MITRE Privacy Engineering Framework and Lifecycle Adaptation Guide

MITRE Privacy Engineering

Version 2

29 September 2019

For questions or comments about this document, please send a message to privacy@mitre.org

MITRE

© 2019 The MITRE Corporation. All rights reserved.

Approved for Public Release; Distribution Unlimited. 19-00598-5

Purpose

- **This document provides a Privacy Engineering Framework that can be used to integrate privacy into the traditional systems engineering "V" life cycle.**
- **Guidance for adapting the Framework to other life cycles beyond Waterfall types, such as Agile (incremental) and Spiral (iterative) life cycles, is provided in an Appendix.**
- **For questions or comments about this document, please send a message to privacy@mitre.org.**

© 2019 The MITRE Corporation. All rights reserved.

MITRE

Structure of the Privacy Engineering Framework Document

- **Background information**
  - Why privacy engineering is needed
  - Foundational concepts behind privacy engineering
  - Definition of privacy engineering
- **The Privacy Engineering Framework**
  - A sequential privacy engineering life cycle based on a traditional systems engineering "V" life cycle, grouping activities into three broad categories:
    - Privacy Requirements Definition
    - Privacy Design and Development
    - Privacy Verification and Validation
  - A description of specific activities associated with each category
  - Road map for development of a Privacy Engineering Implementation Strategy
- **Appendix A: MITRE Privacy Engineering Framework Life Cycle Adaptation Guide**
  - Guidance for adapting the Framework to other life cycles beyond Waterfall types, such as Agile (incremental) and Spiral (iterative) life cycles

© 2019 The MITRE Corporation. All rights reserved.

MITRE

Why Privacy Policy and Process Is Not Enough

Many organizations rely on the following activities to address privacy risks:

- Policy
- Risk assessments (PIAs)
- Notice
- Records management
- Accounting of disclosures
- Data flow mapping
- Data loss prevention
- Metrics

Yet privacy risks remain and privacy breaches continue to rise. Why? Because these things alone do not *proactively a*ddress privacy risks at the appropriate level of specificity for a given system. To be effective, systems containing PII must be capable of:

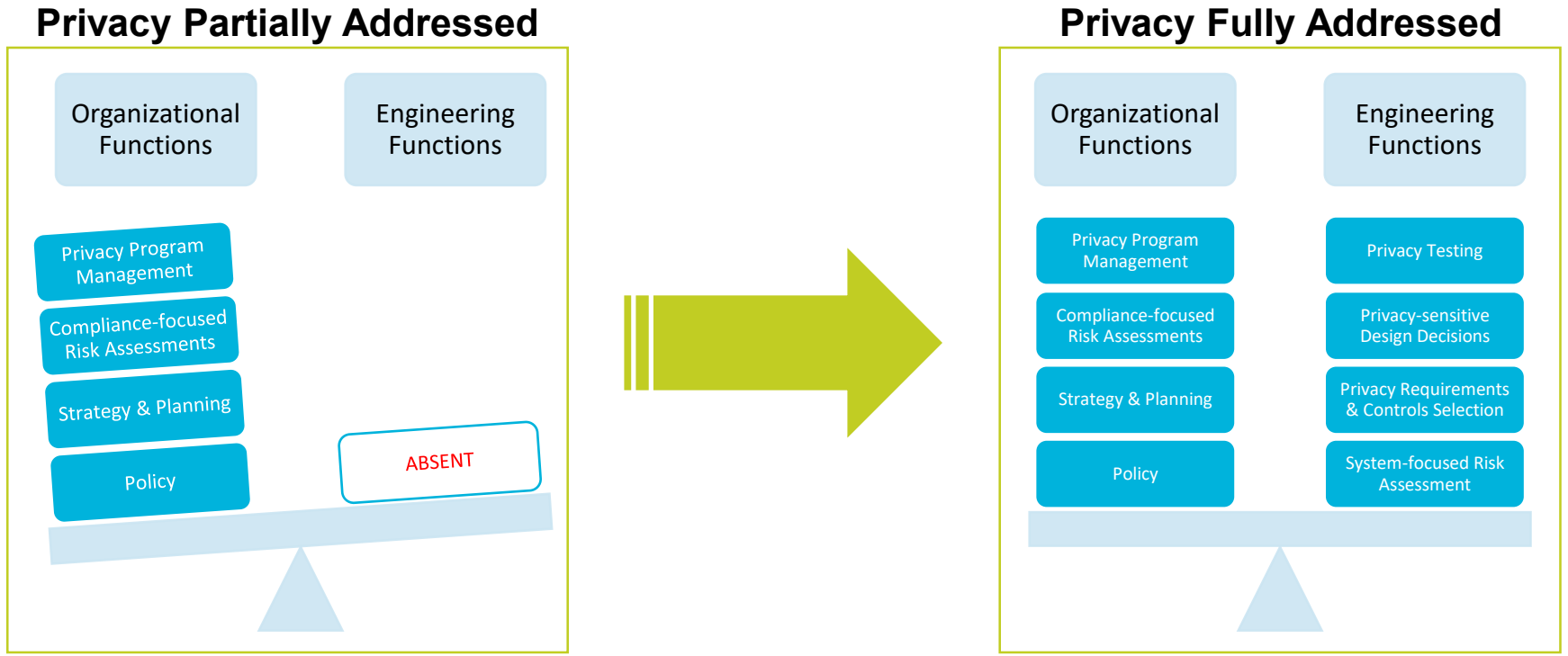
- Preventing or minimizing the effect of human error or fallibility
- Appropriately constraining system actions

