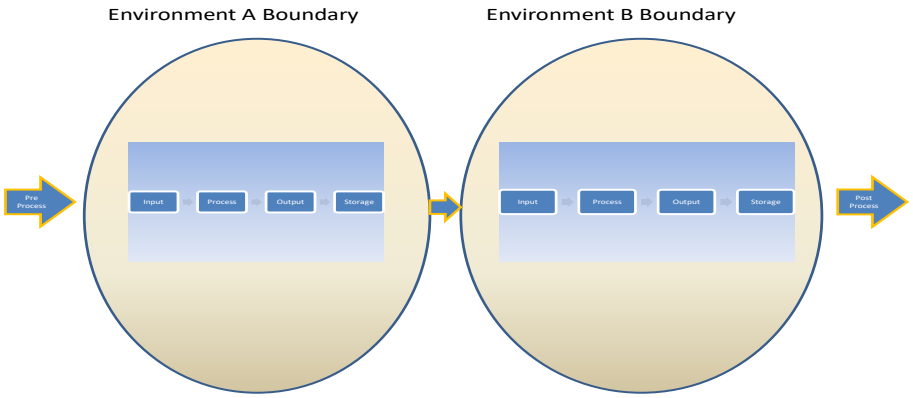
The processing life cycle includes pre-process, process and post-process phases. The pre-process phase refers to the data prior to its being recognized and captured by the entity. The post-process phase includes the use of information that has been produced by the system. The process phase, in its most parsimonious decomposition, includes input, processing, output and storage/disposal sub-phases. Although the process phase is informed by the pre-process sub-phase and the post-process sub-phase, these sub-phases are outside the boundaries of the process itself. For example, data acquired from another business unit may have been partially processed by that unit before becoming input in the process of interest for the current business unit. The business unit would take this into account in the way it handled the data. Similarly, the output processed by the business unit would be used by other business units and information about such uses would be taken into account in the way the business unit processed the output. Several processes may be linked to create a system within an environment, as illustrated in Panel A of Figure 4.1. Alternatively, two or more environments may act on a set of processes that together make up a system; for example, when the IT infrastructure used by a system is outsourced or when a system spans several geographic or jurisdictional boundaries, as illustrated in Panel B of Figure 4.1.

Figure 4.1 Information Processing

Panel A: Information Processing Within an Environment



Panel B: Information Processing Across Environments



Specifying System Boundaries

When assessing or assuring controls for a particular system it is essential to define its boundaries, to make it clear which processes are of interest and to establish the excluded pre-process and post-process phases.

Processing Phases and Sub-Phases

The processing lifecycle of information (as contrasted with the development lifecycle of information) consists of several phases and sub-phases:

1. Creation or identification of data
2. Measurement or observation
3. Documentation or recording
4. Input
5. Processing, change (including update or synchronization) or aggregation to transform data into information
6. Storage or archiving
7. Output or retrieval
8. Use
9. Destruction

These phases, in turn, can be decomposed into sub-phases, as summarized in Table 4.1.

Table 4.1 Summary of Processing Phases and Sub-phases

CODE	Phase	Explanation
Input Sub-Phases		
i-INIT	Initiation of phase • Identification or recognition of relevant events or instances;	activities that are required to commence input
i-CPTR	Input capture, observation or measurement/data preparation	activities required to gather content for entry
I-REG	Registration/recording/logging	capture of input activities for audit or monitoring
I-ACS	Matching classifications against access privileges	verifying individuals are authorized to access and approve input related activities
I-EC	Error prevention, detection and correction	errors or irregularities associated with input are proactively (or reactively) identified and rectified
I-META	Assignment/update of metadata	metadata associated with input is appropriately assigned or updated

CODE	Phase	Explanation
I-TRNS	Transmissiondistribution of information to other phases or processes	electronic or physical transfer of information from one system to another
I-BR	Backup and recovery	activities related to securely storing data for the purpose of recovery in the event of data loss during processing, including those copies sent offsite
I-CM	Maintenance and change management	activities involved in the regular upkeep or changes to the system
I-AT	Audit trail	capture of input activities for audit or monitoring

Processing Sub-Phases

P-INIT	Initiation of phase	activities that are required to commence processing
P-PROC	Processing • Transformation of input by aggregating information; • Performing calculations, logic functions and analyses; • Performing updates to temporary files (e.g. suspense files); • Performing updates to permanent or semi-permanent files, tables and databases; synchronizations	activities required to transform raw data into semi-processed data or processed output
P-REG	Registration/recording/logging	capture of processing activities for audit or monitoring
P-ACS	Matching classifications and privileges to permissions for functions/applications:	verifying individuals or software agents/programs are authorized to access and approve processing related activities
P-EC	Error prevention, detection and correction:	errors or irregularities associated with processing are proactively (or reactively) identified and rectified
P-META	Assignment/update of metadata	metadata associated with semi-processed or processed data is appropriately assigned or updated.
P-TRNS	Transmissiondistribution of information to other phases or processes	electronic or physical transfer of information from one system to another
P-BR	Backup and recovery	activities related to securely storing data for the purpose of recovery in the event of data loss during processing, including those copies sent offsite
P-CM	Maintenance and change management	activities involved in the regular upkeep or changes to the system
P-AT	Audit trail	capture of processing activities for audit or monitoring

Output Sub-Phases

O-INIT	Initiation of phase	activities that are required to commence output
O-OTPT	Output • Output display; • Transmission and distribution to users	activities related to transforming processed data into usable information (e.g. a report, spreadsheet, statistics, etc.).

CODE	Phase	Explanation
O-RTRV	Retrieval:	activities related to retrieving or extracting processed data from a repository
O-AT	Registration/recording/logging	capture of output activities for audit or monitoring
O-ACS	Matching classifications and privileges to permissions for access	verifying individuals or software agents/programs are authorized to access and approve output related activities
O-EC	Error prevention, detection and correction	errors or irregularities associated with output are proactively (or reactively) identified and rectified
O-META	Assignment/update of metadata	metadata associated with output is appropriately assigned or updated
O-TRNS	Transmission: electronic or physical transfer of information from one system to another.	electronic or physical transfer of information from one system to another
O-BR	Backup and recovery	activities related to securely storing data for the purpose of recovery in the event of data loss during processing, including those copies sent offsite
O-CM	Maintenance and change management	activities involved in the regular upkeep or changes to the system

Storage Sub-Phases (**not** included in the document)

S-INIT	Initiation of phase	activities that are required to commence storage
S-AT	Registration/recording/logging	capture of storage activities for audit or monitoring
S-MTCH	Matching classifications against access privileges	verifying individuals or software agents/program are authorized to access and approve storage related activities
S-EC	Error prevention, detection and correction	ensures that errors or irregularities associated with storage are proactively (or reactively) identified and rectified
S-META	Assignment/update of metadata	metadata associated with stored data is appropriately assigned or updated.
S-TRNS	Transmission/distribution of information to other phases or processes	electronic or physical transfer of information from one system to another
S-RTN	Retention in onsite/offsite storage	activities to ensure that information (and the media it resides upon) is retained for a period of time that reflects the operational, statutory, and regulatory requirements of the entity
S-BR	Backup and recovery, including onsite vs. offsite storage	activities with storing/restoring processed output, including those activities pertaining to offsite storage, restoration and testing
S-CM	Maintenance and change management	activities involved in the regular upkeep or changes to the system
S-DISP	Disposal	destruction of data (or the medium that stores the data) to the point that the data is not accessible or usable.

Flaws in the creation, operation and change of a process can result in a process that does not have appropriate information integrity enablers and controls, resulting in content and related metadata that are incomplete, out of date, incorrect or invalid. Risk magnifiers such as complexity, inherent nature of the process, the presence of malicious intent and other such factors increase process domain risks.

Process domain risks by information integrity attribute are summarized below by processing phase and sub-phase.

Input Phase

Of all the causes of processing errors, the most common are those related to the input phase. Input phase errors can result from flaws in input capture procedures such as incorrect recording of information by data entry personnel, errors by customers using web forms and errors by automated input devices. Significant attention has been devoted by information system analysts and engineers to reduce the frequency of input phase errors by improving the design of forms, screens and data entry system interfaces and introducing new data capture processes, particularly automated input devices to eliminate one of the most significant sources of error - human error.

Type of Input

The input phase of processing can involve a variety of media and data types, including name and address data, demographic data, geospatial data, and business rules used for data validation. The following types of input are identified in this publication and were specifically considered in identifying risks and controls.

- Automated input (Sensor based, EDI, RFI, etc.)
- Externally generated by Outsourcers (B2)
- Externally generated by Business (B2B) - Network
- Externally by Consumers (B2C) - Network
- Internal-system generated
- Internal- manually inputted
- Transfer of non-routine (data file restoration, conversion, merger, acquisition, other)

Each of these input capture methods is subject to particular risks as described below by information integrity attribute.

Completeness

Omitted/missing data—During the input phase, it is possible to omit relevant events from being recorded. For example, goods might be shipped, but the shipment may not be recorded; purchases may be received, but the receiving report transaction may not be initiated. In these cases, the data representing the event would be missing; although the events themselves have occurred. In some cases, as in the receiving report example, external parties with significant incentives to do so (e.g., suppliers) would likely bring the omission to someone's attention. In other cases, such as the unrecorded shipment, there may be little incentive for anyone to make the enterprise aware of the omission.

