CHAPTER 10

Privacy Engineering and Quality Assurance

If you don't have time to do it right, you must have time to do it over.

—Unknown

This chapter will look at best practices for managing privacy issues within the process of quality assurance (QA) for developing and deploying products, systems, processes, and applications that involve personal information. Quality assurance is done continuously throughout the development process.

Privacy impact assessments (PIAs) will be presented as a tool for both identifying where privacy controls and measures are needed and for confirming they are in place. The benefit of a PIA for the many stakeholders in protecting personal information will also be discussed.

Quality Assurance

Similar to the creation of privacy policies, there already exists a fairly extensive body of literature regarding QA as a discipline, a process, and an art form. So this book will not go into extensive detail on the concept of QA other than to say that it is the planned and systematic set of activities in the development process of a product or service ensuring that quality requirements are consistently met. In practice, QA is making sure that what is produced works how it was designed to work and whether it meets an enterprise's requirements. The privacy development structure for QA is presented in Figure 10-1.

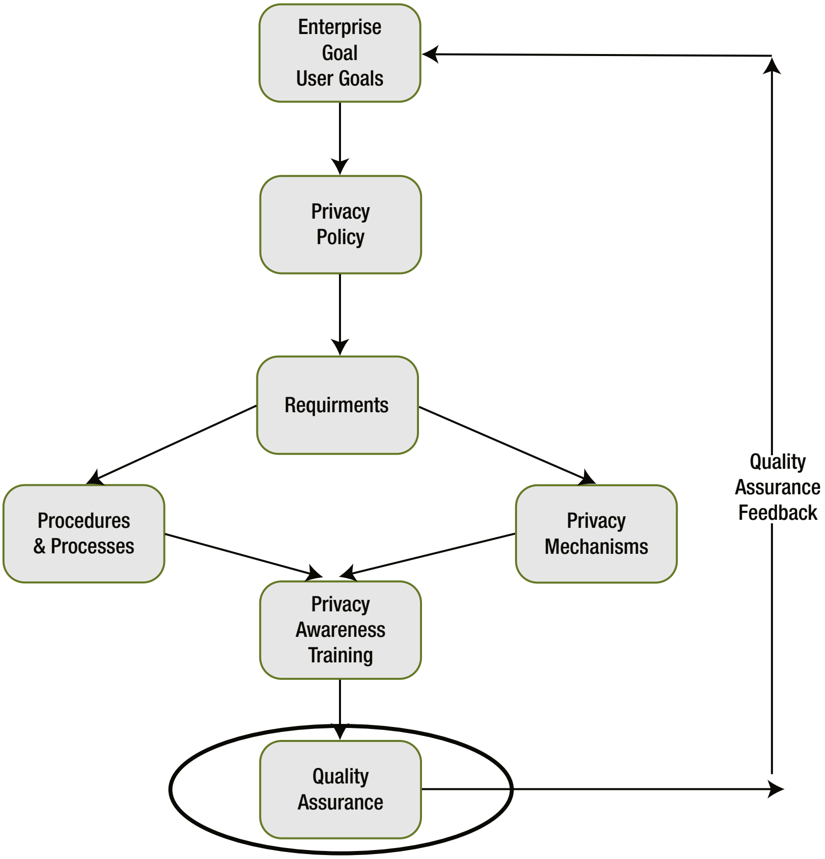

Figure 10-1. *Privacy development structure for quality assurance*

In many engineering programs, these underlying QA activities are part of each phase of the development lifecycle with a final QA check, including testing, as the last phase of development before deployment or release (Figure 10-2).

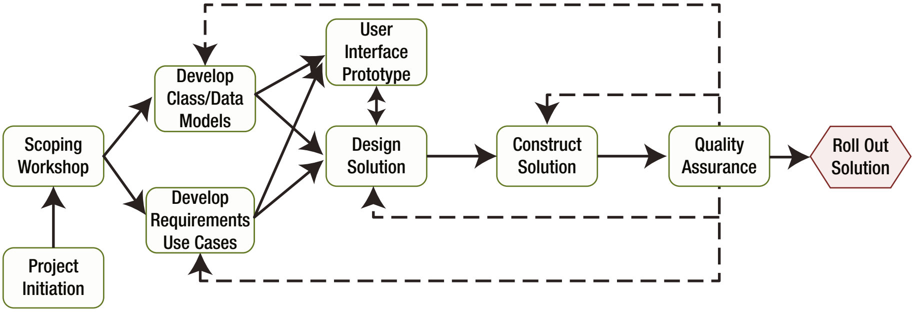

Figure 10-2. *Systems' engineering Lifecycle*

Using Frameworks to Create a Privacy Quality Assurance Checklist

As in any other type of change management, using existing frameworks and standards hastens adoption of the new desired state. They provide a home base for known ways of doing things and thinking about things where appropriate and open the possibility to tackle positive change. The most well-known and accepted frameworks for fair information governance are understood by the data privacy community, but they may be less known or understood by the technical or management actors in an enterprise. When these time-tested governance principles are leveraged to create design and feature development requirements,[1] they are joined to well-known and time-tested basic technical practices and a new but grounded framework emerges.