© 2019 The MITRE Corporation. All rights reserved.

MITRE

Overcoming Policy and Process Gaps

To adequately address privacy risks, systems that manage PII must behave in a privacy-sensitive manner. Systems engineering processes are a largely untapped opportunity to embed privacy requirements into organizational activities in a way that provides major impact and will proactively address privacy risks.



© 2019 The MITRE Corporation. All rights reserved.

MITRE

Privacy by Design Philosophical Framework

What is Privacy by Design?

Privacy by Design (PbD) advances the view that **privacy cannot be assured solely by compliance** with regulatory frameworks; rather, **privacy assurance must become** an organization's **default mode of operation**.

PbD applies to:

- **IT**
- **Accountable business practices**
- **Physical design**

Simply stated, privacy is not ensured by policy alone. Adequate privacy requires thoughtful integration with every layer of an organization, including:

- Organization policies and governance;
- Business processes;
- Standard operating procedures;
- System and network architectures;
- IT system design and development practices
- Management of data sources

PbD 7 Foundational Principles*	Practical Application
Proactive not **Reactive**; Preventative not Remedial	Anticipate issues; prevent problems before they arise
Privacy as the **Default Setting**	Personal data protected from inception; individuals need not act to protect data
Privacy **Embedded** into Design	Privacy protections are core, organic functions; not bolted on after the fact
Full Functionality — **Positive-Sum**, not Zero-Sum	Privacy enhances, not degrades, security and functionality
End-to-End Security — **Full Lifecycle Protection**	Security applied to each data lifecycle stage, from creation to archiving or deletion
Visibility and **Transparency** — Keep it Open	Individuals understand data use; privacy practices audited
Respect for User Privacy — Keep it **User-Centric**	Organizational imperative = privacy is about personal control and free choice

*http://www.privacybydesign.ca/index.php/about-PbD/7-foundational-principles

© 2019 The MITRE Corporation. All rights reserved.

MITRE

Privacy Engineering

- **Privacy engineering is a systematic, risk-driven process that operationalizes the Privacy by Design philosophical framework within IT systems by**
  - Segmenting PbD into activities aligned with those of the systems engineering life cycle (SELC) and supported by particular methods that account for privacy's distinctive characteristics
  - Defining and implementing requirements for addressing privacy risks within the SELC using architectural, technical point, and policy controls
    - Privacy requirements must be defined in terms of implementable system functionality and properties
    - Privacy risks, including those beyond compliance risks, are identified and adequately addressed
  - Supporting deployed systems by aligning system usage and enhancement with a broader privacy program
  - The goal is to integrate privacy into the existing systems engineering process; it is not meant to be a separate new process