Lost transaction/data—It could be that a transaction is, in fact, initiated but is then "lost" sometime before it is actually processed. This type of error is similar to the error of omitting a transaction required to reflect an economic event; however, there is one major difference between these two types of omissions. In the case of a transaction which is never prepared to reflect an economic event, it may be difficult to identify the nature of the omission. In the case of lost data, assuming that adequate records are kept at the transaction initiation site, it may be possible, based on such records, to trace, or even recreate, the information that was lost.

Incomplete transmission—Communications such as e-mail, text messaging and voice mail are critically important for today's business activity. It is often assumed that all sent communications such as e-mails are received, but this may not be the case for a variety of reasons and can have significant business impacts.

Currency/Timeliness

Delayed data entry/cut-off errors—The last batch of transactions initiated near a period-end might not make the cut-off. Or, error correction transactions may not be processed in the appropriate period. In this case, the data would be missing in the correct period but would appear in the

following period, perhaps causing yet another error which would wash out the effect of the error in the previous period.

Delayed transmission—Delayed transmissions may impact the acceptability of information. For example, in certain jurisdictions, there are limitations as to the period of time a telecommunication company can "back bill" calls. Thus if CDR information needs is not relayed from the switch to the bill system in a timely manner the revenue is lost forever.[1]

Accuracy/Correctness

Data entry/keying/scanning errors— Inaccuracies in data entry, for example the incorrect keying of transaction information such as date, amount, customer, vendor, quantity or transaction code, plague computer-based information systems.

Garbled transmission—Error correction algorithms correct most garbled transmissions. When this is not the case, the risk here is in accepting data with embedded errors and discovering later that garbling has occurred when it is no longer simple to correct them.

Inadequate review of content correction or generation algorithms – mobile device users often rely on auto-correct or text prediction algorithms which can generate incorrect predictions or corrections. This would apply to spelling and grammar checking features in word processing programs and web browsers.

Validity/Authenticity/Authorization/Non-Repudiation

Insufficient editing/validation—In many organizations, simple validation rules are used for controlling the integrity of input. However, simple rules may not catch errors resulting from complex conditions and complex business rules used for validating data can be misapplied. The increasing use of online real time systems and web-based forms has tended to reduce the amount of data checking performed at the data entry phase and this, in turn, increases the likelihood of errors. Data validation rules may be sufficient for one process, but may not take into account multiple processes with mismatched syntax (e.g., first name followed by last name vs. last name

[1] Charkowicz, Deborah; op. cit.

followed by first name); data format (e.g., 8 byte date field vs. 6 byte date field); or code structures (e.g., male or female vs. 1 or 2). In the absence of a consolidated data collection point providing for comprehensive integrity checking, mismatched syntax, data format or code structures will not be noticed.

Unauthorized data/overrides—Sales order clerks may process unauthorized discounts or approve unauthorized expenditures for reimbursement.

Duplicated data—Failure to cancel documents to prevent their re-use or the use of copies (e.g., photocopies, faxes, or scans) rather than original forms, can lead to the multiple processing of transactions. System failures, retransmissions and backup restorations can lead to the duplication of input data. Data can also be duplicated when it is being converted to be imported into a different system.

Fraudulent transaction/data—The processing of fraudulent transactions is the single largest cause of computer-related fraud. The difference between processing unauthorized transactions and fraudulent transactions is sometimes in the eye of the beholder.[2]

Interception of transmission/tampering—When unprotected information is intercepted it can be changed without detection.

Some of the causes of input phase risks include:

- Keying errors

- Failure to cancel documents to prevent their re-use or the use of copies (e.g., photocopies, faxes, or scans) rather than original forms, can lead to the multiple processing of transactions.

---

[2] Fraud is defined as: an intentional act to deceive or mislead, convert assets to one's own benefit, or make intentional false statements or representations often accompanied by omission, manipulation of documents or collusion. Fraud requires three elements: 1) A perpetrator with the motivation to commit the fraud; 2) The opportunity to commit and conceal the fraud; and 3) Lack of integrity or ethics or other attitude that does not reign in the first two elements.

- Operator failure to initiate, execute, restart or complete a process

- Poor design of source documents, data entry screens and user interfaces that lead users to commit mode errors, description errors, consistency errors, capture errors and activation errors.[3]

- Changes in the method of carrying out the work

- Working conditions and poor ergonomic design of work stations, contributing to data entry and operator fatigue

- Poor hiring, training, documentation (e.g., user manual, operations manual) and supervision of data entry and operations personnel

- Equipment failures (e.g., POS, scanner, sensor); system outages due to server overloads or denial of service attacks.

- Software flaws (e.g., spam filters may drop certain e-mails without notifying either sender or intended recipient).

- Media containing input data may be lost during physical movement.

- Unauthorized overrides of controls such as limit checks, reasonableness checks or logical relationship checks

- Intentional suppression or delay of transactions by individuals within the entity, or by management.

- Fraudulent transactions and/or master file change documents or inputs.

---

[3] Norman, Donald A.; 1983

Risk Magnifiers

Complexity

Complexity factors may magnify risks in the input phase. Such factors include:

- Multiple divisions may share a process but may have different views about inputs, uses of outputs, and priority for process maintenance and change management.

- Multiple segments and geographic regions sharing the system environment may create difficulties for prioritizing, completing and maintaining security, availability and other enablers affecting a particular process.

- Multiple systems, environments, operating systems, interfaces, data-types, handoff points, data formats, logs, log formats, and differences in codes may make it difficult to operate processes consistently.

- Many and distributed business segments, many and diverse sources of input, and many users with varying needs create difficulties in creating and updating process phases.

- Multiple types of entry devices and subsystems increase difficulties of defining data capture steps

- Third party information service providers may themselves rely on other parties for processing, recovery, and other functions that impact input received by the entity.

- Various/numerous data flows (e.g. car manufacturer aggregating sales data from all its dealers), users and uses (e.g. marketing and accounting capturing sales data.)

Inherent Nature of Content

The dynamic nature of some types of content can pose challenges to information integrity. For example, customer data can change frequently without notice being given to the entity. For example, one entity deals with “age” by only recording birth date, and does not even attempt to

deal with marital status (other than on point-in-time documents, such as applications for credit) since there is no reliable "trigger" that enables the system to easily recognize changes. It is unlikely that a customer will volunteer a divorce, separation or a common-law arrangement unless there is some financial arrangement that, as a consequence, needs adjustment.

Currency data can create difficulties for financial organizations, since it is so dynamic, multifaceted and prolific. Aside from the number of currencies that can be converted, there is the added dimension of the type of exchange rate being used, since there are different rates applied in different settings (e.g., US $ charged on a Canadian $ Visa account, vs. when withdrawn from a US $ ATM on a Canadian $ account, vs. large dollar corporate transactions.). This type of information is very transient and data stores often contain information from several time periods. Users can easily apply the wrong rates.4

Malicious Intent

Risks to a processing system can arise from intentional malicious or through unintentional errors, omissions, accidents or acts of acts of nature. Unintentional events such as erroneous data that is entered into the system can be the result of incomplete editing or validation, poor source document design, ergonomically poor working conditions, data entry errors and failing to include users in deploying changes to input screens or devices.

Users with malicious intent can also exploit unintentional errors (e.g., input handling errors such as buffer overflows that hackers can exploit to gain unauthorized access to a system). A malicious user can input fraudulent data (e.g. a fraudulent order) or a user may intentionally configure changes or input edit controls to permit the inclusion of unauthorized transactions. The opportunity to commit fraud typically involves access to real assets, or proxies which can be converted into assets, and the opportunity to obtain and remove those assets. Concealment of a fraud can involve disguising the actual fraud by making it look like an error, misrepresenting the

---

4 For example, an Australian man was able to purchase Sri Lankan rupees for (Australian) $104,500 and then sell them to another bank the next day for $440,258. (The first bank's computer had displayed the Central Pacific franc rate in the rupee position.) Because of the circumstances surrounding the bank's error, a judge ruled that the man had acted without intended fraud and could keep his windfall of $335,758 (Neumann, Peter G.; 1992d)

values of physical assets, manipulating the book values of assets, or confusing the identity of the fraud perpetrator.

Link to IS Environment Domain

Poor controls in the Environment domain may result in errors and abuses of input, for example:

- Incentive programs that reward employees for entering data as fast as possible, have a higher risk of input error than those that have a balanced approach that rewards both speed and accuracy.

- Users are not consulted about the design of input entry screens resulting in keying errors.

- Lack of adequate audit trails allows the entry of unauthorized transactions without being detected.

- Omission of required input capture steps may result in incomplete, delayed or inaccurate input.

- Failing to revise input capture for changes within the environment may result in integrity flaws in input capture (and vice-versa)

- Use of applications outside the production environment, such as spreadsheets, that are not subject to the required developmental controls or testing protocols

Table 4.2 illustrates some of the risks by sub-phase and information system development lifecycle category.