In each step in the system engineering lifecycle (Figure 10-2), a privacy engineering QA checklist, like the one outlined in the following sections, should be referred to.

Purpose

While answering the following questions, the use case and data model, including the metadata, should be considered:

- Are the purposes of this project (and uses of personal information) clearly defined? Are they legitimate and known to the user?
- Does each data element and attribute, related to personal information, have a direct relationship to the purpose for which it is collected and processed?
- What privacy rules are needed to ensure that the purpose principle is satisfied?
- Are there metadata that support the purpose principle?
- Is there a chance that a data subject, whether an individual or an enterprise, would be embarrassed or damaged by the processing or publication of the personal information?
- Should the data be segmented?
- Are the types of information allowed to be collected limited?

[1] The concept of policy that is created and leveraged for systems and governance requirements is covered in Chapter 4.

Notice

While answering the following questions, the use case and data model, including the metadata, should be considered:

- Does the requirements statement define a complete notice that satisfies the notice principle?
- Does the notice accurately describe the processing?
- Is the notice(s) presented to the user in a timely manner?
- Are there statutory or common law requirements concerning notice in all jurisdictions wherever the system impacts?
- Is the notice clear, consistent, and relevant to the intended reader?
- Does the technique used to meet the notice requirement encourage review and facilitate understanding? For instance, would animation or a pop-up video make the notice more appealing and clearer?
- Is the notice context based or discoverable?

Choice or Consent

While answering the following questions, the use case and data model, including the metadata, should be considered:

- Are choices clearly shown to the user throughout the design?
- Does expressing choice require action by the user? Can choices be missed or easily overlooked?
- Are defaults explained clearly? Do they put privacy at risk?
- Are defaults set to either lessen the sharing of PI or so clearly tied to the notice and the context that the only reasonable expectation for a user would be that the information is shared?
- Are tools used so that choices made by the data subject may be recorded, audited, and corrected along the way?

Transfer

While answering the following questions, refer to the use case, data model, including metadata, and database design:

- Is data transferred to and from a third-party protected by contract, administrative, technical, logical, and physical means?
- Does the transfer of data from or to different geographic areas, such as member-states of the European Union, require a legal mechanism (such as Safe Harbor Certification or Model Contracts) to make the transfer legitimate?

- Are the proper procedures in place for all types of third-party transfers and all impacted jurisdiction?
- Are encryption and obfuscation techniques used both appropriately and effectively?

Access, Correction, or Deletion

While answering the following questions, refer to the use case, data model, including metadata, and database design:

- Has the requestor been authenticated?
- Is the segmented appropriately so that different segments can be handled with different privacy or security rules?
- Can roles be defined so that privacy risks can be managed by means of privacy rules?
- Are rules concerning correction and deletion in compliance with the laws or regulations of all jurisdictions impacted by the system or process or by the enterprise policies?

Security

While answering the following questions, check the use case, data model, including metadata, and design documentation:

- Has the data been classified so appropriate controls can be determined?
- Are ISO and other standards for information and security leveraged to ensure the necessary confidentiality, integrity, and availability of the data?
- Are the information security teams within your enterprise included on the project team?
- Are the security rules (including encryption) defined for each data attribute?
- Are security rules covered for data transfers, especially across jurisdictional lines?

Minimization

While answering the following questions, check the use case, data model, including metadata, and design documentation:

- Is each personal information data attribute being collected needed for the solution being designed or is it being collected "just in case"?
- If data is being collected for potential big-data purposes, can big-data analysis be used to identify a person, thus raising a potential privacy issue?

Proportionality

While answering the following questions, check the use case, data model, including metadata, and design documentation:

- Is the data being processed proportional to the purpose of the processing?
- Are risk and value balanced? Is the risk to an individual's privacy outweighed by the benefit (to the individual or society at large) and if not, what are the compensating controls?

Retention

While answering the following questions, check the use case, data model, including metadata, and design documentation:

- Are archiving rules for each data attribute well established?
- Have data destruction tactics such as degaussing or permanently encrypting and destroying keys or overwriting the data after a specific deadline been adequately considered?

Act Responsibly

- Is the privacy team included on the project team? Has a data governance or data stewardship program that include privacy been established?

This checklist is comprehensive and can be used throughout the system development process.

Privacy Concerns During Quality Assurance

At a conceptual level, QA for privacy-engineered products, systems, processes, and applications is no different from other engineered products especially since the privacy requirements should have been factored into the design and the development from the early stages of planning. What needs to be emphasized, however, is the operational level, which has three vectors:

- The first vector concerns making sure the very act of QA doesn't create privacy issues.
- The second vector is the use of the privacy impact assessment (PIA) tool to determine whether the processing of PI in a given situation meets (or surpasses) an enterprise's privacy requirements and hence its quality requirement.
- The third vector is the importance and value a PIA has for a variety of stakeholders from internal and external regulators to the wide range of roles associated with developing products, systems, and processes that use personal information.