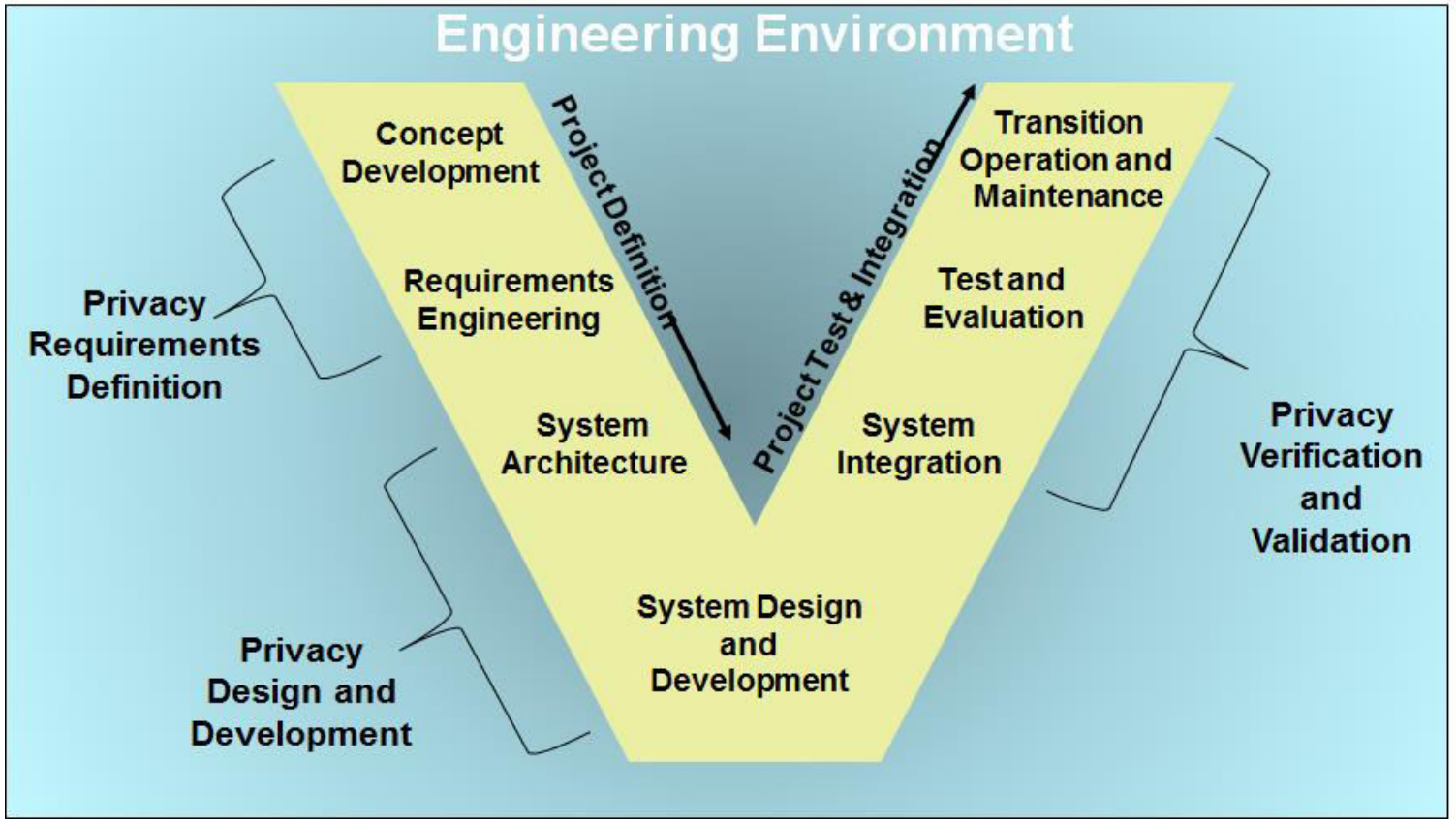
RESULT: *Privacy is integrated into systems as part of the systems engineering process*

© 2019 The MITRE Corporation. All rights reserved.

MITRE

Privacy Engineering Framework

- **This diagram illustrates how the core privacy engineering activities map to stages of the classic systems engineering life cycle.**
- **A mapping exists for every systems engineering life cycle, including Agile development, since every life cycle includes the core activities in some form.**



© 2019 The MITRE Corporation. All rights reserved.

MITRE

Privacy Engineering Activities and Methods

Core Life Cycle Activity	Privacy Method	Method Description
Privacy Requirements Definition: Specification of system privacy properties in a way that supports system design and development	Baseline & custom privacy system requirements	Granular technical privacy requirements derived from first principles and from risk analysis
	Privacy empirical theories & abstract concepts	Methodological constructs based on theories of privacy and socio-technical systems
Privacy Design and Development: Representation and implementation of those elements of the system that support defined privacy requirements	Fundamental privacy design concepts	Explicit or tacit consensus understandings of how privacy works in a system
	Privacy empirical theories and abstract concepts	Methodological constructs based on theories of privacy and socio-technical systems
	Privacy design tools	Specific techniques for achieving privacy
	Privacy heuristics	Experientially developed rules of thumb regarding privacy properties of artifacts
Privacy Verification and Validation: Confirmation that defined privacy requirements have been correctly implemented and reflect stakeholder expectations	Privacy testing & review	Executable tests and targeted document reviews associated with privacy requirements
	Operational synchronization	Analysis of privacy policies & procedures and system behaviors for inconsistencies

© 2019 The MITRE Corporation. All rights reserved.

Privacy Engineering Life Cycle Activities: Privacy Requirements Definition*

The table below lists life cycle activities and their associated inputs and outputs for the Privacy Requirements Definition part of the Privacy Engineering Life Cycle. Use these as references to identify the types of life cycle activities and documentation to add to an existing systems engineering life cycle so that privacy is integrated into the concept development and requirements engineering parts of the life cycle.

Inputs	Privacy Requirements Definition Life Cycle Activities	Outputs
• Baseline privacy requirements [and tests] • Applicable privacy statutes, regulations, policies, and procedures • Functional requirements • Privacy risk model	• Select and refine baseline privacy requirements [and tests] • Analyze privacy risk of functional requirements • Develop custom privacy requirements [and tests] based on results of privacy risk analysis	• System-specific privacy requirements [and tests]

*Brackets indicate optional elements.