Table 4.2 Examples of Information Integrity Risks by Sub-Phases of Input

Phase (General) **Phase (Specific)**		**Creation**	**Operation**	**Change**
Initiation of routine	Link to Assessment of the Design of the Process	Requirements definition not complete or accurate The starting point for the recognition of content is not accurately defined Origin of content not specified Design, Development and Deployment of effective manual and automated initiation procedures to enact requirements not accurate or complete	Initiation procedures do not operate reliably and continuously as designed Operations procedures during system processing may affect processing integrity Errors may flow from one subsystem to another	Changes to source of the content not reflected in the initiation process(es) Changes in the initiation process(es) not reflected in the content (e.g., an authorization procedure requires metadata on access privileges that is not provided in the content) Process not revised in response to changes in the IS environment, leaving exploitable gaps in the process
	Link to Assessment of the Environment that the Process Operates in	Omission of required processing steps may result in incomplete, delayed or inaccurate processing Flaws in design of enablers such as security, availability, etc. may permit errors and abuses of processing	Errors and delays in system operations may affect processing integrity	The process enablers in the Environment may change but this change may not be assessed for its impact on the process, leaving exploitable gaps in the process
Registration/ recording/ logging		Requirements definition for registration/ recording/ logging not complete or accurate Failure to define what should be	Registration/ recording/ logging not executed Logs are not correctly aggregated	Changes to content not reflected in the registration/ recording/ logging process(es) Changes in the registration/ recording/ logging process(es)

Phase (General) Phase (Specific)	Creation	Operation	Change
	logged Sensitive content is not classified. Definition, Design, Development and Deployment of effective manual and automated procedures to enact registration/ recording/ logging requirements not accurate or complete Incomplete capture of logging events	from multiple systems Logs from different systems, devices, firewalls, etc. are not standardized to assist users in identifying patterns Tampering with logs	not reflected in the content. New systems are not integrated into logging process
Match access/ privilege to initiate	User access is not defined at the system resource/table/function level System is not designed to limit access on a “need to know, need to do basis	User are able to initiate input that is unrelated to their job function or expertise	Personnel change (promotion, transfers, terminations, etc.) are not implemented in a timely manner

Phase (General) Phase (Specific)		Creation	Operation	Change
Input Capture	Capture: Automated input (Sensor based, EDI, RFID, etc.)	Incomplete editing and validation rules for raw data Incomplete checking of semi-processed data when next processing step is removed by time or space from the previous step Incorrectly configured/modified interface Lack of filters, meta data, data integration hub, consolidated data collection points Error messages display sensitive information upon receipt of incorrect input., which may allow unauthorized access to the system Unreliable sensing and transmission equipment in automated systems such as point-of-sale systems in retail stores Delayed transmission Automated capture of information is non-compliant with relevant policies, statutes or regulations	Incomplete collation due to missing or lost data/files Delayed receipt of content that is to be included in final output Duplicated input Unauthorized content (especially master file record) inserted during data entry or processing Unauthorized access: theft, tampering Fraudulent input Incorrect data validation Data in lookup tables not approved Fraudulent actions perpetrated by individuals or management	Changes in systems or output specifications may not be reflected in the data capture requirements Unauthorized Configuration/software changes Interface system changes prevent capture of information Changes in data format are not reflected in the interface Poor system documentation limits the ability of support personnel to make reliable changes

Phase (General) Phase (Specific)		Creation	Operation	Change
	Capture: Externally generated by Outsourcers (B2)	See above Incompletely defined metrics to monitor inputs handled by outsourced entities Oversight does not involve independent verification of input monitoring metrics or other procedures	See above Fail to comply with service level agreement Fail to monitor metrics defined	See above; system changes implemented by outsourcer are not communicated to the entity
	Capture: Externally generated by Business (B2B) - Network	See above, plus: Information integrity responsibilities are not assigned or defined between supplier(s) and the customer Lack of synchronization between two parties Poor design of input screens Access to system resources is not designed to handle access by external parties	See above, plus: Unauthorized employee submits order Unauthorized data is submitted through the 3rd party's system Web session terminates during processing Lack of synchronization between two parties Supplier access to internal systems is not managed	See above, plus: System changes are not communicated to supplier thereby preventing processing of transactions
	Capture: Externally by Consumers (B2C) – Network	See above, plus: Users access agreements are	See above, plus: Web session terminates during	See above, plus: Failure to maintain routine maintenance (e.g. patch levels) to

Phase (General) Phase (Specific)		Creation	Operation	Change
		incorrectly/incompletely defined Bad design of input screens Incomplete editing/ validation rules	processing Submission of invalid/ unauthorized data: • Malicious data (e.g. SQL injection) • Dummy data • Fraudulent data (e.g. credit card info) Inexperienced users	protect against malicious users and malicious code
	Capture: Internal-system generated	See above, plus: Incomplete data mapping, models and other documentation Incomplete user review Design inhibits downstream use of data Failure to develop new documentation and procedures when processing changes are implemented	See above, plus: Incomplete upstream data feed/job schedule Delayed upstream data feed/job schedule Inaccurate upstream data feed Corrupted job schedule	See above, plus: Job streams modified in an unauthorized manner

Phase (General) Phase (Specific)		Creation	Operation	Change
	Capture: Internal-manually inputted	See above, plus: Incomplete anticipation procedures Incomplete editing/ validation Poor design of input screens • Poor source document design • Poor user interface design • Delays in sending, entering and processing data, leading to the users' perception that the data are in error, when they are merely incomplete • Changes in the method of carrying out the work • System-generated transactions, with the systems not being adequately "tuned" to their environment • Working conditions and poor ergonomic design of work stations, contributing to operator error	See above, plus: Data entry keying errors: • Misspelling • Transposition of numerals • Incorrect or omitted codes • Data placed in wrong fields • Unrecognizable names, nicknames, abbreviations, acronyms • Appear approved, but not authentic, forged • Unapproved • Approved, but *ultra vires* • Approved, but fail to comply with policies, statutes or regulations Reliance on false information (e.g. stock pump & dump schemes)	See above, plus: Manual input process is negatively impacted by system changes (i.e. users were not consulted)

Phase (General) Phase (Specific)		Creation	Operation	Change
		• Equipment failures • Poor hiring, training, documentation and supervision of data entry personnel	Poor training	
	Capture: Transfer of non-routine (merger, acquisition, other)	Incomplete documentation of data structures (e.g. legacy systems) Data conversion strategies do not have defined audit requirements Poor designed data conversion strategies	Data conversion omissions Data conversion delays, cut-off errors Data conversion errors in details or summary figures Data conversion: unauthorized initiation or modification	Adding unauthorized vendor, employee or other payee during conversion
Input error prevention, detection and correction		Incomplete definition of error codes System is not designed to identify all errors Review of unidentified errors does not result in the update of error identification routines	Incomplete error correction Errors are corrected too late Error correction is not appropriate for data entry or is incomplete Error correction is implemented in unauthorized manner Poor training	Error codes from new systems are not integrated into the error identification process

Phase (General) **Phase (Specific)**		**Creation**	**Operation**	**Change**
Assignment/ Update of metadata	General	Failure to define metadata Sensitivity levels are not defined Failure to design metadata	Failure to assign metadata	Failure to update metadata
	Audit trail	Incomplete audit trails Design of the audit trail may omit manual procedures or fail to capture other inputs in a complete manner	Sensitive content not classified Incorrect/incomplete audit trails Audit trails are tampered/modified in an unauthorized manner	Failure to update audit trail
Transmission		Transmission standards and protocols are not defined or agreed (e.g. sending sensitive material over public networks, email, ftp, etc.) Server capacity insufficient for the volume of processing (overload) Spam filter Incomplete load testing Spoofing	Delayed transmission Incomplete/lost transmission Garbled transmission Duplicated transmission Unauthorized manipulation of data in transit	Transmission modified in an unauthorized manner
Maintenance and		Failure to define change	Failure to roll out changes to all	Failure to update input phase for

Phase (General) Phase (Specific)	Creation	Operation	Change
change management	requirements to all sub-phases caused by a change in one; e.g., new edit and validation procedures to accommodate new input media types Failure to develop new documentation and procedures when input process changes are implemented	input capture locations, etc. Unauthorized changes to input	changes in business process Failure to update the change process itself to respond to changes in the business, the organization chart, processes, technology, and personnel

Processing Phase

The processing phase involves the transformation of data into output for display, distribution or storage. Processing errors can result from incorrect logic applied in a program when originally written or applied during program maintenance to repair identified flaws or to enhance program functionality. Although processing phase errors are less frequent than input phase errors, their impact can be much more significant, since they can affect entire files and can last for extended periods of time if they are due to errors embedded within program logic.

Type of Processing

Content processing includes:

- Aggregating
- Calculating,
- Choosing
- Analysing
- Updating
- Manipulating
- Reporting

Processing phase risks are described below by information integrity attribute.

Completeness

Incomplete processing—Incomplete processing is a fairly common problem in many installations. It may be a result of program logic errors, operator errors or even equipment problems. For example, at year-end, an entity may process only 51 weeks of information when preparing annual

summaries of key accounts; or, it may consolidate some, but not all, branches in preparing organization-wide reports.

Currency/Timeliness

Cut-off errors—Failure to process accounting information in the right accounting period can lead to incomplete information in one period, followed by a compensating error in the subsequent period. These errors can lead to errors in decisions based upon the reports produced from the data.

Back-up/recovery delays—Delays in restoring backups after a system failure can lead to non-current information, particularly in online systems such as e-commerce systems. Such systems require the use of online recovery systems based on data mirroring techniques to handle routine recovery of online transactions.