Vector 1: Managing Privacy During Quality Assurance

To ensure a product, system, or process works, it needs to be tested and the results examined, diagnosed, reported, and shared. For products, systems, or process that use personal information, this presents a potential privacy conundrum: How do you test that the proper thing is happening without unnecessarily or improperly exposing the underlying data?

Best practice is to conduct QA of a system, product, service, or process that involves personal information with fake or dummy data. This data can be made up whole cloth or at least suitably deidentified from a real dataset. The reason for this approach is threefold:

- First, during system testing data often gets manipulated and changed. You don't ever want changed data corrupting production or live data should it ever migrate back into the live system by accident before deployment (e.g., in the case of system or process upgrade or migration). Also, you don't want to create an incident or breach due to real data not being properly deleted and later being "found."
- Second, data is provided for specific purposes and are supposed to only be used for such purposes. Therefore, it is not proper for data from one system or process to be used to test or model another system without permission from the owner of the data.
- Third, using real data for testing may expose it to people who, under normal circumstances, would not have had access or reason to see the data. Although the type of data may not necessarily be the kind contemplated by data breach notification legislation and regulation, this may be considered unauthorized use and access may be a violation of most organization's privacy policies.

Unfortunately, this practice is increasingly difficult to regulate with mobile apps and large, complex enterprise applications and systems, especially when it comes to replicating errors or testing new functionality.

Therefore, there are some steps that can be taken to manage privacy during the QA of systems, products, processes, or applications that contain "live" personal information. Below are some things to consider:

1. Test in read-only mode.
2. Test in a secure environment with tightly controlled access.
3. If testing requires any manipulation of data, ensure the test file is destroyed at the end of testing.
4. Mask data whenever possible.[2]
5. Do not give testers greater access than they would have as system users under normal circumstances.
6. Perform a PIA on the testing environment and QA test plan (more about PIAs below).

PRIVACY CAN BE A COMPONENT OF DATA QUALITY

Data quality has been defined as creating and maintaining data that consistently meets knowledge worker and end-customer expectations.

To implement data quality, an enterprise needs to develop a data quality strategy, and to develop this, in conjunction with the data stewardship working team, the project must devise:

- An enterprise-wide data quality policy and procedures regarding the move to production activities
- A data governance charter as a part of data governance
- Data quality controls
- Data quality reviews and sourcing analysis methodology as a part of the architectural reviews during development
- Enterprise standards on unique identifiers and reference attributes
- Error logging and tracking
- An integration plan with the metadata strategy

[2] Masking data is hiding or deidentifying actual data to protect the actual data while having a functional substitute for occasions (like testing or prototyping) when the real data is not required.

The controls and measures, policies, standards, and guidelines listed above support and align with the goals of data privacy provided they factor in privacy requirements.

From Table 10-1, it is easy to see how the characteristics of data quality and the associated benefits align with privacy. However, many data quality programs do not factor in privacy. If an organization has a data quality program, it does not necessarily mean that it has factored in privacy. However, an enterprise cannot consider its data of highest level of data quality unless privacy concerns are fully addressed.

Table 10-1. *Benefits Related to Data Quality Characteristics*

Quality Characteristic	Benefit
The right information	Timely information from the right source
With the right completeness	All the information I need
In the right context	Whose meaning I understand
With the right accuracy	I can trust and rely on it
In the right format	I can use it easily
At the right time	When I need it
At the right place	Where I need it
For the right purpose	I can apply it

Vector 2: Privacy Impact Assessment: A Validation Tool

The second vector of QA for privacy engineering is to ensure the necessary privacy controls and measures are in place by using the PIA tool. The PIA tool can be used during the design and development phases of a project to determine which controls and measures are needed. It can also be used to validate that the prescribed controls and measures are in place or that suitable alternative risk management activities have been implemented.

The PIA provides a living document that becomes the "system of record" for how, within a given activity, personal information is collected and managed; where risks exist to the data, the enterprise, and the people impacted; and which controls and measures are used to mitigate the risks and legitimize the processing of the personal information. It is an interactive process that looks at business, operational, and technical issues.

The PIA is a five-phase process, as shown in Figure 10-3.