© 2019 The MITRE Corporation. All rights reserved.

MITRE

Privacy Engineering Life Cycle Activities: Privacy Design and Development*

The table below lists life cycle activities and their associated inputs and outputs for the Privacy Design and Development part of the Privacy Engineering Life Cycle. Use these as references to identify the types of life cycle activities and documentation to add to an existing systems engineering life cycle so that privacy is integrated into the system architecture and system design and development parts of the life cycle.

Inputs	Privacy Design and Development Life Cycle Activities	Outputs
• System-specific privacy requirements • Functional architecture	• Identify privacy design strategies and patterns • Identify architectural, technical point, and policy privacy controls • Develop data and process models reflecting identified privacy controls • Align, integrate, and implement privacy controls with functional elements • Analyze privacy risk of overall design	• Implemented system components • Acceptable residual privacy risks

*Brackets indicate optional elements.

© 2019 The MITRE Corporation. All rights reserved.

MITRE

Privacy Engineering Life Cycle Activities: Privacy Verification and Validation*

The table below lists life cycle activities and their associated inputs and outputs for the Privacy Verification and Validation part of the Privacy Engineering Life Cycle. Use these as references to identify the types of life cycle activities and documentation to add to an existing systems engineering life cycle so that privacy is integrated into the system integration, test evaluation, and transition operation and maintenance parts of the life cycle.

Inputs	Privacy Verification and Validation Life Cycle Activities	Outputs
• Implemented system components • System-specific privacy requirements [and tests] • Applicable privacy policies and procedures	• Develop/refine privacy test cases • Execute privacy test cases • Check operational behavior against applicable privacy policies and procedures	• Privacy test case results • Documented privacy inconsistencies • Privacy remediation plan with schedule

*Brackets indicate optional elements.

© 2019 The MITRE Corporation. All rights reserved.

MITRE

Road Map for Development of a Privacy Engineering Implementation Strategy

Identify stakeholder representatives

- Privacy
  - Chief Privacy Officer and Senior Agency Official for Privacy
  - Component Privacy Officers or officials
  - Privacy analysts
- Engineering
  - Process owner(s)
  - Project leads
  - Engineers
- Business owners
- Chief Information Security Officer and/or other cybersecurity managers and analysts
- Legal

Examine current systems engineering processes

- Life cycle(s) used
- Life cycle activities
- Roles within life cycle(s)
- Supporting tools

Develop integration plan

- Map privacy engineering activities to life cycle activities
- Identify specific privacy methods supporting privacy engineering activities within life cycle(s)
- Define necessary changes to existing life cycle activities, including roles, and supporting tools
- Prototype updated systems engineering processes, guidance, and tools
- Pilot prototypes and revise as necessary
- Roll out updated systems engineering processes, guidance, and tools

© 2019 The MITRE Corporation. All rights reserved.