Accuracy/Correctness

Wrong file—A common processing phase error is the use of the wrong file or the wrong version of a file. This may include either master or transaction files, or both. This can lead to the loss of data or the mixing up of data from various sources, time periods, etc.

Incorrect logic—Processing errors can result from programming flaws such as incorrect logic applied in a program. According to literature in this area, many of the top programming errors are not well understood by programmers; their avoidance is not widely taught by computer science programs; and their presence is frequently not tested by organizations developing software for sale.[5]

[5] "CWE/SANS TOP 25 Most Dangerous Programming Errors" at http://www.sans.org/top25errors/?cat=top25#cat1; accessed May 11, 2009.

Validity/Authenticity/Authorization/Non-Repudiation

Duplicate processing—Data processing may be reperformed multiple times during processing. For example, the data for a particular branch may be processed more than once or the data for a particular time period (e.g., day, week, month) may be processed more than once.[6]

Unauthorized/untested logic—testing and maintenance errors—Erroneous logic may result when a programmer substitutes what he/she believes to be a preferable method of implementing a particular concept or policy for that specified, and inadvertently changes the algorithm behind the specification. A commonly used example of this is the substitution of average cost in a FIFO or LIFO costing system in an attempt to make the program operate more efficiently.[7]

Fraudulent logic and malware—The literature on computer fraud contains many examples of fraudulent coding designed to enrich a programmer. While the incidence of programmer fraud is comparatively low, the amounts involved are comparatively large. A category of software known as malware, which stands for malicious software, can lead to the corruption or destruction of information. Malware includes viruses, worms, trojan horses, time bombs and hacker penetration.

Overrides of programmed controls—Users sometimes intentionally override programmed procedures to either make systems "faster" or "more efficient."

Some of the causes of processing phase errors include:

- Incomplete or incorrect user requirements that result from inadequate assessment of end-user needs.

---

[6] For example, a computer glitch at Chemical Bank of New York double-charged every withdrawal and affected at least 70,000 of the bank's one million customers *Globe and Mail*, 18 February 1994

[7] On 31 May 2004 the Royal Bank of Canada suffered a failure due to a program maintenance error. The disruption lasted until June 4, preventing millions of customers from accessing funds or receiving electronic payroll transfers from employers.Weber, T., 2004.

- Inaccurate translation of user requirements into the design specifications which govern the programming phase of systems development.

- Incorrect program logic due to reliance on incompetent, poorly-trained or poorly-supervised programming personnel, especially logic involving currencies and time; system-generated transactions, with the systems not being adequately “tuned” to their environment.

- Fraudulent code due to insufficient segregation of incompatible functions.

- Incorrect logic due to an incomplete or erroneous definition of user requirements, or inaccurate translation of user requirements into the design specifications which govern the programming phase of systems development.

- Incorrect logic can result from hiring and use of incompetent, poorly-trained or poorly-supervised programming personnel.

- Users that have access to incomplete documentation may not be able to manage the processes – resulting in errors.

- User overrides may result from user requirements not being taken into consideration when the system is designed or from users working around errors in the programmed code.

- Volume of system auto-generated transactions may overwhelm users and undermine user oversight of processing

- Operator overrides and equipment failures are also potential causes of processing phase errors.

- Changes in the method of carrying out the work

- File handling and use errors due to inadequate label checking, operator errors or processing delays

- Duplication errors in processing can result from poor file management, faulty back-up/recovery procedures or the absence of well-defined operating procedures. They can also result from the use of incompetent or poorly trained personnel or working conditions involving a great degree of stress, time pressure, etc.

Risk Magnifiers

Complexity

Complexity factors that may magnify risks in the processing phase include:

- Multiple divisions may share a process but may have different views about inputs and different uses of outputs as well as their priority for process maintenance and change management.

- Multiple segments and geographic regions sharing the system environment may create difficulties for prioritizing, completing and maintaining security, availability and other enablers affecting a particular process.

- Multiple environments, operating systems, applications, etc. may make it difficult to maintain consistency of operations and force data travel between systems through middleware or interfaces

- Part of the audit trail may be in the form of documentation about transformations and aggregations of data/transactions into summary figures.

- Number of changes to the application, process, or interfaces

Nature of Processing

Time: Research indicates that transformations involving time are particularly prone to logic errors, including synchronizing calculations with a "clock setting" or taking the difference between dates.[8]

User developed applications: This is in contrast to applications that are acquired, developed and managed by the technology department at the organization. These include spreadsheets, "rogue-

[8] As summarized in *Computer Control & Audit Guide* 14th edition Chapter 2 by J.E. Boritz, Waterloo: UWCISA, 2008.

use" of public clouds (e.g. Amazon Web Services), databases, and the like. Factors, related to user developed applications, that can magnify risks in the processing phase include:

- Unawareness of significance of applications that are developed and managed by users
- Policy that outlines the appropriateness of using user managed applications instead of main information system is defined in an incomplete manner
- Undefined accountability, ownership, custodianship
- Delays in receipt of data from upstream processes
- Information may be lost or corrupted traveling back and forth between main processing environment, intermediary points, and the user developed application environment
- Non-compliance with the entity's IT design, development and deployment standards
- Use of incorrect version, accidental modification
- Lack of coordination, synchronization with upstream processes
- Files are processed by unauthorized users
- Unauthorized logic
- User developed application environments that rely on secondary systems instead of source system are prone to delays and errors introduced in such systems (e.g. data warehouses)
- User developed applications are not subject to the same standards as regular IS changes

Malicious Intent

Processing can be impacted by the development of poor quality applications caused by inadequate resources, unrealistic IT project deadlines, inadequate definition of requirements, and unqualified systems personnel. Processing can be impaired due to over-relying on user developed applications (i.e., instead of the IT department), process overrides, and failing to assign information integrity responsibilities to operational personnel.

Processing can also be impacted by malicious acts such as the insertion of fraudulent code into programs during program development, change or patching and manipulation of master file records (e.g., set-up fictitious vendors which are used to route goods).

Link to IS Environment Domain

Flaws in the creation, operation and change of environment level enablers such as security, availability, etc. may permit errors and abuses of processing. For example, omission of required processing steps may result in incomplete, delayed or inaccurate processing. Processing integrity may also be negatively impacted by errors and delays in the system operations. Processes that fail to include changes made in the environment may result in processing with integrity impairments (and vice versa).

Table 4.3 illustrates some of the risks by enabler and risk category.

Table 4.3 Examples of Information Integrity Risks by Sub-Phases of Process

Phase (General) **Phase (Specific)**		**Creation**	**Operation**	**Change**
Initiation of routine	Link to Assessment of the Design of the Process	Requirements definition errors during system development may affect fitness for use and all core attributes of information integrity. Design errors during system development may affect fitness for use and all core attributes of information integrity	Operations procedures during system processing may affect processing integrity Errors may flow from one subsystem to another	The change management process may not have the resources to update the process in response to changes in the IS environment, or vice versa, leaving exploitable gaps in the process
	Link to Assessment of the Environment that the Process Operates in	Omission of required processing steps may result in incomplete, delayed or inaccurate processing Flaws in design of enablers such as security, availability, etc. may permit errors and abuses of processing	Errors and delays in system operations may affect processing integrity	The process enablers in the Environment may change but this change may not be assessed for its impact on the process, and vice versa, leaving exploitable gaps in the process

Phase (General) **Phase (Specific)**	**Creation**	**Operation**	**Change**
Registration/ recording/ logging	Failure to define what should be logged Registration/ recording/ logging not applied to all relevant processing	Incomplete registration/ recording/ logging of events Logs are not correctly/ completely aggregated from multiple systems Logs are not normalized to assist users in identifying patterns Tampering with logs Registration/ recording/ logging files lost, destroyed or corrupted	New systems are not integrated into logging process
Match access/ privilege to execute functions/ applications	User access is not defined at the system resource/table/function level Users access agreements are incorrectly/incompletely defined System is not designed to limit access on a “need to know, need to do basis”	User are able to access information that is unrelated to their job function	Personnel change (promotion, transfers, terminations, etc.) are not implemented in a timely manner

Phase (General) Phase (Specific)		Creation	Operation	Change
Processing	Aggregating	Data aggregation requirements are not completely documented or defined Metrics to monitor processing are not defined Incorrectly coded logic	Non-current/incorrect files used, incorrect items aggregated Duplicate processing	Job streams are modified in an unauthorized manner
	Updates temporary files (e.g. suspense files)	Suspense review standards are not defined Documentation is not maintained Incorrectly coded logic	Incorrect files Duplicate processing Poor training Transactions are not reprocessed in a timely manner	System changes are not reflected in processes involving temporary files
	Updates to perm or semi-perm files, tables, databases	Suspense review standards are not defined Documentation is not maintained Incorrectly coded logic	Incomplete processing Lost information Incorrect files Non-current files Delayed processing Duplicated processing	System changes are not reflected in processes involving semi-permanent files