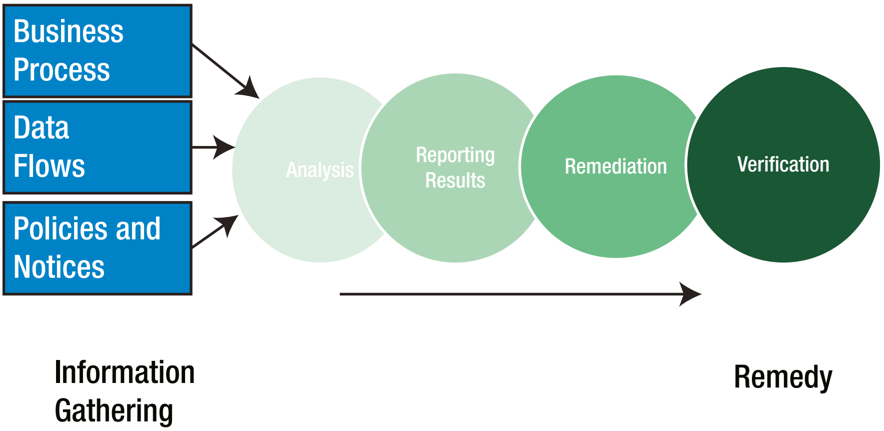

Figure 10-3. *The privacy impact assessment has five phases*

- *Phase 1:* Information gathering—Business and technical stakeholders will be interviewed; appropriate use cases and class and data models will be reviewed; and privacy policies, procedures, standards, guidelines, and best practices will be assessed.
- *Phase 2:* Analysis of the information gathered—The privacy team will analyze the information gathered.
- *Phase 3:* Reporting results—The PIA report will be developed containing controls and processes currently in place, an identification of the gaps between the current state and desired state, controls and mitigations where needed, and recommendations.
- *Phase 4:* Remediation needs are determined.
- *Phase 5:* Verification that privacy requirements have been met.

As data flows and usage of controls and measures change as products, systems, processes, and applications evolve, information will need to be regathered, reanalyzed, rereported, and possibly reremediated and definitely reconfirmed. The PIA process does not end until the data is disposed of or deleted.

In addition to being interactive, PIAs are iterative. Until the development stage (and even then) not all the controls and measures are always known or fixed. It is the same with usage of data. The development and functional specifications process, especially in the age of Agile development, can be quite fluid. Depending on where in the development cycle the PIA is being conducted, the PIA can serve as a tool to indicate what is needed or a tool to confirm what is in place or planned.

10 BEST PRACTICES FOR CONDUCTING PRIVACY IMPACT ASSESSMENTS

By Denise Schoeneich, CIPM, CIPP/IT, CISA, PMP, IT Risk, Compliance, and Audit Professional at Resources Global Professionals (RGP)

1. *Craft an elevator pitch:* Do not assume that everyone has the same level of understanding of the definition for personal information and the objectives and requirements for a PIA.
2. *Request a demonstration:* A demonstration of the process, product, application, or technology (system) will help provide an understanding of how personal information is processed.
3. *Engage the right people:* Identify the system's business and technical owners responsible for each process related to the flow of personal information. Documenting a PIA is usually a progression; be prepared to contact additional system owners as the processing of personal information becomes clearer. Have another member of the privacy team review the PIA through fresh eyes to identify privacy impacts not previously recognized.
4. *Conduct PIAs in real time:* Guide system owners through the steps of completing a PIA by scheduling working sessions; completing a PIA can be daunting the first time. Blocking out time to walk through the PIA rather than waiting for the system owners to respond is a good way to ensure timely completion.
5. *Describe the "big picture":* In the description, broadly describe the system and include whether any personal information will be processed. The description should identify any links with other systems or processes.
6. *Right size the PIA:* No one size fits all; the PIA effort should be commensurate with the complexity of the system and the level of privacy risk identified. For complex systems, separate PIAs with detailed process narratives and flowcharts should be created for each major component. The PIA documentation should be brief for a system with minimal privacy implications.

7. *Document the system's personal information flows:* A process narrative is a high-level description of the system's personal information process flows and identifies how the processing of personal information complies with GAPP. By documenting and understanding the personal information flows, the controls, or absence of controls, will stand out.
8. *Map the data flow of personal information:* Illustrating the data flows using diagrams that identify all key processes in the information's lifecycle can provide a clear picture that pinpoints where information is collected, used, stored, transferred, and retained and visually depict the risks, controls, and gaps.
9. *Be aware of scope:* Consider all the uses of personal information including those that may be expected but are uncommon such as administrative use of data and customer and technical support.
10. *Trust but verify:* Obtain and review database schemas, integration documentation, system guides, and architectural diagrams to confirm the accuracy of information provided by the system's owners.

Who Is Usually Involved in a PIA?

The roles involved in a PIA vary from organization to organization. Therefore, it is better to discuss the functions or areas of activity that are usually involved:

- Business
- System development
- Engineering
- User experience representatives
- Data governance
- Legal
- Privacy team members

Why so many? The short answer is that a PIA looks at the entire lifecycle of the personal information in a system, product, process, or application. Rare is the case in which one or two individuals have a sufficient functional or operational understanding to perform the PIA. Just as it takes a village to raise a child, it takes a team to design, develop, and launch a product, system, process, or application.