MITRE

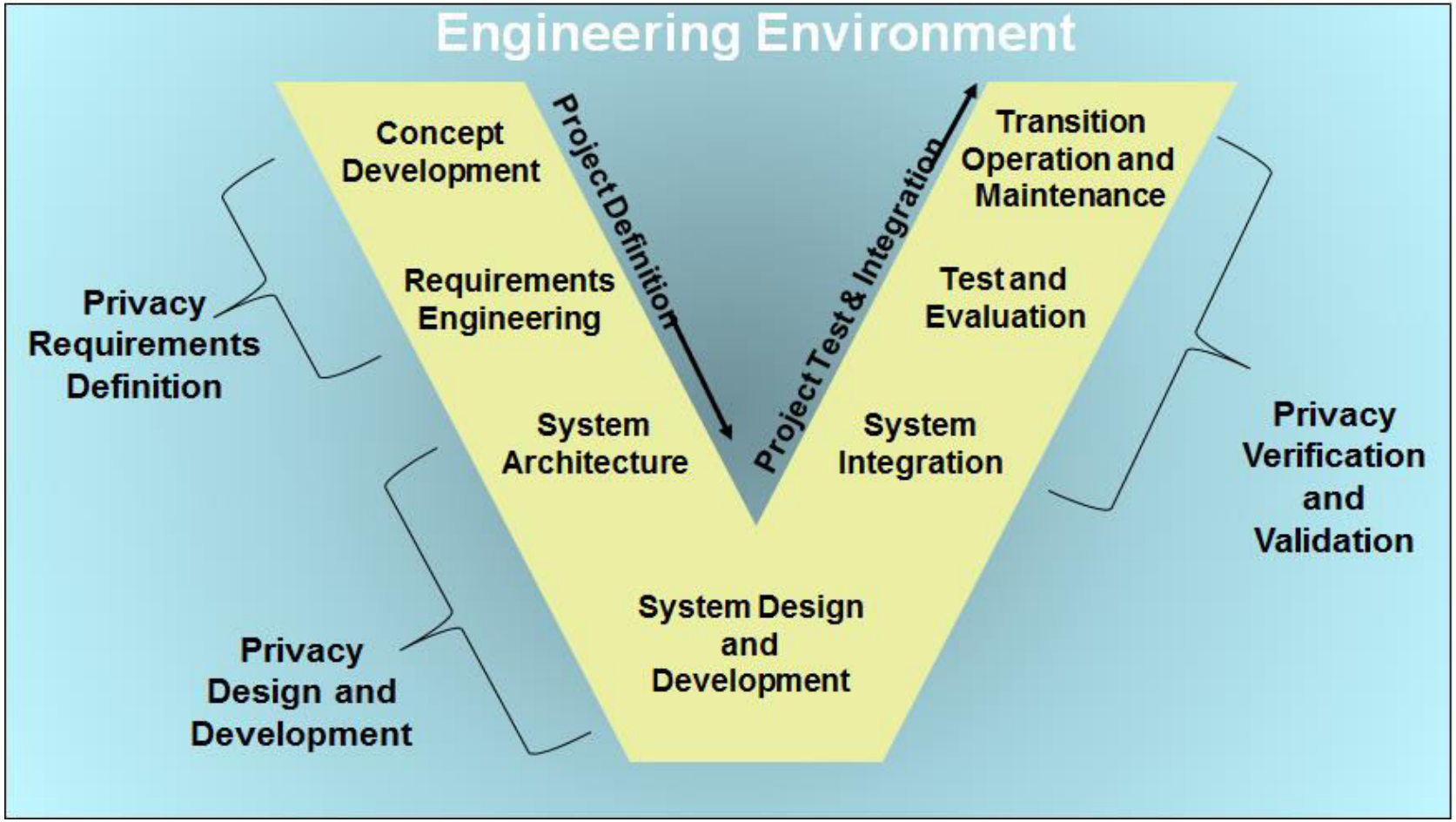
Appendix A
MITRE Privacy Engineering Framework Life Cycle Adaptation Guide

Organization and Use (1/6)

- **Privacy Engineering Framework**
  - MITRE's Privacy Engineering Framework describes a sequential privacy engineering life cycle based on a traditional systems engineering "V" life cycle, grouping activities into three broad categories—Privacy Requirements Definition, Privacy Design and Development, and Privacy Verification and Validation. This organization is depicted on the next slide. The Framework describes specific activities associated with each category.
- **Adapting the Framework**
  - Adapting the Framework to other life cycles involves a series of determinations regarding the extent to which a given activity should be applied to a given object at a particular point in the life cycle.
  - This guide provides structured guidance in making these determinations.

© 2019 The MITRE Corporation. All rights reserved.

Organization and Use (2/6)



© 2019 The MITRE Corporation. All rights reserved.

MITRE

Organization and Use (3/6)

- **Activity**
  - A life cycle activity within any Framework category can be viewed as an action-object pair. Examples of privacy engineering life cycle activities are:
    - Select and refine baseline privacy requirements [and tests]
    - Identify privacy design strategies and patterns
    - Execute privacy test cases
- **Action**
  - Actions refer to significant verbs associated with life cycle activities, e.g., select, refine, analyze, and execute.
- **Object**
  - Objects refer to significant nouns associated with life cycle activities, e.g., requirements, tests, and controls. Objects typically include multiple constituent elements.

© 2019 The MITRE Corporation. All rights reserved.

Organization and Use (4/6)

- **Adaptation**
  - Adapting a Framework life cycle activity consists of appropriately allocating the relevant action and its object.
  - There are three principal types of adaptation: mapping (to objects), scoping (of actions), and distributing (actions and objects).
- **Mapping**
  - Mapping is the process of carrying out an action on specific elements of larger objects. It is appropriate for incremental life cycles such as Agile and DevOps and guidance is based on the type of action. The action is fully applied to a fraction of the object in any given increment.

© 2019 The MITRE Corporation. All rights reserved.

Organization and Use (5/6)

- **Scoping**
  - Scoping is the process of gradually adding detail to a particular object. It is appropriate for iterative life cycles such as Rapid Prototyping and Spiral and guidance is based on the type of object. The action is only fractionally applied to the full object in any given iteration.
- **Distributing**
  - Distributing is the process of spreading action-object pairs across sequential life-cycle stages. This is appropriate for a Waterfall life cycle and is not further addressed in this document as it is the default Framework structure.

© 2019 The MITRE Corporation. All rights reserved.

