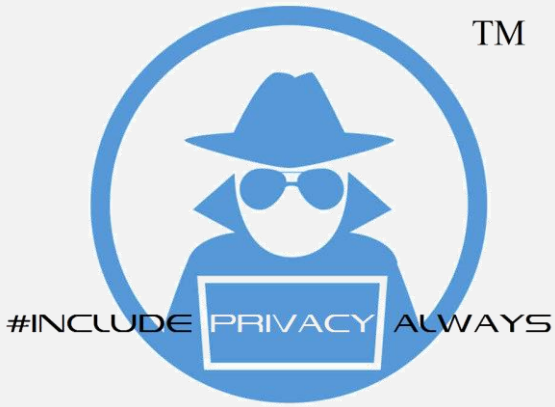
PRIVACY VIRTUOSO PRIVACY VIRTUOSO PRIVACY VIRTUOSO PRIVACY VIRTUOSO

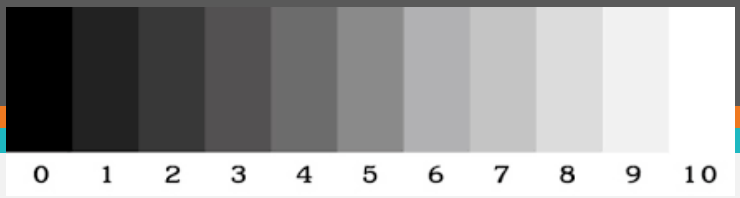


Privacy By Design (PbD) In System Development V1.0

PV™

PRIVACY VIRTUOSO

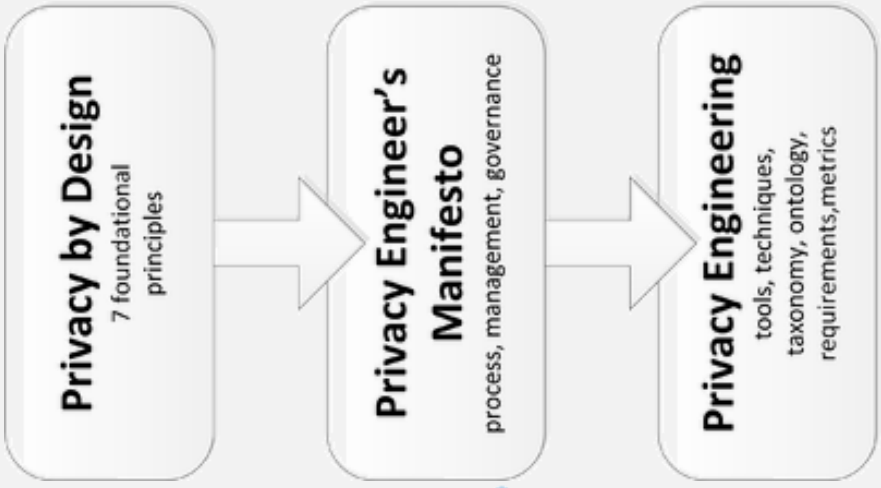
Value Sensitive Design



Value sensitive design (**VSD**) is a theoretically grounded approach to the design of technology that accounts for human values in a principled and comprehensive manner. It addresses design issues within the fields of information systems design and human-computer interaction by emphasizing the ethical values of direct and indirect stakeholders.

VSD preserves human values into account in a well defined matter throughout the whole process. Designs are developed using a three-way investigation consisting of three phases: **conceptual**, **empirical** and **technological**. These investigations are intended to be **iterative**, allowing the designer to modify the design continuously.

Privacy by Design (**PbD**) is an example of VSD that is concerned with respecting the privacy of end users of (IT) systems.



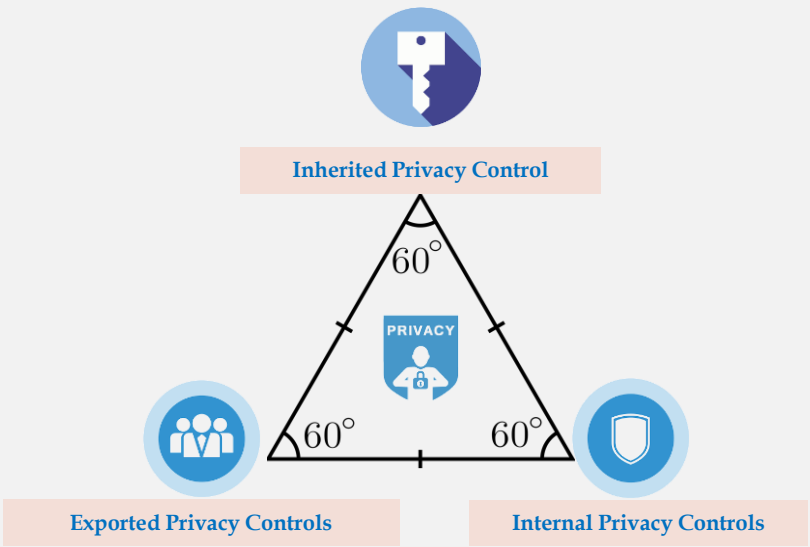
Privacy By Design (PbD)