PRIVACY ENGINEERING REQUIRES BOTH QUALITY AND SECURE CODE (PART 1)

By James Ransome, PhD, CISSP, CISM, Senior Director, Product Security at McAfee

Quality and Secure Code

Privacy engineering requires both quality and secure code, but quality and secure code need to be understood and work together. I will start by defining what quality software is and then move on to the differences and synergies and differences between quality and secure code and then the importance of privacy in the security development lifecycle (SDL).

Quality Software

Software quality refers to two related but distinct concepts:

1. How well its functional aspects comply with or conform to a given design, based on functional requirements or specifications.
2. How well the structural aspects comply with the nonfunctional requirements that support the delivery of the functional requirements, such as robustness or maintainability.

The structure, classification, and terminology of attributes and metrics applicable to software quality management are typically derived or extracted from ISO/IEC 25010:2011—*Systems and software engineering—Systems and software Quality Requirements and Evaluation* (SQuaRE)—*System and software quality models.*[3] The initial Consortium for IT Software Quality (CISQ) version of the CISQ Software Quality specification was first published in 2012. The software quality characteristics included in this specification were selected in the CISQ Executive Workshops held in Washington, D.C.; Frankfurt, Germany; and Bangalore, India. These quality characteristics include:

- Reliability
- Performance efficiency
- Security
- Maintainability

[3] www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=35733

The current version of the CISQ quality standard is version 2.1.[4] This specification is currently being prepared in the formats required by the Object Management Goup (OMG) and will be submitted into the OMG approval process in early 2014. When finalized, OMG will submit these specifications through their fast-track process to ISO.

I believe one of the most relevant descriptions of software quality for this article is that provided in *Juran's Quality Control Handbook*:

> The word quality has multiple meanings. Two of these meanings dominate the use of the word: 1. Quality consists of those product features which meet the need of customers and thereby provide product satisfaction. 2. Quality consists of freedom from deficiencies. Nevertheless, in a handbook such as this it is convenient to standardize on a short definition of the word quality as "fitness for use."[5]

In general, producing quality software is the degree to which software meets its specifications and satisfies its intended purpose and that the customer is satisfied with the product. It is generally accepted that the customer is satisfied with the quality of the software when they believe the product has delivered exactly what was promised, their product experience does not result in any negative consequences, and they believe the product meets or exceeds their expectations.

Many software quality practitioners describe quality as the elements that can be built into the software development process. If this is a reflection of customer needs and expectations, then the software can be deemed good quality. It is important to meet the needs and expectations of the customer. In order to do so, the elements of software quality must be built into your software. Elements of quality include:

- Capability
- Flexibility
- Maintainability
- Portability
- Readability
- Reliability
- Reusability
- Testability
- Understandability
- Usability

[4] This can be found at: http://it-cisq.org/wp-content/uploads/2012/09/CISQ-Specification-for-Automated-Quality-Characteristic-Measures.pdf
[5] J. M. Juran, *Juran's quality control handbook*. New York: McGraw-Hill, 1988.

The software developer has not completed the process of developing a software program of good quality until the customer has declared satisfaction with the product delivered. Although much of the quality is focused on end-user requirements, it also includes nonfunctional and system function requirements.

Ultimately, security, privacy, and reliability issues are quality bugs. The relationship between security, privacy, and reliability as elements of quality can overlap. For example:

- Security mechanisms can be used to mitigate privacy concerns.
- A security issue can result in a reliability issue. Security bugs that lead to reliability issues could mean reduced uptime and failure to meet service-level agreements, and security bugs that lead to disclosure of sensitive, confidential, or personally identifiable information are privacy issues and can have legal ramifications.
- Reliability and security issues can result in a failure of the software to protect PI, which in turn becomes a privacy issue.

Reliability, security, and privacy can also be independent of quality and fall out of that overlapping relationship. For example:

- An employee may neglect to shred paper print-out copies of a database containing PI in a software program and it is found by a cybercriminal in the local dumpster; this is not a security/privacy–quality issue but rather a security/privacy issue outside the purview of quality software.
- A power outage may occur that results in downtime of a software product because the affected machine doesn't have a UPS; this is not a software reliability/quality issue but rather an operational reliability issue unrelated to the design of the software.

Overall, security and privacy should not be considered separate tasks but approached in a holistic sense intersected with reliability and quality. To be effective, the principles of quality must be ingrained in the software developer's mindset so that it becomes second nature and part of the process by which they correctly develop code on a daily basis. As you will see later in this chapter, the attribute of quality also includes security and privacy. Although all three may be dealt with separately in the development process, they must be dealt with in a coordinated fashion with equal importance.

One of the key challenges in producing quality software is the desire to keep costs down and meet aggressive schedules which exacerbate the inconsistency in the application of quality requirements in the software development lifecycle, even for mission-critical and human-life dependent systems. The speed of delivery required by Agile development processes has made this even more challenging. Another key challenge is that the practice of software quality is still an art form, and it is costly and hard to find those who are talented with an ability to create software that can meet the ever-changing challenges we face in today's cyber environments.[6]

What Should a Privacy Impact Assessment Document Contain?