Phase (General) Phase (Specific)		Creation	Operation	Change
			User/operator overrides of business rules	
	Executing the logic, applications, computations, and analyses	Scope of process is not correctly/ completely defined; thereby omitting key requirements Upstream and downstream impacts are not defined in a complete manner Information integrity responsibilities are not assigned or defined for operational personnel Data conversion does not address processing requirements (e.g. hard coded values) Sensitivity levels are not defined Poorly designed data conversion strategies Incorrect input, output statements Incorrect assignment statements Incorrect branching statements Incorrect looping statements	Incomplete processing Delayed processing Poor training Capacity of the system is below optimal performance requirements Rush to meet processing deadlines results in errors	System changes are not tested (e.g. stress testing, volume testing, etc.) in a complete manner Misapplying IFRS Data hard coded into logic is out of date (e.g. sales tax rate has been reduced)

Phase (General) **Phase (Specific)**		**Creation**	**Operation**	**Change**
		Incorrect data type definitions Untested code Poor user manuals, poor training, Processing logic modifies meta-data in an incomplete, incorrect manner Incorrect general logic Incorrect time logic Unauthorized logic		
	Back-up and Recovery		Irregular backup procedures lack of familiarity with irregular backup process Offline/distributed devices (e.g. laptops)are not regularly backed up	New applications/ processes Application/process updates and changes New personnel Complexity of program or process maintenance or change
Processing phase error prevention, detection and correction		Incomplete definition of error codes System is not designed to identify	Incomplete error correction Errors are corrected too late Error correction is not appropriate	Error codes from new systems are not integrated into the error identification process

Phase (General) **Phase (Specific)**	**Creation**	**Operation**	**Change**
	or report all errors Review of unidentified errors does not result in the update of error identification routines	for data entry or is incomplete Error correction is implemented in unauthorized manner	Mistake in uploading code Uploading fraudulent code during system maintenance Adding unauthorized vendor, employee or other payee during conversion
Assignment/ update of metadata	Audit trail is defined in terms of what it should capture Audit trail is not designed to capture all required events	Incomplete audit trails Incomplete audit trails Audit trails are tampered/modified in an unauthorized manner	Audit trails are not updated to reflect changes in related system
Transmission	Transmission standards are not defined (e.g. sending sensitive material over public networks, email, ftp, etc.) Server overload Spam filter Incomplete load testing Spoofing	Delayed transmission Incomplete/lost transmission Garbled transmission Duplicated transmission Theft of data in transit	Transmission modified in an unauthorized manner

Phase (General) **Phase (Specific)**	**Creation**	**Operation**	**Change**
Maintenance and change management	Failure to define change requirements to processing phase caused by a change in input or output Failure to develop new documentation and procedures when processing changes are implemented	Failure to roll out changes in operations procedures to co-ordinate with changes in process	Failure to test changes in processing phase in a complete manner Failure to maintain routine maintenance (e.g. patch levels) to protect against malicious users and malicious code Failure to update process phase for changes in business process Failure to update the change process itself to respond to changes in the business, the organization chart, processes, technology, and personnel

Output Phase

The output phase is the culmination of the processing phase that transforms processing results into information and displays or distributes it to users of the information. The distribution or delivery phase may be an identifiable and separate sub-phase of the output phase, particularly when output is produced for storage in a database or data warehouse and users access the output through query facilities.

Output phase errors can result from the flowthrough of errors in input and processing phases leading to output use errors. In addition, output use errors can result from poor labelling of printed data or screen displays or inappropriate aggregation of data, leading to output containing too much detail or too little detail to support the purpose of the output.

Disclosure of information to unauthorized parties is often a critical error at this stage of processing (e.g. emailing a tax return to the wrong client because of an error due to the auto-complete feature). However, confidentiality and privacy related issues are outside the scope of this publication.

Type of Output

The output phase includes several types of output, including:

- Scheduled output
- Ad hoc (i.e. unscheduled) reporting (e.g., through query facilities, data mining,)
- Output to Consumer
- Output to Business
- Output to system
- Output to manual process
- Output over network

Output phase risks are described below by information integrity attribute.

Completeness

Incomplete output—Output can be incomplete even if the data entry was complete and even though processing might have been complete if the output has been tampered with and some the content has been removed or deleted.

Lost output—Output can be lost and, as a result, it may be unavailable for use in normal control procedures such as balancing, error correction and monitoring.

Currency/Timeliness

Late output—Output can be delayed due to input delays, processing delays, error correction delays, etc. As a result, it may be unavailable in the time period when it is needed to make a decision, evaluate someone's performance, or discover an error or fraud. Output delays can lead to misjudgment of the information currency, accuracy and completeness and may lead users to attempt to "adjust" for the delayed output. Such 'work around' procedures will likely be suboptimal, inefficient and/or contribute to erroneous decision making or operational responses.

Accuracy/Correctness

Flow-through of input, transmission, stored data and processing errors—Errors in input, transmission, processing or stored data will translate into output errors. In some cases, the errors may be exacerbated by poor output design that fails to enhance or even detracts from users' ability to identify such errors.

Validity/Authenticity/Authorization/Non-Repudiation

Fraudulent transmission—False information ranging from false news releases to false transactions may be transmitted and presented as legitimate enterprise information output.

Intentional Manipulation of Output—Output can be manipulated separately, apart from any possible manipulation of programs or data.

Some of the causes of output phase risks include:

- Poor output design - inappropriate levels of aggregation or disaggregation; problems with information labelling.

- Mismatch of the output and the tasks that users of the output are expected to perform with the output arising from a failure to correctly identify user information needs relating specific information requirements to specific user tasks during the information requirements analysis phase of system development.

- Failure to ensure that the output is defined to meet the requirements of downstream processes

- Unpredictable schedule of output relative to input.

- Users are not trained to interpret or use output generated by data mining tools or statistical models.

- Output routines are not updated to reflect changes in upstream processes.

- Pressure to publish or generate the output without adequately reviewing it.

Risk Magnifiers

Complexity

Complexity factors that may magnify risks in the output phase include:

- Complexity of program logic/calculations may make it difficult for users to review reasonableness of output.

- Lack of clarity/ transparency/ understandability of content increases risk of non-use or misuse.

- Large number of program-generated transactions may make verification of results difficult or excessively time consuming.

- Lack of knowledge, skills, experience or familiarity of the users may limit their capabilities to manage process.

- Employee morale and ethics may affect their attention/diligence.

- Output may reside on multiple devices or media leading to inconsistencies between different versions of the output.

- Flow-through of errors from other subsystems through interfaces may complicate review of output.

- Variety of output formats (e.g. transaction vs spreadsheets)

- Part of the audit trail may be “invisible” and may reside solely in digital form.

- Various/numerous log formats, various/numerous codes

Nature of Use

User developed application: User developed application errors in applying query languages and data mining tools generate erroneous output.

Malicious Intent

Unintentional errors that can impact the output phase include errors in handling of outputs, unsuitable level of granularity of output, and accidental deletion of output.

Output error messages can be intentionally exploited to attack the system, output can be stolen, or output can be modified in an unauthorized manner.

Link to IS Environment Domain

Flaws in the creation, operation and change of environment domain enablers such as security, availability, etc. may permit errors and abuses of output. For example, requirements definition or design errors during system development may affect output integrity. Inadequate operational procedures during system processing may affect output integrity. The change management process may not have the resources to update the output process in response to changes in the IS environment, or vice versa, leaving exploitable gaps in output.

Table 4.4 illustrates some of the risks by enabler and risk category.

Table 4.4 Examples of Information Integrity Risks by Sub-Phases of Output

		Illustrative Risks by Stage of Information System Lifecycle		
Phase (General) Phase (Specific)		**Creation**	**Operation**	**Change**
Initiation of routine	Link to Assessment of the Design of the Process	Requirements definition errors during system development may affect output integrity Design errors during system development may affect output integrity	Operations procedures during system processing may affect output integrity Errors may flow from one subsystem to another	The change management process may not have the resources to update the output process in response to changes in the IS environment, or vice versa, leaving exploitable gaps in the process
	Link to Assessment of the Environment that the Process Operates in	Omission of required processing steps may result in incomplete, delayed or inaccurate processing Flaws in design of enablers such as security, availability, etc. may permit errors and abuses of processing	Errors and delays in system operations may affect processing integrity	The process enablers in the Environment may change but this change may not be assessed for its impact on the process, and vice versa, leaving exploitable gaps in the process
Registration/ recording/ logging		Failure to define what should be logged Incomplete capture of logging events	Logs are not correctly aggregated from multiple systems Logs are not normalized to assist users in identifying patterns Tampering with logs	New systems are not integrated into logging process

		Illustrative Risks by Stage of Information System Lifecycle		
Phase (General) **Phase (Specific)**		**Creation**	**Operation**	**Change**
Check access privileges before distributing or permitting access		Sensitive content not classified Output restrictions are not defined for output devices System is not designed to limit access on a “need to know, need to do basis” Output devices do not restrict access	User access is not defined at the system resource/table/function level Users are able to access information that is unrelated to their job function	Personnel change (promotion, transfers, terminations, etc.) are not implemented in a timely manner
Output	Scheduled output, reporting, abstraction, and summarization	Metadata related to content creation, modification or use not defined or not enforced Metrics to monitor output are not defined Information integrity responsibilities are not assigned or defined for handling output Output design does not facilitate spotting errors or omissions in content Output does not display on all browsers (e.g. Firefox)	Flow through of errors from previous phases: outputs contain non-current information Delayed outputs Lost outputs Output overwritten Wrong granularity/level of aggregation Unauthorized access: theft, tampering Content tampered with by an	Output routines do not reflect system changes Job streams modified in an unauthorized manner