Organization and Use (6/6)

- **Depending on the type of life cycle being used, actions will need to be mapped or objects will need to be scoped for each of the privacy engineering life cycle activities (see Slides 9-11 for the lists of privacy engineering life cycle activities)**
  - Relevant guidance is determined by actions (incremental life cycles) or objects (iterative life cycles)
- **The slide after the example that follows lists current actions (left column) and objects (right column) invoked in the Privacy Engineering Framework**
- **Enter slide show mode, click on the relevant action or object, and you will be taken to a content slide with a definition and mapping or scoping guidance**
- **Click on the back arrow on the content slide to return to the list of actions and objects**

© 2019 The MITRE Corporation. All rights reserved.

MITRE

Example: *Select* baseline privacy *requirements*

Agile (incremental) life cycle

Baseline privacy requirements selected for a given sprint must address those features that involve personally identifiable information (PII) at any point in the information life cycle: collection/creation, processing, use, disclosure, retention, and destruction. Use relevant information life cycle stages as an initial filter, then consider feature specifics to find potentially applicable requirements. Translate the general requirement as necessary so that it is framed in terms related to the feature(s).

Spiral (iterative) life cycle

Baseline requirements selection will depend on the objectives of each Spiral. The extent to which those requirements are specified will depend on the overall risk identified for that Spiral. For example, to the extent there is uncertainty about the types of PII in the system, especially their sensitivity, or relevant aspects of the information life cycle as it pertains to PII, applicable privacy requirements may need to be contingent or imprecise. Selection of interdependent requirements will be conditioned on those with the most uncertainty attaching to them.

© 2019 The MITRE Corporation. All rights reserved.

MITRE

Actions

- Select (Go to Slide 23)
- Refine (Go to Slide 24)
- Analyze (Go to Slide 25)
- Develop (Go to Slide 26)
- Identify (Go to Slide 27)
- Align (Go to Slide 28)
- Integrate (Go to Slide 29)
- Implement (Go to Slide 30)
- Check (Go to Slide 31)
- Execute (Go to Slide 32)

Objects

- Requirements (Go to Slide 33)
- Tests (Go to Slide 34)
- Risks (Go to Slide 35)
- Design Strategies & Patterns (Go to Slide 36)
- Controls (Go to Slide 37)
- Models (Go to Slide 38)
- Functional Elements (Go to Slide 39)
- Design (Go to Slide 40)
- Operational Behavior (Go to Slide 41)
- Policies & Procedures (Go to Slide 42)

© 2019 The MITRE Corporation. All rights reserved.

MITRE

Select

(Return to Slide 22)

Definition

Selection involves choosing something from a defined set of options. Selection may be deterministic, employing a decision algorithm, or it may be non-deterministic, relying upon judgment. The criteria for making the selection may be explicit or implicit and objective or subjective.

Mapping Guidance

The defined choices for selection may be a situational subset of a larger set of choices. Ensure that only choices relevant for applicable object elements are considered by filtering as necessary. Irrelevant criteria and/or values may introduce a degree of non-determinism and/or subjectivity into nominally deterministic/objective selection processes.

Choices made in one application cycle may constrain or dictate those in other cycles. Document these cases and account for them as necessary in subsequent cycles.

© 2019 The MITRE Corporation. All rights reserved.

Refine

(Return to Slide 22)

Definition

Refinement involves changing something for the purpose of rendering it more precise (i.e., less ambiguous), more detailed, or more properly formed with respect to some set of rules (including syntax).

Mapping Guidance

As long as the elements of the target object are cleanly separable, refinement of specific elements is relatively straightforward. However, this may not be the case as ambiguities may create overlap between elements. In that event, it may be necessary to refine all of the overlapping elements in tandem, even if some are not the intended targets of the refinement.

© 2019 The MITRE Corporation. All rights reserved.

Analyze

(Return to Slide 22)

Definition

Analysis involves methodically examining something. This can be in the broad sense of intensively but informally scrutinizing something, however, it more typically involves the systematic application of a structured method which produces a defined result.

Mapping Guidance

Analysis of only a distinct subset of an object can be problematic if that subset exhibits interdependencies with parts of the object outside that subset. Therefore, while the analysis may only target a subset of the object, it's advisable to consider the object holistically to the extent other aspects of the object are available. If interdependencies exist that could or will affect the results of the analysis, those additional factors must be accounted for.

© 2019 The MITRE Corporation. All rights reserved.

Develop

(Return to Slide 22)

Definition

Developing something involves constructing or otherwise creating it. Contrary to dictionary definitions, however, for these purposes it does not include elaboration in and of itself as in this scheme that is considered refinement.

Mapping Guidance

The nature of develop is such that it does not operate on an existing subset of an object, but rather brings an aspect of an object into existence. The development of this aspect, though, may only be partial, with additional development taking place at future points in time, assuming this adds elements that are independent of ones that have previously been created. If future work is focused in some way on pre-existing elements, it is refinement, not development.

© 2019 The MITRE Corporation. All rights reserved.