The PIA report acts as a record of compliance for OECD Guidelines, GAPP, or other regulatory or corporate privacy requirements as reflected in the privacy policies. A PIA acts as a tool to surface risk and drive risk acceptance or mitigation. Specifically, the PIA report will contain:

- A baseline of controls and processes currently in place
- Identification of the gaps between the current state and desired state
- The framework for implementing controls and mitigations where needed

To get sufficient answers about product, system, process, or application, the following list of areas must be delved into and explored. These are pretty much the same as those discussed in terms of setting requirements, but now the purpose for examining them has changed. It is not which controls and measures should be designed or requirements set, but rather what was actually done and does it meet or exceed the requirements

- *Data:* What data is involved? Are they sensitive? Are they proportional? Do they constitute the minimum necessary?
- *Purpose:* How and why is the data being processed? Is the data being collected in alignment with the services for which the data is being collected? Is the need and reason for each data element documented?
- *Means of collection:* How was the data acquired? From the individual? From another system? From a third party? Were they legitimately collected with notice and choice?

[6] Portions of this article are reprinted from *Core Software Security: Security at the Source* by James Ransome and Anmol Misra. © 2014 CRC Press. Reprinted with permission. www.crcpress.com/product/isbn/9781466560956

- *Notice:* Where was notice presented? What was in the notice? Did it adequately explain how the personal information would be processed? Was it a just-in-time notice or via a link to a privacy notice?
- *Choice/Consent:* What kind of choice is the owner of the data given? Is the use of the data an option? Is consent to process the personal information required? If check boxes were used, was there a prechecked box?
- *Transfer:* Is it possible to transfer the data to third parties or another system? For what and whose purpose? Are contracts in place with the third parties? Has a privacy review been conducted? Is the data protected during transfer? Are there cross-jurisdictional issues?
- *Access, Correction, Deletion:* Does the user have a means of accessing his or her personal information and the ability to correct or delete it should it be false or inaccurate? How is the data segmented to facilitate this? Is it a self-service model? Is there a process documented and tested?
- *Security:* Is the data secure at rest or in motion? Are both required? Is the means of authentication and authorization process sufficient? Is the security mechanism overly invasive?
- *Minimization:* Is the data collected the minimum necessary to achieve the intended purpose? Has the data passed the "minimization test" (as discussed earlier in this chapter)?
- *Proportionality:* Is the processing of the data proportional to the need, purpose, and sensitivity of the data? If the purpose of the processing were to be reported in the media, would it be "embarrassing" to the enterprise?
- *Retention:* Is the deletion strategy defined and enforced within the system or the enterprise? If so, how?
- *Third parties:* If third parties are involved, what is the relationship? Has a contract been signed? What is in the contract? Is a separate PIA required? Has a security review of the third party been completed?
- *Accountability:* Are responsibilities defined and the internal enforcement mechanisms in place? What are they? Who "owns" the program? How is it managed?

Based on an enterprise's specific privacy policy, there may be additional items explored, but for most, this is the basic framework. Also, depending on how detailed or complex the PIA, there may be multiple layers to these questions, and sometimes, additional PIAs are required. For instance, if the subject of an initial PIA is an application that transmits data back to the enterprise, then a PIA is required on the application and the backend system or systems to which the data is transmitted.

Many of the answers to questions the PIA asks can be found in the requirements documentation, such as use case, activity diagrams, use-case metadata, data models, and so on. The difference is that these are source documents and the PIA is a structured analysis of the source material. The PIA is meant to provide a focused discussion of how the privacy requirements of product, system, process, or application are being met within the context of its functionality and data flows.

A set of improvement or remediation recommendations will be included in a PIA. The status of each recommendation will be tracked and reported. Thus, as mentioned earlier, the PIA is a living document. As shown in Figures 10-1 and 10-2, a feedback loop will ensure that goals, policies, processes, and procedures and privacy mechanisms are kept up to date.

PRIVACY ENGINEERING REQUIRES BOTH QUALITY AND SECURE CODE (PART 2)

By James Ransome, PhD, CISSP, CISM, Senior Director, Product Security at McAfee

Quality vs. Secure Code

Although secure code is not necessarily quality code, and quality code is not necessarily secure code, the development process for producing software is based on the principles of both quality and secure code. You cannot have quality code without security or security without quality, and their attributes complement each other. At a minimum, quality and software security programs should be collaborating closely during the development process; ideally, they should be part of the same organization and both part of the software development engineering department. The organizational and operational perspective is discussed further in my latest book, *Core Software Security: Security at the Source*.