PRIVACY VIRTUOSO PRIVACY VIRTUOSO PRIVACY VIRTUOSO

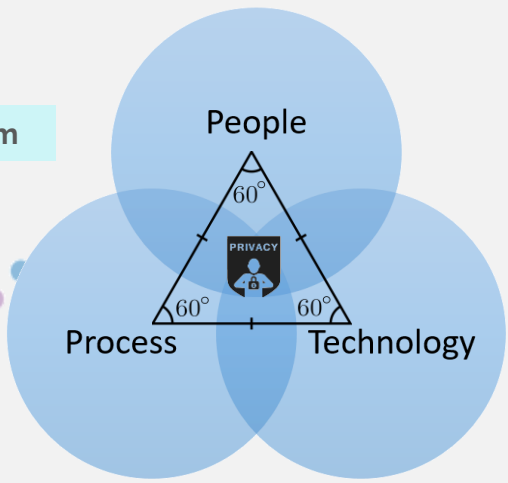
Privacy by Design is an approach to systems engineering which takes privacy into account throughout the whole engineering process.

This should always be **net resultant** of all the expected privacy controls required.

We depict it with our Privacy Iron Triangle (**PIT**).



Fitment of PIT in development ecosystem



PbD seven foundational principles

1. Proactive not Reactive

Preventative not Remedial, which means to anticipate and prevent privacy invasive events before they happen.

2. Privacy as the Default Setting

To ensure that personal data are automatically protected in any given IT system or business practice. No action is required by the user – privacy is built in by default.

3. Privacy Embedded into Design

Not bolted on as an add-on. Privacy becomes an essential component of the core functionality being delivered.

4. Full Functionality — Positive-Sum, not Zero-Sum

Meaning that one seeks to accommodate all legitimate interests and objectives in a positive-sum "win-win" manner. The purpose is to avoid dichotomies, such as privacy vs. security or privacy vs. functionality.

5. End-to-End Security

Full Lifecycle Protection, to ensure that all data are securely retained throughout its entire lifecycle, and then securely destroyed at the end of the process, in a timely fashion.

6. Visibility and Transparency

Keep it Open, to assure all stakeholders that whatever the business practice or technology involved, it is in fact, operating according to the stated promises and objectives.

7. Respect for User Privacy

Keep it User-Centric, which requires architects and operators to keep the interests of the individual uppermost by offering such measures as strong privacy defaults, appropriate notice, and empowering user-friendly options.

Proposed Model of System Development Life Cycle

PRIVACY VIRTUOSO PRIVACY VIRTUOSO PRIVACY VIRTUOSO

Historically three classic models have been used for SDLC – Waterfall , Iterative and Spiral , they offered a set of benefits and had their own inadequacies to tackle the changing development and IT landscape.

Since late 2003 while Agile has certainly addressed a significant portion of such shortcomings, it also has diluted and make complexities on matters that concerns – NFR (non Functional Requirement) such as Security , Compliances , Legal and to add to list **Privacy**.

With our model which brings to table the benefits of both Agile and PbD we are addressing the issues of Negligence on matters that concerns Information and Data Privacy from the very inception of system. Our model also allows you to address any other NFR that may be of higher precedence while on your development life cycle.

We help organization to adapt these principles into a company's chosen SDLC model. There are two commonly accepted models: waterfall and agile. One size fits all will not work for Privacy Related Checks & Controls.

Adding privacy by design into the waterfall model

Waterfall Model -uses a linear series of phases in which the output of each phase becomes the input for the next phase. At a phase gate review - each phase has a hard stop, there is no going back once a phase has reached completion. The 6 phases typically include:

(1) Requirements

(2) Design

(3) Develop

(4) Test

(5) Deploy and

(6) Maintenance.

Requiring that Privacy Threshold Analysis (PTA*), Privacy Impact Assessment (PIA*), and Data Protection Impact Assessment (DPIA*) processes are conducted during the first **two phases** ,will increase the likelihood that "Data Protection by Design and by Default" requirements — and thus privacy and data protection requirements — are embedded into projects at the outset. Projects with personal data should have strong change-management controls and in effect not advance to the next phase until privacy and data protection requirements are approved.