Identify

(Return to Slide 22)

Definition

Identification involves designating things from an arbitrary set of possibilities to potentially serve a particular purpose. This contrasts with selection, which involves choosing from a defined set of options.

Mapping Guidance

When identifying an element for an object, it is important to keep in mind that the arbitrary set of possibilities will include any other relevant elements previously designated. Therefore, identification at any given point in time should explicitly consider any related prior designations and may not necessarily result in an additional designation.

© 2019 The MITRE Corporation. All rights reserved.

Align

(Return to Slide 22)

Definition

Alignment refers to situating something relative to something else, implying the definition some kind of coordination. This association can be physical as well as abstract, however, the former is likely to be preceded by the latter in the form of systems engineering documentation.

Mapping Guidance

Alignment may involve elements within the same object or elements from different objects. Either way, the elements will be bound together in some kind of relationship. That relationship may be a dependency relationship, including output of one element being used as input to another element, or it may be that one element modifies the other.

© 2019 The MITRE Corporation. All rights reserved.

Integrate

(Return to Slide 22)

Definition

Integration involves combining components into a whole. This may amount to simple inclusion within the same logical or physical boundary or something more extensive, such as enabling linkages between interdependent components (e.g., ensuring components reference the correct common object with which they all interact). Integration differs from alignment in that alignment defines the coordination between components while integration establishes the environment in which that coordination takes place.

Mapping Guidance

Integration requires that all interdependencies in the set of elements it targets are resolved such that all defined coordination *between those elements* can take place. Elements may include defined coordination with other, unintegrated elements as long as the absence of that coordination does not affect defined coordination between the integrated elements.

© 2019 The MITRE Corporation. All rights reserved.

Implement

(Return to Slide 22)

Definition

Implementation involves translating an abstraction into an operational artifact. These starting and end states distinguish it from development, which does not involve this particular combination of states.

Mapping Guidance

The extent to which specific elements of a larger object can be implemented is generally a function of how decomposable the object is. If the object is highly decomposable, fairly granular elements can be independently implemented. If the object is extremely interconnected, though, implementation may need to include elements beyond the specific ones targeted or the use of formal placeholders.

© 2019 The MITRE Corporation. All rights reserved.

Check

(Return to Slide 22)

Definition

Checking is examining something to confirm that it is as expected. That expectation may be captured in an artifact, or it may be implicit in the target or the method used to examine it.

Mapping Guidance

As long as the granularity of the target of the check is consistent with the granularity of the reference it is being checked against, specific elements can be singled out and checked. If the reference is implicit in the elements, the granularity should be reflexive.

© 2019 The MITRE Corporation. All rights reserved.

Execute

(Return to Slide 22)

Definition

Execution is carrying out a defined task in a defined manner.

Mapping Guidance

The granularity of the defined steps of the task must match the granularity of the elements on which it is being carried out. If the granularities do not match, elements must either be combined or decomposed. In the case of the former, this may require elements outside the scope of interest to be included. In the case of the latter, if the elements of interest cannot be further decomposed this implies that there may be an incompatibility between the elements and the task.

© 2019 The MITRE Corporation. All rights reserved.

Requirements

(Return to Slide 22)

Definition

Requirements define behaviors or properties that a system is intended to support or demonstrate, including those related to privacy. They consist of functional (mission-related) and non-functional requirements. The latter are also known as quality attributes. Privacy requirements are typically non-functional requirements. Requirements may be baseline (i.e., generally applicable unless otherwise determined) or system specific.

Scoping Guidance

Requirements in most iterations will exhibit incomplete semantic content, i.e., they will not fully specify any aspect of the system, including where personal data is coming from, what's being done with it, and where it's going. Elements of actions that are dependent on particular kinds of absent specifics will have to be delayed until those specifics are available. To the extent interdependencies exist across different sets of specifics, applicable action elements will have to be delayed until all of the interdependent specifics are available. If the life cycle is structured such that particular sets of specifics can be projected onto designated iterations, action elements can be explicitly mapped to that structure.

© 2019 The MITRE Corporation. All rights reserved.

MITRE

Tests

(Return to Slide 22)

Definition

Tests, broadly defined, involve execution of specific system functionality (e.g., processing of database queries) to verify it works as expected, or examination of system artifacts (e.g., SQL statements) to verify the particulars. In either case, the objective is to establish that the relevant requirements have been correctly implemented. Tests are usually specified by test cases that describe precisely what is being tested, how it is being tested, and the criteria for passing/failing the test.

Scoping Guidance

Because specific tests are associated with specific requirements, tests will evolve as their associated requirements evolve. Therefore, actions applied to tests are subject to the same constraints and contingencies as actions applied to requirements. These include incomplete semantics and interdependencies.

© 2019 The MITRE Corporation. All rights reserved.