"The foundation of software applications, and the development processes that produce them, is based on the common best principles of quality code and secure code. These principles are the driving force behind the concepts and design of industry best practices. To produce secure code that will stand the test of time, you must learn how to incorporate these principles into the development process." Remember that secure code is not necessarily quality code, and quality code is not necessarily secure code.[7]

[7] J. Grembi, *Secure software development: A security programmer's guide*. Boston: Course Technology, 2008. p. 58

Secure code does not mean quality code: You must know how to write quality code before you can write secure code. A developer can write very secure code that authorizes and authenticates every user transaction, logs the transaction, and denies all unauthorized requests; however, if the code does not return expected results, then even this very secure code may never see the light of day. Software quality characteristics are not the same as security. Quality is not measured in terms of confidentiality, integrity, and availability, but rather in terms of ease of use and whether it is reusable and maintainable.[8]

Quality code does not mean secure code: A developer can write efficient code that is easy to maintain and reusable, but if that code allows an unauthorized user to access the application's assets, then the code is of no use. Unlike software quality, software security is not subjective. Sensitive information is either exposed or it is not, and there is no second chance to get it right. Ultimately, quality, security, and maintainability are the three primary goals the industry considers to be of the upmost importance in any secure software development process.[9]

You cannot have quality without security or security without quality. These two attributes complement each other, and both enhance overall software product integrity and market value. Good developers should be able to identify what quality factors are in software and how to code them. Likewise, good developers should know how the software they develop can be attacked and what the weakest areas are in the software; if the code allows an unauthorized user to access the application's assets, then that code is either exposed or it's not, and there is no second chance to get it right.[10]

Privacy and the Security Development Lifecycle

Protecting users' privacy is another important component of the SDL process and should be considered a system design principle of significant importance in all phases of the SDL. Just as with a failure in security, a failure to protect the customer's privacy will lead to an erosion of trust. As more and more cases of unauthorized access to customers' personal information are disclosed in the press, the trust in software and systems to protect customers' data is deteriorating. In addition, many new privacy laws and regulations have placed an increased importance on including privacy in the design and development of both software and systems. As with security, software that has already progressed through the development lifecycle can be very expensive to change; it is much less expensive to integrate privacy preservation methodologies and techniques into the appropriate phases of the SDL to preserve the privacy of individuals and to protect personally

[8] Ibid, pp. 58-60
[9] Ibid. p. 60
[10] Ibid. p. 72

identifiable information data. Some key privacy design principles included in an SDL are the ability to provide appropriate notice about data that are collected, stored, or shared so users can make informed decisions about their personal information; enable user policy and control; minimize data collection and sensitivity; and the protection of the storage and transfer of data.[11]

It is imperative that privacy protections be built into the SDL through best practices implemented within the SDL. Ignoring the privacy concerns of users can invite blocked deployments, litigation, negative media coverage, and mistrust. In my recent book Core Software Security: Security at the Source, my co-author and I have incorporated privacy protection best practices into our SDL.[12]

Vector 3: The Importance and Value of Privacy Impact Assessment to Key Stakeholders

A PIA also serves as a tool that provides confirmation of:

- *Accountability:* External regulators—Should there ever be an inquiry from external regulators, such as data protection authorities, a PIA shows that the organization has a proactive program in place and takes responsibility.
- *Compliance with internal guidelines:* Internal regulators—Should there be a question from an internal regulator such as for an internal audit, a PIA is a quick reference for answering questions. It also shows internal regulators that controls and measures were determined through an analytical process and deliberate steps were taken to avoid risk.
- *QA and continuity:* Product team—A PIA acts as a document from which the product team validates and confirms that the required controls and measures are in place and meet the enterprise's requirements. A PIA acts as a central document so that as requirements and functionality change, privacy requirements are not lost, obscured, neglected, or overlooked, especially as the project moves between teams.
- *Quick reference:* Data incident response teams—In the event of a data incident, a PIA acts as a quick reference to the potential scope of it.

[11] Microsoft Corporation. Microsoft Security Development Lifecycle (SDL), Version 3.2. 2012. www.microsoft.com/en-us/download/details.aspx?id=24308

[12] Portions of this article are reprinted from *Core Software Security: Security at the Source* by James Ransome and Anmol Misra. © 2014 CRC Press. Reprinted with permission. www.crcpress.com/product/isbn/9781466560956

- *Data maps:* IT and data governance team—Because PIAs usually contain data flow diagrams and maps, they can combine to form a "data" atlas for the IT and data governance teams.
- *Peace of mind:* For all—A successful PIA will give all involved peace of mind that the necessary controls and measures are in place and are the result of a structured analysis (as opposed to happenstance).

QUALITY ASSURANCE DOESN'T END AT LAUNCH

By Jules Polonetsky, Executive Director, Future of Privacy Forum

One of the most useful privacy engineering tips that I have picked up over my years as a privacy professional is a very simple concept: Make sure you only get what you intend to get. In the messy world of data, this is easier said than done. It can be hard to know which data a system will eventually need, and it is often easier to collect and log and then figure out what should be used. But consider the backlash over revelation that Google Streetview cars logged the content of Wi-Fi transmissions as they drove by homes to understand the intense criticism and liability that can flow from logging more than is intended.