Six phases of the SDLC waterfall model

	PHASE 1	PHASE 2	PHASE 3	PHASE 4	PHASE 5	PHASE 6
	Requirements	Design	Develop	Test	Deploy	Maintenance
ZONE A	Perform an initial PTA, PIA or DPIA (conceptual)	Minimize data collection	Obtain initial data subject consent on personal data collection, use, disclosure, and retention	Monitor and report privacy controls through periodic testing and evaluation	Integrate any new privacy protection methods or controls into systems for improved privacy	Ensure proper management of new applications and technology in production
ZONE B	Review privacy and information security policies, standards, and controls to ensure they are following requirements for collection, use, retention, and disposal of personal data	Perform a formal PTA, PIA or DPIA (formal and feasibility)	Ensure data subjects understand systems and process (transparency)	At large developments (Scope , Cost ,Time) recheck the PTA, PIA and DPIA tools for their current applicability.	Analyze privacy policies, standards, and procedures and system performances for irregularities	
ZONE C	Define baseline and custom privacy system requirements	Analyze the relevant privacy controls to ensure they are designed, developed, and implemented	Ensure data subjects are informed on how to access their personal data and ensure it is up to date and accurate			
ZONE D		Design and implement feedback to control privacy mechanisms into system	Implement security measures to ensure protection of personal data			
ZONE E			Perform ongoing testing and evaluation			

Six phases of the SDLC waterfall model & privacy

* *This will be discussed in details while the training and overview engagement meetings.*

Adding privacy by design into the Agile model

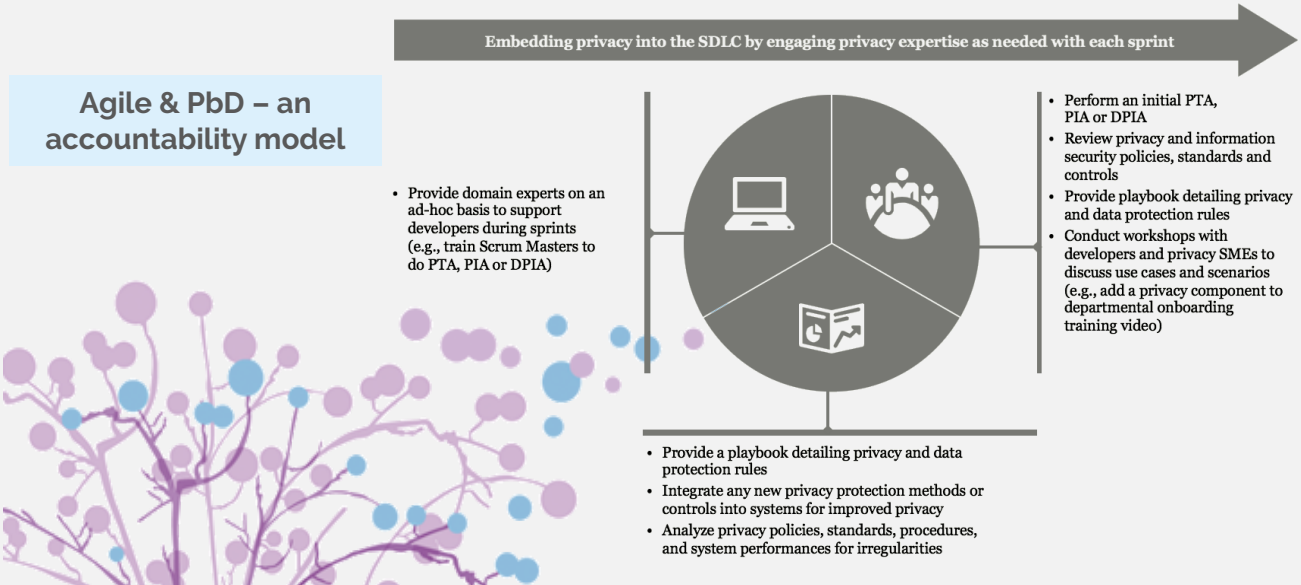
Agile advocates of an iterative approach to software development in which software is developed in incremental, rapid cycles often called sprints. Agile development is ad-hoc and continuous, while adhering to project documents as guiding principles. During the agile SDLC process, the privacy office must work with the business and technology teams to increase privacy-awareness.

A pivotal step is ensuring agile teams consider privacy when deciding on which working rules and project documents to incorporate privacy principles throughout the lifecycle.

The following asks are recommended to increase privacy awareness during Agile sprints:

- Provide developers — designers, coders, testers, release managers — with a playbook
- Provide access to domain experts
- Conduct workshops with developers and privacy subject matter experts

Agile & PbD – an accountability model



What needs to be addressed ?

