# *Remarks and Proposals on Privacy by Design Documentation for Software Engineers Version 1.0 Working Draft 0.1*

Antonio Kung

July 26th, 2013

Here are some remarks on the scope and content of the OASIS document.

## Software or System?

The document we are working on is entitled *Privacy-by-design Documentation for Software Engineers*. It probably does not mean that pure software products are targeted. Rather it will target systems where software plays an important role (e.g. smart grid system, surveillance system, intelligent transport system, social networks).

An interesting argument is provided by Len Bass from CMU/SEI (in *Software Architecture in Practice, 3rd edition Addison Wesley*, page 8) when he elaborates on the difference between System architecture and Software architecture:

***System and Enterprise Architectures***

*Two disciplines related to software architectures are system architecture and enterprise architecture […]*

***Are these in the scope for this book? Yes! (Well, no.)***

*System and enterprise architectures share a great deal with software architecture. All can be designed, evaluated and documented […]*

*So to the extent that these architectures share commonalities with software architecture, they are in the scope of this book. But like all technical disciplines, each has its own specialized vocabulary and techniques, and we won’t cover those. Copious other sources do.*

## Documentation Artifacts

During last conf call (July 17th, 2013), there was a discussion on the types of document artifacts to use: UML diagrams, tables and so forth. There was also a remark that agile programming developers tend not to use such diagrams. Martin Fowler (one of the 17 signatories of the agile manifesto) makes an interesting point on this (<http://martinfowler.com/articles/designDead.html#UmlAndXp>) :

*“keep in mind what you're drawing the diagrams for. The primary value is communication […]”*

It probably means that using UML diagrams in the document is not a problem. The OASIS document does not aim at standardizing the documentation format, but at agreeing on the documentation content.

## Documentation Complexity Vs Process Complexity

During the conf call and further to an email from Frank Dawson listing all the possible steps in a privacy data lifecycle, it was pointed out that this was too complex for situations involving one or two software engineers.

This raises the issue of how much documentation is needed.

An interesting scale is Cockburn scale (<http://en.wikipedia.org/wiki/Cockburn_Scale>) as showed in the figure below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **L** | L6 | L20 | L40 | L100 |
| **E** | E6 | E20 | E40 | E100 |
| **D** | D6 | D20 | D40 | D100 |
| **C** | C6 | C20 | C40 | C100 |
|  | **1-6** | **7-20** | **21-40** | **41-100** |

The scale has two dimensions:

* Criticality where
	+ L refers to a system where loss of Life is at stake (e.g. a civil aircraft),
	+ E means a system where Essential money is at stake (e.g. a car defect might imply huge recall costs),
	+ D means a system where Discretionary money is at stake,
	+ C means a system where loss of Comfort is at stake).Frank explained a number of steps
* Size in terms of number of persons

This scale probably implies that more documentation is needed as the project size grows, or when the project criticality is higher. But the complexity of PbD documentation is not necessarily related to criticality or to size. We probably need other scales to capture PbD complexity (and hence PbD documentation complexity):

* amount of collected data?
* category of applications? Surveillance, social networks, location oriented services?
* number of privacy engineers in organization (e.g. google and facebook should have significant staff..)
* amount of legislation constraints?

I am not sure about the way to go: should the document target several levels of documentations (e.g. small scale documentation vs full scale documentation with a screening process – see for instance the ICP PIA handbook - <http://www.ico.org.uk/pia_handbook_html_v2/html/3-app1.html>)? Or should we just say that the document will provide templates for documentation?

## Relationship with Domain Specific Design Processes

There are many different processes and methodologies used to develop a system (e.g Automotive, Railways, Agile programming). One can even wonder whether the iOS apps approval process is not influencing the design process of the developers…

We should probably state that the OASIS document is universal as it focuses on content not format; We could add companion documents in the future on how a specific process could apply the OASIS PbD documentation.

## Proposed Engineering Concepts for PbD

I suggest to write a section inspired from the below picture (from A.Kung, J.C.Freytag, F.Kargl. Privacy-by-design in ITS applications. 2nd IEEE International Workshop on Data Security and Privacy in wireless Networks, June 20, 2011, Lucca, Italy):



* The leftmost column will be changed to follow the SDL cycle used in the OASIS document.
* The second column will focus on privacy as a default and data minimization,
* The third column with focus on E2E security
* The last column will focus on visibility and transparency

## Proposed Body of Concepts and Terminology to Use

Since the document is about PbD documentation for software engineers, it would be useful that we adopt a terminology and vocabulary that is widely used by software engineers. I suggest to use the work from CMU/SEI, well explained in (*Software Architecture in Practice, 3rd edition Addison Wesley)*

In a nutshell, the following concepts are used:

* quality attribute (e.g. availability, security, modifiability)
* the impact on design is described through quality attribute scenarios which includes a source, a stimulus, an artifact, the environment, a response and a response measure



* the responses depends on a architecture decisions called tactics



* the possible tactics for security are summarized in the picture below



I suggest to describe the PbD engineering principles using CMU/SEI software architecture framework. Privacy-by-design would then include tactics for “privacy as a default”, “E2E security”, “Visibility and transparency”.

+++