I first learned the lesson of limited collection by design early in my career. From 2000 to 2002, I served as chief privacy officer at DoubleClick, a new job title that had recently emerged at companies that want to show they were serious about privacy. I found myself running around like a madman, trying to educate employees, get privacy clauses into contracts, and support consumer-friendly policies.

One problem that plagued ad serving companies then and still today was a privacy snafu that occurs when ad tags or tracking pixels were placed on web pages that collected sensitive information. A company might wish to learn which ads were leading to consumers actually purchasing or registering and would set a tracking pixel on a page where a consumer typed in a credit card number or provided an e-mail address.

If the tags were implemented improperly, sensitive personal information would be sent by web sites to DoubleClick's adservers. Now we didn't want these data, we didn't use these data, and we didn't even know when they were being sent to us. Log files are often messy and adserving backend systems would analyze the adserving log files to pluck the fields that the adservers needed to track and target ads. Our policies promised that we served ads anonymously.

It turned out that it was trivial for critics to scan web sites and "discover" that companies were sending sensitive data to DoubleClick. I responded by educating our web site clients, warning them, cajoling, and inserting contract language barring them from sending DoubleClick personal information. All to no avail, as news headline after headline appeared, exposing the latest "data leak." Policy couldn't solve the problem, contracts couldn't solve the problem, and promises that we didn't use the data or even know we were getting them didn't restore the faith.

I was finally able to persuade the engineering team to implement a system that ensured that the adservers truncated on the fly any personal data before we logged them. We continued educating clients not to "leak data" to us, but only when I could show that our system was engineered to only do what we said it would do—serve ads anonymously—could we show that our code backed up our promises.

I wish I could say that the industry took this lesson seriously and future companies avoided making the same mistake. But every year we read of new versions of this type of data leakage at companies that promise that they do not collect or use or share sensitive data. It is against their policy! Unfortunately, policy doesn't trump technology and companies continue to be castigated as data continues to leak. As I write this, the Affordable Health Care web site is being criticized for leaking sensitive health registration data to analytics companies, a blow that this already troubled web site didn't need.

"Privacy by design" has become the mantra of many of us who practice at the intersection of data, policy, and technology. To succeed at privacy by design, privacy professionals need to think like engineers and engineers need to think like privacy professionals. Doing any less leaves the privacy professional subject to the failings of technology and leaves the engineer frustrated with the constraints of policy. In a fast changing world of data innovation, getting policy and technology to align can be the difference between success and failure.

Resources for Conducting Privacy Impact Assessments

There are a number of resources available on the Internet regarding PIA that are worth reviewing:

Privacy Impact Assessments: The Privacy Office Official Guidance. June 2010: www.dhs.gov/xlibrary/assets/privacy/privacy_pia_guidance_june2010.pdf

Privacy Impact Assessment: Towards Best Practices. http://ehealthrisk.wikispaces.com/file/view/PIA%20Best%20Practices%20Guide_PSCIOC.doc/389845494/PIA%20Best%20Practices%20Guide_PSCIOC.doc

Privacy Policy Appendix B Privacy Impact Assessment Template: www.novascotia.ca/just/IAP/_docs/Appendix%20B%20PIA%20Template.pdf

Privacy Impact Assessment (PIA) Guide: www.sec.gov/about/privacy/piaguide.pdf

Privacy Impact Assessment Guide: www.oaic.gov.au/privacy/privacy-resources/privacy-guides/privacy-impact-assessment-guide

THINK WHOLE SYSTEM WHEN DOING QUALITY ASSURANCE

By David Mortman, Chief Security Architect, Dell Enstratius, and Contributing Analyst, Securosis

When hearing about penetration testing, people usually think "security" and not "privacy," but the fact is that testing your application to make sure your privacy controls are effective is just as important. Many security vulnerabilities will lead to privacy failures as well. Any time you have an attack that leads to a data disclosure you have a potential privacy issue as well. So in a very real sense, privacy vulnerabilities are a subset of security vulnerabilities. This is yet another example of the truism that security doesn't require privacy, but privacy requires security. When your security team plans the application penetration test, work with them to identify the areas of the application that contain privacy-related data so the pen-test team can prioritize those areas. Remember when performing the assessment to focus not only on the web frontend, but also on any mobile apps, administrative interfaces, and APIs that are exposed as they are also vectors of unintentional disclosure.

Conclusion

Quality assurance and conducting privacy reviews go hand in hand. The PIA is the tool that enables the privacy engineer to discerns issues and characterize where they fit into the puzzle of privacy compliance. It acts as a map for the privacy engineer to understand historical aspects of a product and it acts as a headlight to show the future. The next chapter will will discuss how to access and ready your organization for Privacy Engineering.