		Illustrative Risks by Stage of Information System Lifecycle		
Phase (General) **Phase (Specific)**		**Creation**	**Operation**	**Change**
		Information transfer delays due to poor interface design/poor integration of multiple systems incomplete outputs Information transfer duplications due to poor interface design/poor integration of multiple systems; misdirected outbound information Poor user manuals, poor training	authorized user: fraud Fail to comply with policies, statutes or regulations	
	Ad hoc reporting based on query tools	Incorrect data mapping, fields results in incorrectly defined queries Incomplete queries Incorrect queries	See above, plus: Output of the queries is unusable format Output is routed to an unauthorized user	Unauthorized modification of report Reliance on outdated, incomplete data dictionary
	Data mining outputs	Data mining or statistical model is incorrectly defined Delay in retrieval Erroneously coded logic	See above, plus: Poor user training to use data mining tools, structure data mining/statistical model, interpret output	Unauthorized modification of output
	Manual end user	Procedures to collate are not	See above, plus:	Unauthorized modification of

		Illustrative Risks by Stage of Information System Lifecycle		
Phase (General) **Phase (Specific)**		**Creation**	**Operation**	**Change**
	collation	defined Incomplete restrictions over collation process	Items collated in an inaccurate manner Incomplete collation due to missing, lost files Delayed receipt of physical receipt of materials that compose final report	output Users are not informed of system changes that impact manual end user collation
	Output to Consumer (e.g. e-commerce)	Users access agreements are defined in an incomplete manner Error correction output is displayed to the user	Web session terminates during processing Output does not display on all browsers (e.g. Firefox) Output is garbled Output captured by an authorized party Incomplete/lost transmission Slow connection; Delay in transmissions Error messages display sensitive information Transmission of confidential data without encryption (i.e.	Failure to maintain routine maintenance (e.g. patch levels) to protect against malicious users and malicious code Transmissions are modified in an unauthorized manner

		Illustrative Risks by Stage of Information System Lifecycle		
Phase (General) **Phase (Specific)**		**Creation**	**Operation**	**Change**
			transmitted in the open)	
	Output to Business	Lack of synchronization between two parties Assets commingle with other customers (i.e. should be segregated)	See above, plus: Unauthorized data is submitted through the 3rd party's system	Output to business process is not updated for system changes
	Output to system	Lack of synchronization between two systems Information transfer delays due to poor interface design/poor integration of multiple systems Incomplete outputs Information transfer duplications due to poor interface design/poor integration of multiple systems; misdirected outbound information	Flow-through of errors from previous phases: outputs contain non-current information Delayed outputs Lost outputs Output overwritten Wrong granularity/level of aggregation Unauthorized access: theft, tampering Content tampered with by an authorized user: fraud	Config/software changes are not reflected downstream Config/software changes do not reflect downstream, audit, processing Unauthorized Config/software changes

		Illustrative Risks by Stage of Information System Lifecycle		
Phase (General) **Phase (Specific)**		**Creation**	**Operation**	**Change**
	Output to manual process	Lack of synchronization between two parties Wrong granularity/level of aggregation Tampering with output	See above	Manual processes do not reflect changes made in upstream systems
	System generated output (e.g. purchase)	Lack of synchronization between two systems Information transfer delays due to poor interface design/poor integration of multiple systems Incomplete outputs Information transfer duplications due to poor interface design/poor integration of multiple systems; misdirected outbound information	Malfunction of physical device generating output Manipulation/Fraud Incorrect scheduling of output	Config/software changes are not reflected downstream Config/software changes do not reflect downstream, audit, processing Unauthorized Config/software changes
	Output over networks	Transmission standards are not defined (e.g. sending sensitive material over public networks, email, ftp, etc.) Delayed transmission	Garbled transmission Duplicated transmission Unauthorized modification of transmission Distribution to unauthorized recipients	Config/software changes are not reflected downstream Config/software changes do not reflect downstream, audit, processing Unauthorized Config/software

		Illustrative Risks by Stage of Information System Lifecycle		
Phase (General) **Phase (Specific)**		**Creation**	**Operation**	**Change**
				changes
Assignment/ update of metadata	General	Metadata not defined Sensitivity levels are not defined Failure to design metadata	Failure to assign metadata	Failure to update metadata for system changes
	Disposal	Disposal requirements are not defined for different media types and data classification Missing disposal parameters	Failure to destroy data in a timely results in unauthorized access of data Assets are released without correct disposal procedures exposing the organization to information leakage	Changes in operational, regulatory, and statutory requirements are not reflected in retention procedures
	Retention	No index or catalogue to ensure completeness Undefined retention requirements expose the organization to increased risk of unauthorized access Missing retention parameters	Disposed too soon Wrong files/information Unauthorized access: theft/fraud; hacking/ tampering; virus attack Fail to comply with policies, statutes or regulations requiring safeguarding of information	Changes in operational, regulatory, and statutory requirements are not reflected in retention procedures

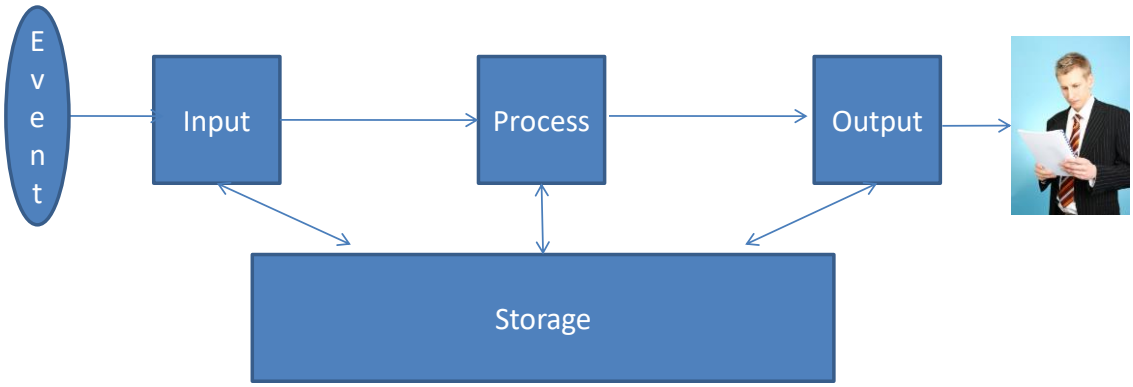
		Illustrative Risks by Stage of Information System Lifecycle		
Phase (General) **Phase (Specific)**		**Creation**	**Operation**	**Change**
	Audit trail	Audit trails are not comprehensive and do not monitor all outputs Incomplete audit trails	Incomplete audit trails Audit trails are tampered/modified in an unauthorized manner	Audit trails are tampered/modified in an unauthorized manner
Error prevention, detection and correction		Incomplete definition of error codes Errors are corrected too late System is not designed to identify all errors Review of unidentified errors does not result in the update of error identification routines	Error correction is not appropriate for data entry or is incomplete Error correction is implemented in unauthorized manner	Error codes from new systems are not integrated into the error identification process
Transmission		Failure to define output transmission restrictions Output transmission not synchronized with user needs	Garbled transmission Duplicated transmission Unauthorized modification of transmission	System changes are not reflected in transmission processes and systems
Maintenance and change		Failure to define change requirements to output use	Failure to roll out changes in operations procedures to co-	Failure to adequately test changes in output associated with

	Illustrative Risks by Stage of Information System Lifecycle		
Phase (General) **Phase (Specific)**	**Creation**	**Operation**	**Change**
management	caused by a change in output Failure to develop new documentation and procedures when output changes are implemented	ordinate with changes in output	changes in processing phase Failure to maintain routine maintenance (e.g. patch levels) to protect against malicious users and malicious code Failure to update output phase for changes in business process Failure to update the change process itself to respond to changes in the business, the organization chart, processes, technology, and personnel

Storage Phase

The storage phase is one of the endpoints of the processing phase whereby information is stored for subsequent access. The storage phase includes short term working storage, medium term storage and long-term archival storage. Thus, the storage phase may be used to hold inputs until they are processed, tables, files and databases used by application programs and backups and archives for subsequent access if required. The disposal phase may be an identifiable and separate sub-phase of the storage phase, particularly when stored information must be disposed of in a prescribed manner pursuant to security requirements or laws and regulations such as those pertaining to personal information.

Figure 4.2 Storage Phase



Storage phase errors can result from the flowthrough of errors in input, processing and output phases as well as flaws in the operation and handling of storage devices and media and unauthorized tampering with stored information.

Type of Storage

Storage can be provided in a variety of ways including:

- On line vs off line storage (including archives, desk and filing cabinets)

    - Short term storage files, tables, and databases (i.e. readily accessible including databases, ERP systems, data warehouse, datamarts)
    - Intermediate storage in various physical/logical storages (e.g. onsite tape libraries) -
    - Long term storage in data warehouse, master database –

- Structured data vs unstructured data
- Relational database vs flat file
- Disk vs tape vs USB
- Encrypted vs clear text

Storage phase risks are summarized below by information integrity attributes.

Completeness

Conversion Errors—Omissions—The initial source of stored data errors in many systems may be data conversion errors leading to omissions of data upon transfer from predecessor systems to new or updated systems.

Currency/Timeliness

Conversion errors—Cut-off—When data is being transferred from one system to another, cut-off errors can lead to synchronization errors as one process may depend on another.