Risks

(Return to Slide 22)

Definition

Risks are generally considered adverse consequences (defined by a privacy risk model) together with degrees of likelihood and severity. However, it may be difficult or impossible to assign quantitative values for likelihood or severity. Where these can be assigned, risks can be arithmetically calculated as likelihood x severity (impact). If there is a basis for assigning qualitative values, then an explicit mapping can be used to calculate a categorical risk value. If it is simply infeasible to assign likelihood and/or severity, as a practical matter the risk amounts to the adverse consequence.

Scoping Guidance

Privacy risks are typically a function of requirements or design. Therefore, actions applied to risks are subject to the same constraints and contingencies as actions applied to requirements or design, as the case may be. These include incomplete semantics and interdependencies.

© 2019 The MITRE Corporation. All rights reserved.

MITRE

Design Strategies & Patterns

(Return to Slide 22)

Definition

Privacy design strategies and patterns are generic solutions to privacy problems. Strategies are general approaches while patterns are more specific and may include implementation skeletons. Both strategies and patterns must be tailored to the specific context of the system and the privacy issue or risk they are addressing.

Scoping Guidance

Because design strategies and patterns are fixed objects external to a specific project or system, there will be a threshold of detail necessary in associated objects before an action can be usefully applied to a strategy or pattern. Beyond that point, since strategies and patterns are unspecified below that level of granularity, they can be acted upon consistent with the degree of specificity in the relevant objects.

© 2019 The MITRE Corporation. All rights reserved.

MITRE

Controls

(Return to Slide 22)

Definition

Controls are defined elements that address risk or a requirement that addresses risk. These elements can manifest themselves in architectures and designs (and, by extension, in implementations) as well as in the environment, including associated policies and procedures.

Scoping Guidance

Controls residing at different levels will, of necessity, be fully addressed at different times. Architectural controls, by virtue of being incorporated into the architecture of the system, will need to be fully treated sooner than those incorporated into design, for example. The granularity of actions applied to controls at any given step, therefore, will vary depending on the nature of the control. As a result, controls will vary in the rates at which they approach full resolution.

© 2019 The MITRE Corporation. All rights reserved.

Models

(Return to Slide 22)

Definition

Models are representations of particular aspects of a system. Process models, for example, represent the ordered actions a system will take (often in the form of a flowchart). Data models, on the other hand, represent the data that the system will operate on and maintain and relationships between data elements.

Scoping Guidance

Because models are abstract representations, actions can be easily applied to them with varying degrees of granularity. Models also typically lend themselves to some form of hierarchical decomposition and therefore can be decomposed to whatever degree is required. This enables individual model elements to be particularized, enabling variations in granularity across a model.

© 2019 The MITRE Corporation. All rights reserved.

Functional Elements

(Return to Slide 22)

Definition

Functional elements are parts of the system that address one or more functional requirements. These components typically will most clearly manifest themselves in system architecture and/or design and particular non-functional elements may need to be aligned with them.

Scoping Guidance

Because functional elements are associated with one or more functional requirements, their granularity should correspond to the granularity of the related requirements. Due to the relationship between functional elements and requirements, actions applied to functional elements are subject to the same constraints and contingencies as actions applied to requirements. These include incomplete semantics and interdependencies.

© 2019 The MITRE Corporation. All rights reserved.

MITRE

Design

(Return to Slide 22)

Definition

A design is a system representation that describes a system in sufficient detail to support implementation.

Scoping Guidance

A design can include variations in granularity such that some, but not all of it, can be implemented; actions can be applied in the context of those aspects that are implementable. Depending on the action, interdependencies between implementable and non-implementable aspects may make it impractical to treat those aspects differentially at any given time.

© 2019 The MITRE Corporation. All rights reserved.

Operational Behavior

(Return to Slide 22)

Definition

Operational behavior is how a system implementation acts in practice. Potentially varying levels of implementation granularity, though, will manifest as corresponding differences in operational behavior.

Scoping Guidance

By definition, relevant actions can only be applied to those aspects of the system that are operational at that point in time. Inconsistent degrees of granularity, however, may necessitate differential application across behavioral facets. Even where there are nominal interdependencies, the use of placeholders (component stubs) may result in marked differences across a facet of an integrated system.

© 2019 The MITRE Corporation. All rights reserved.

Policies & Procedures

(Return to Slide 22)

Definition

Policies and procedures govern how a system behaves. In principle, they should be reflected in the system requirements, thus ensuring that they are carried through the remainder of the relevant actions and objects. However, if they are system-specific, they may be developed in tandem with the system.

Scoping Guidance

There is no need to scope application of an action to pre-existing (i.e., general) policies and procedures as they cannot be scoped by definition. Policies and procedures being developed in tandem with the system, on the other hand, will need to be scoped such that their granularity corresponds to the granularity of the objects to which they relate.

© 2019 The MITRE Corporation. All rights reserved.