	Steps to Resolve	Under New Management
People	**Process**	**Technology**
• Changing behaviors around data collection, use • Consent	• Means to address inquires • Limits on what is collected • Data lifecycle management	• Find GDPR data • Classify GDPR data • Track GDPR data • Confirm where GDPR data isn't

*"data should be processed in a manner that ensures appropriate security of the personal data, including protection against unauthorized or unlawful processing and against accidental loss, destruction or damage, using appropriate technical or organizational measures ('**integrity and confidentiality**')."*

"When **developing, designing, selecting and using applications, services and products**…take into account the **right to data protection**…with due regard to the state of the art, to make sure that controllers and processors are able to fulfil their data protection obligations. "

Key elements to factor while entrusting PbD in SDLC

Design patterns

Design patterns are useful for making design decisions about the organization of a software system. A design pattern "*provides a scheme for refining the subsystems or components of a software system, or the relationships between them. It describes a commonly recurring structure of communicating components that solves a general design problem within a particular context.*"

Design strategies

Many design patterns are very specific, and therefore cannot be applied directly in the concept development phase. However, there exist design patterns of a higher level of abstraction, called architecture patterns. Such architecture patterns "express a fundamental structural organization or schema for software systems. They provide a set of predefined subsystems, specify their responsibilities, and include rules and guidelines for organizing the relationships between them."

Privacy-enhancing technologies

"Privacy-Enhancing Technologies is a system of ICT measures protecting informational privacy by eliminating or minimizing personal data thereby preventing unnecessary or unwanted processing of personal data, without the loss of the functionality of the information system."

Limits of privacy by design

PbD is a technical approach to a social problem. Obviously technology cannot help with all related aspects. Especially in the field of privacy, which touches various basic rights topics, such as freedom of expression and press, or protection from discrimination, issues have to be tackled in a grander scheme by society as a whole.

There is a caveat to this: a significant part of the low-level privacy invasion is the direct result of the internal functioning of technical systems. Thus, while the incentives and will to invade privacy may be social problems, the actual ability to do so is a technical problem in many instances. Thus, dealing with it at the technology level is necessary.

Fragility of privacy properties

Many privacy properties are fragile with respect to composition, i.e., if a system that fulfils a certain property is embedded within or connected to another system, it is hard to assess if that privacy property is preserved. Similarly, even if two systems that fulfil a given property are combined, the resulting composed systems might not fulfil the property that was individually provided by both systems.

This non-composability can easily be proven; however, the proof is by counter-example: For instance, given a message that is considered privacy-relevant data, i.e. it should not be disclosed, and two systems, where system 1 encrypts the message and sends the cypher text over a public channel and system 2 encrypts the message and sends the key over a public channel.

Both systems alone do preserve the confidentiality of the message; however, if combined in a way that the same key is used for the same message in both systems, the confidentiality of the message cannot be guaranteed. By the nature of such a proof by counter-example, it does not provide properties for composition. Hence one must also evaluate how they have been put together to ensure the resulting system is also compliant.

Privacy metrics and utility limitations

In general a privacy metric allows comparing two systems with the same or similar functionality with respect to a set of privacy properties. From an information-theoretic point of view these metrics could be generalized. Since privacy has a social component, i.e., in different societies the value of a certain data item differs, the information-theoretic point of view would fall too short for data minimization. Hence the notion of what usage of data is minimal differs. This lack of metric limits the practicality of the data minimization principle, since it is not clear how to construct the objective function that is to be minimized. For instance, systems claiming data minimization properties may indeed abstain from a few data attributes, but at the same time be far from an optimal solution. It may even be the case that those systems with some first steps towards data minimization would need an entire re-design for optimizing this privacy property because there would not be a transition possibility in the chosen system construction towards further data minimization.

Increased complexity

Improving privacy might add user burden and friction to a system. Mental models and metaphors from the brick and mortar world might not always work in the cyberworld. Furthermore, the complexity of the system increases even if naive implementations are considered. Privacy properties might add even more complexity, often by distribution of knowledge and power.

Unclear or too narrow interpretation of privacy by design

A further limitation of privacy by design occurs when it is very narrowly interpreted. Many decision makers use the terms privacy and data protection interchangeable ignoring certain aspects of privacy.

PRIVACY VIRTUOSO PRIVACY VIRTUOSO PRIVACY VIRTUOSO PRIVACY VIRTUOSO

Authored by : Rakesh Jha PMP CIPM CSM

Mails : rakesh@privacyvirtuoso.com

Special thanks and acknowledgements to references

-European Union Agency for Network and Information Security

-Scott Amber Associates – SDLC

-Compliance Forge & Softeng

- Wikipedia and google common artworks

All rights reserved
Privacy Virtuoso Advisory LLP © 2018

PRIVACY MANAGEMENT SOLUTIONS

ASSESSMENTS | ADVISORY | TRAINING

Registered Office:
PRIVACY VIRTUOSO ADVISORY LLP.
402 SRINIVASA RESIDENCY, SERILINGAMPALLY, HERITAGE FRESH LANE,
KONDAPUR, HYDERABAD, TELANGANA, 500084, INDIA.

(O) IN 989 907 4042 (O) IN 931 381 6270 (M) IN 971 114 5647
Email: mails@privacyvirtuoso.com | www.privacyvirtuoso.com