Non-current files/Updating delays—Distributed systems store data in files and tables throughout the network of workstations, point of sale terminals, and so on. Keeping all the files and tables synchronized in a highly distributed environment can be a very demanding task.[9]

Non-current files/Environmental change—Even when stored data is initially correct and current, environmental conditions over time can lead to a deterioration of the quality and usability of the storage media, thereby potentially corrupting the stored data or making recovery difficult or impossible. Additionally, over time, the representational faithfulness of data declines. Conditions, situations, and reality changes with time that has a direct influence on the continued integrity of the data for use in decision making for current conditions.[10]

Accuracy/Correctness

Conversion Errors—Inaccuracies—Today's systems are constantly changing in response to requests for new features, bug fixes, new system implementations or as a result of mergers and acquisitions. Diverse systems are likely to have mismatched syntax, format or code structures. Integration of multiple systems may focus on a few key fields since comprehensive integration of the systems may be too time consuming or costly. As a result, conversion of data stored on legacy systems to new systems can yield inaccurate stored data.

Loss of Encryption Keys or Legacy Version of Software: The inadequate retention of encryption keys or legacy software can effectively render the stored data as corrupted (i.e. it exists, but is unreadable).

Validity/Authenticity/Authorization/Non-Repudiation

Conversion errors—Invalid/fraudulent data—A potential source of invalid data is duplicate processing of data at the point of conversion. Conversions also represent opportunities for invalid

---

[9] For example, checkout overcharges by US retail chains using bar-code scanners (presumably due to delays in updating stored price tables to reflect promotional campaigns) may amount to US $2.5 billion annually. Several US retail chains have been fined for incorrect pricing. (Bartholomew, D.; 1992)

[10] For example, customers die, divorce, age, marry, and move. It has been estimated that the integrity of stored data declines at a rate of 2 percent per month. (Eckerson, W. W.; op. cit.)

data to be created by a fraudster. In particular, introduction of unauthorized master file records during a conversion can later lead to unauthorized disbursements.

Backup and recovery—Errors in backup and recovery procedures may cause files to be restored more than once, leading to duplication of data.

Error correction—Delays in data processing can lead to the creation of duplicate data by entity personnel seeking to rectify what appear to be errors but what are, in fact, simply delays in processing information.

Inconsistencies due to data redundancy—Intentional duplication of data records is common in many systems. For example, sales data may be stored in the sales sub-system, the costing sub-system, the general sub-ledger system, the commission payroll sub-system, the customer history and analysis sub-system, and so on. In addition, distributed systems commonly replicate some or all of the data at various sites for the sake of operational efficiency. Inconsistencies can arise in the different versions of the same records as some data may be updated at a different time than other data. Users may not be able to determine which data is the most current data and which is the "old" data resulting in reliance on inappropriate information.

Uncontrolled use of data fields—Other changes affecting stored date include unexpected changes such as adding new fields, new code values and reuse of fields by users adapting existing fields for different purposes.

Some of the causes of storage phase risks include:

- Information retention and/or disposal actions that are not consistent with user requirements.
- Flaws in backup and recovery procedures can result in operator or software errors that lead to errors in stored data. Examples of errors in backup and recovery procedures include:
    - Making a backup of the wrong version of a file
    - Making incomplete backups, then using them restore files

    - Restoring files more than once, leading to duplication of data.
    - Restoring the wrong version of a backup file.

- Software Errors—Faults in programs, database management systems and other systems software can cause errors in stored data.
- Inadequate documentation/training prevents the correct retrieval of information
- Changes in operational, statutory, and regulatory environments are not translated into changes in storage specifications.

Risk Magnifiers

Complexity

Complexity factors that may magnify risks in the storage phase include:

- Balancing retention and disposal requirements with risk of unauthorized access
- Managing various formats, obsolete data formats/applications/operating systems, encryption keys (i.e. to decrypt data)
- Managing large volumes of data, geographically dispersed systems, supporting different releases
- Need to meet non-uniform/ contradictory operational, regulatory, and statutory archiving requirements
- Managing storage classification (short term, intermediate, long-term)
- Numerous business segments and statutory jurisdictions make it difficult to establish comprehensive retention and disposal requirements

Malicious Intent

Unintentional errors in the storage phase of processing can result if there is a lack of adequate storage retrieval procedures, stored information is not corrected for errors identified in the production environment, and backups are not tested in a “disaster” scenario to identify issues with recovery process.

Stored information can be intentionally accessed and tampered with by unauthorized users if it is not destroyed in a timely manner or if access controls are not operating properly.

Link to IS Environment

Flaws in the creation, operation and change of enablers such as security, availability, etc. may permit errors and abuses of stored information. For example, omission of required processing steps may result in incomplete, delayed or inaccurate information being stored. Errors and delays in system operations may affect the integrity of stored information. The change management process may not have the resources to update the storage process in response to changes in the IS environment, or vice versa, leaving exploitable gaps in the storage process

Table 4.5 illustrates some of the risks by enabler and risk category.

Table 4.5 Examples of Information Integrity Risks by Sub-Phases of Storage

		Illustrative Risks by Stage of Information System Lifecycle		
Phase (General) Phase (Specific)		**Creation**	**Operation**	**Change**
Initiation of routine	Link to Assessment of the Design of the Process	Online, offline, archive storage requirements not defined Recovery and restoration requirements not defined completely and accurately Data retirement and destruction requirements not defined accurately and completely Poor design during system development may affect processing integrity	Operations procedures during system processing may affect processing integrity Errors may flow from one subsystem to another	Storage requirements and processes may not be appropriately amended to reflect changes to modifications to information processes, business requirements or regulatory requirements
	Link to Assessment of the Environment that the Process Operates in	Omission of required processing steps may result in incomplete, delayed or inaccurate processing Flaws in design of enablers such as security, availability, etc. may permit errors and abuses of processing	Errors and delays in system operations may affect processing integrity	The process enablers in the Environment may change but this change may not be assessed for its impact on the process, and vice versa, leaving exploitable gaps in the process
	Note: Link to Availability/Accessibility Operations – Data	Retention requirements are not defined Storage process is not designed to enable retrieval of data in a	Unavailable (system down) Inaccessible (data not in path)	Non-current files

		Illustrative Risks by Stage of Information System Lifecycle		
Phase (General) Phase (Specific)		**Creation**	**Operation**	**Change**
	management	manner that is easy for users		
Registration/ recording/ logging		Failure to define what should be registered/ recorded/ logged Incomplete implementation of registration/ recording/ logging capability	Logs are not correctly aggregated from multiple systems Logs are not normalized to assist users in identifying patterns Tampering with logs	New systems are not integrated into logging process
Match access/ privilege		User access is not defined at the system resource/table/function level Users access agreements are incorrectly/incompletely defined System is not designed to limit access on a “need to do basis”	Unauthorized access: ~~theft~~/fraud; ~~hacking~~/tampering; virus attack Fail to comply with policies, statutes or regulations requiring safeguarding of information	Personnel change (promotion, transfers, terminations, etc.) are not implemented in a timely manner
Storage Types	Short term storage files, tables, and databases (i.e. readily accessible including databases, ERP systems, data warehouse, datamarts)	Retention requirements do not account for specific requirements of the application (e.g. ERP) System was implemented without consideration of storage requirements	Incorrect files/information Incomplete logging	Upgrades/ system changes to ERP undermine storage procedures Storage procedures are not modified to match system changes or upgrades Documentation is not maintained or updated when processing

		Illustrative Risks by Stage of Information System Lifecycle		
Phase (General) **Phase (Specific)**		**Creation**	**Operation**	**Change**
				changes are implemented.
	Desk and filing cabinets	Backup strategy is not defined Sensitivity levels are not defined Physical layout does not lend itself to filtering access to desk and filing cabinets	See above, plus: Lost files, Delayed filing physical loss of cabinets Incorrect files Knowledge of (physical) file management is not retained	Personnel changes do not result in changes to access management (e.g. locks to filing cabinets, electronic access codes, etc.)
	Archives	Archiving strategy is not defined; no standard on how to deal with unstructured data versus structured data Metrics to monitor storage/ archival are not defined Information integrity responsibilities are not assigned or defined for storage personnel Archiving strategy is designed independent of retention requirements	See above, plus: Incorrect files/ information Duplicated processing Omission or loss of information due to archiving failures Poor training	Changes in operational, regulatory and statutory environments are not reflected in the archival process

		Illustrative Risks by Stage of Information System Lifecycle		
Phase (General) **Phase (Specific)**		**Creation**	**Operation**	**Change**
		Archival strategy is designed for mainframe (i.e. centralized) processing environment and does not cater to a distributed decentralized) processing environment Poor user manuals, poor training		
	Structured data vs unstructured data	Structured data (see above) Unstructured data is inaccessible due to inadequate design of search features of storage software Archival strategy is designed for structured data and not unstructured data Generation of unstructured content is not subjected to same control as structured content	Structured data (see above) Users are not trained to handle unstructured data Knowledge related to management of unstructured data is not retained Insufficient access control procedures are in place over unstructured data Lack of version control over unstructured data	Structured data (see above) Change management procedures do not anticipate impact on unstructured data
	Disk vs tape vs USB	Media characteristics not suitable or appropriate for the intended use or user. For example: • Media (e.g. CD, tape, etc.) is not suited for	Content is inaccessible due to media's susceptibility to environmental risks (magnetic waves, humidity, etc.) Where media is re-used for	Media fails to capture content due to changes in upstream job scheduling/ processing Changes in specifications, format or structure of content, processes

		Illustrative Risks by Stage of Information System Lifecycle		
Phase (General) **Phase (Specific)**		**Creation**	**Operation**	**Change**
		rate/volume of transactions generated • Media is not suited to capture transactions in real time • Media lacks access management features to prevent unauthorized changes to data Lack of clarity or consensus in the definition of media requirements Unusability of media for content format Labelling or version identification standards are poorly designed Job scheduling and processing logic that record data on media are poorly design Inappropriate media design for presentation of content	backup purposes, incorrect “unit” (e.g. tape) is erased Content cannot be accessed due to poor labelling/inability to identify media Inappropriate media used for presentation of content Media/device is lost due to poor library controls over use of device/media	or system environment not reflected in design of media
	Encrypted vs clear text	Clear text (see above) Onerous processing requirements	Clear text (see above) Poor cryptographic key	Clear text (see above) Management does not maintain

		Illustrative Risks by Stage of Information System Lifecycle		
Phase (General) **Phase (Specific)**		**Creation**	**Operation**	**Change**
		of encrypted data is not integrated into system design specifications; making the retrieval of the data difficult (i.e. extremely slow) or effectively inaccessible	management controls prevent the timely retrieval of encrypted data Encrypted information is not scanned for malicious code upon retrieval	older cryptographic keys, when it updates to the new cryptographic algorithm; making the data encrypted by the old algorithm inaccessible
Error prevention, detection and correction		Inadequate definition of error codes Error correction process does not require the correction of stored errors	Error correction is not complete, accurate or timely	
Assignment/ update of metadata	Disposal	Disposal requirements are not defined for different media types and data classification Incomplete disposal parameters	Failure to destroy data in a timely results in unauthorized access of data Assets are released without correct disposal exposing the organization to information leakage	Changes in operational, regulatory, and statutory requirements are not reflected in retention procedures
	Retention	Undefined retention requirements expose the organization to increased risk of unauthorized access Incomplete retention parameters	Incorrect meta-data, which results in the following: • Disposed too soon • Incorrect files/information	Changes in operational, regulatory, and statutory requirements are not reflected in retention procedures

		Illustrative Risks by Stage of Information System Lifecycle		
Phase (General) **Phase (Specific)**		**Creation**	**Operation**	**Change**
			• Unauthorized access: theft/fraud; hacking/ tampering; virus attack • Fail to comply with policies, statutes or regulations requiring safeguarding of information	
	Audit trail	Incomplete audit trails	Incorrect audit trails	Audit trails are tampered with/modified in an unauthorized manner
Backup recovery, including offsite storage		Backup processes are not defined and reconciled with retention and archival processes Delayed recovery of information	See above, plus: Omission or loss of information due to backup/recovery failures Backups are not tested in a "disaster" setting to identify issues with recovery process	System changes are not reflected in backup process
Maintenance and change management		Failure to define change requirements to storage caused by a change in other phases, business or environment (e.g., regulations) Failure to adopt new storage	Failure to roll out changes in backup and recovery procedures to co-ordinate with changes in processing	Failure to adequately test changes in backup and recovery procedures associated with changes in processing and output phases Failure to maintain routine

	Illustrative Risks by Stage of Information System Lifecycle		
Phase (General) **Phase (Specific)**	**Creation**	**Operation**	**Change**
	locations when business changes such as mergers or acquisitions take place		maintenance (e.g. patch levels) to protect against malicious users and malicious code Failure to update storage phase for changes in business process Failure to update the change process itself to respond to changes in the business, the organization chart, processes, technology, and personnel
Disposal	Failure to define disposal requirements Failure to design effective disposal procedures	Failure to execute disposal procedures in accordance with policies	Failure to update disposal requirements for changes in requirements, policies, statutes or regulations

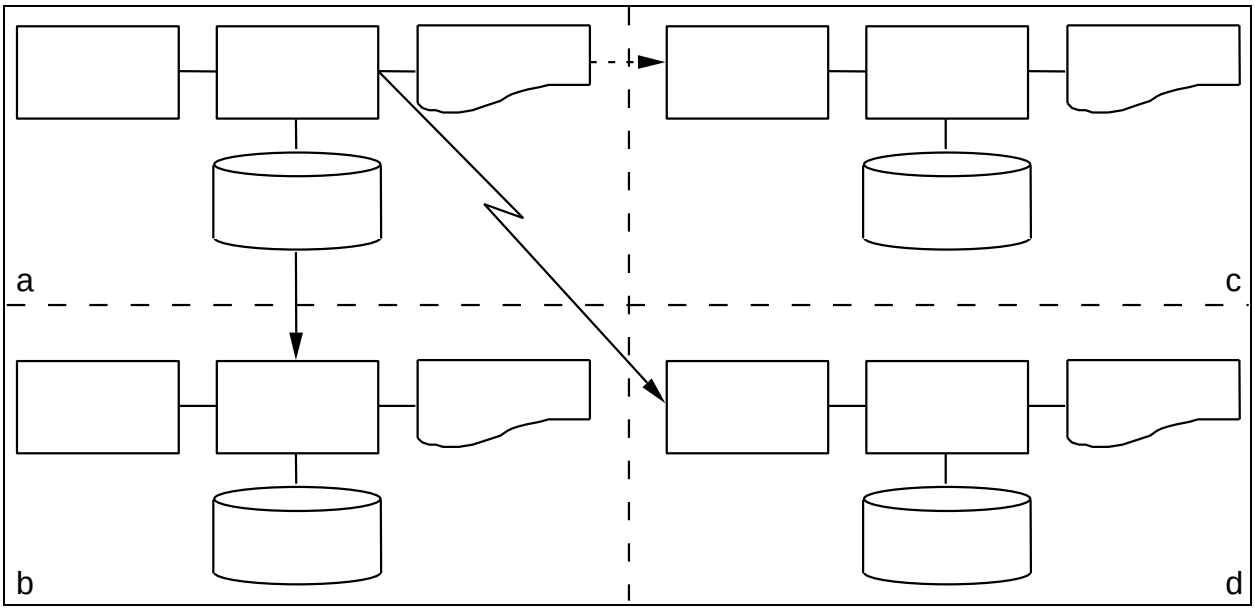
Integration of Processes and Systems Through Various Channels and Interfaces

Most systems consist of several connected processes that are linked through a variety of channels and interfaces. Interfaces between processes and between systems provide potential openings for information integrity impairments to flow from a corrupted process or system to one that was previously pristine. Systems and sub-systems can be integrated in ways that are not always obvious. For example, systems can be integrated when the output of one system is the input of another. Such input/output transfers can be electronic or involve a variety of media such as paper, disks, tapes, flash drives and other media. Systems can also be integrated by sharing a common database. For example, Figure 4.3 illustrates processes connected through physical flows of data on various media (e.g. paper document is received, entered into the system, and flows from panel a to panel c), through electronic transmissions over a network (e.g., electronic message flow from panel a to panel d) and through sharing of data in a data base on a storage device (e.g., data flow from panel a to panel b). These interfaces between systems are prone to generating errors and disseminating them throughout the system.

The challenge also exists between an organization and external providers, such as cloud computing companies, that connect into the organization's system through Application Programming Interfaces (APIs).

Lack of referential integrity checks (e.g. automated reconciliation of hash totals) as data moves between systems and sub-systems can lead to the proliferation of errors far beyond the original data/information source. The problem may be exacerbated in organizations that employ silos without end-to-end monitoring of cross-silo processes. That is, each silo may understand its own contribution to the process, but not how any other silo uses or transforms the information that it processes.

Figure 4.3 Integration of Processes and Systems through Various Channels and Interfaces



Interfaces and Complexity

Interfaces can create complexity because systems and sub-systems can be integrated in a variety of ways that are not always obvious. Lack of data standardization between subsystems and excessive variety and redundancy of transforming interfaces can cause information integrity impairments. For example, moving information from operational systems to data warehouses can result in errors when changes in the data structure in the operational system are not mapped to the data structure of the data warehouse. Interface data integrity controls are particularly important when moving financial data from one system to another. For example, the secondary system should reconcile the sum of the transactions submitted with the control total on the trailer record (and if this fails the systems should cease processing the file and send an error message to the user). The lack of such data integrity controls would carry errors throughout the system.

Synchronization errors are problematic when multiple systems maintain the same data throughout the organization. For example, if a customer's information is changed on system A, but not system B – there will be a synchronization issue and it will not be clear which instance is the most current.

Cultural differences and the use of different languages in multinational organizations can create different terminology or different understandings of a common terminology. This can be a problem

for a variety of data types, particularly currency data. For example, a Canadian subsidiary transmits its information in Canadian dollars, but the American parent consolidates the information as if it was denominated in US dollars.

Upstream and downstream data sharing by multiple units in large organizations can exacerbate problems. Each unit may understand its own contribution to or use of information or a process, but not how any other units use or transform the information that it receives or passes on from/to other units.

Organizations that subscribe to cloud computing services can also experience such issues, when the provider updates their service. This is especially a challenge when the organization had a difficult time integrating the cloud services into their environment.