Table 2.6 - TD Types

| Type	Definitions	Descriptions of situations	Secondary Studies mentioning the TD types
Architecture Debt	It refers to the issues encountered in the architecture of software products. These are considered to be sub-optimal architecture solutions, impacting the internal quality.	- violations of the good and adopted architectural practices (architectural policies and patterns) - Complex architectural behavioral dependencies - Lack of addressing the non-functions requirements	SS1, SS2, SS3, SS5, SS6, SS8 and SS10
Code Debt	It refers to the issues in the source code that may hamper the modularity, reusability, analyzability, and modifiability of the software products	- Complex or duplicate code source - Code source that does not meet the coding standards	SS1, SS2, SS3, SS5, SS8, and SS10
Build Debt	Refers to issues in the build process that make this process harder	- files of build containing code source that does not add value to this task and software products and manual build - "bad" dependencies	SS5 and SS8
Documentation Debt	Refers to the issues found in the documentation	- Lack, insufficient, outdated or inadequate documentation artifacts related to requirement specifications, design models, test plans, test cases, and code comments	SS1, SS2, SS3, SS5, SS7, SS8 and SS10
Design Debt	Refers to technical shortcuts taken in the detailed design, and may be found by analyzing the source code or design models	-Software design deviating significantly from the specified design -Excessive or complex design (methods, class) -Violations of the principles of good object-oriented design - Code smells/grime	SS1, SS2, SS3, SS5, SS8, SS9 and SS10
Requirement Debt	Refers to the distance between the optimal requirements that need to be implemented and the actual software products implementation, under domain assumptions and constraints.	- Implemented requirements but not for all cases - Requirements partially implemented but which do not fully satisfy their required properties - An implemented requirement which, in a way, does not fully satisfy all the non-functional requirements	SS1, SS2, SS3, SS5, SS8, and SS10
Service	Refers to the inadequate using of the software services	- Inadequate selection and use of software services (service-oriented services)	SS8 and SS10
Test/Automation Test Debt	Refers to issues related to testing activities	- Lack, insufficient or inadequate tests; low tests coverage; and deferring testing.	SS1, SS2, SS5, SS7, SS8, and SS10
Versioning	Refers to issues related to source code versioning	- Unnecessary code forks	SS5 and SS8
(*) Usability Debt	Refers to inappropriate decisions related to software usability that will need to be adjusted later	- Lack, insufficient, outdated or inadequate usability standard	SS8
(*) Defect	Refers to the known and deferred defects	- Postergated decisions on fix defects (known and deferred defects)	SS1, SS5, and SS8
(*) People	Refers to issues related to people leading to the incurrence of TD	- lack of knowledge and negligent attitudes	SS8
(*) Infrastructure Debt	Refers to issues related to the infrastructure of development or operation that may contribute to the incurrence of TD	- Lack, insufficient, outdated or inadequate tools or components to support the activities of development, deployment, and operation.	SS1, SS2, SS3, SS5, and SS8
(*) Process Debt	Refers to issues relayed to the adopted software process that may contribute to the incurrence of TD	- Lack of process definition - Insufficient, outdated or inadequate process not providing proper support to the development activities.	SS8
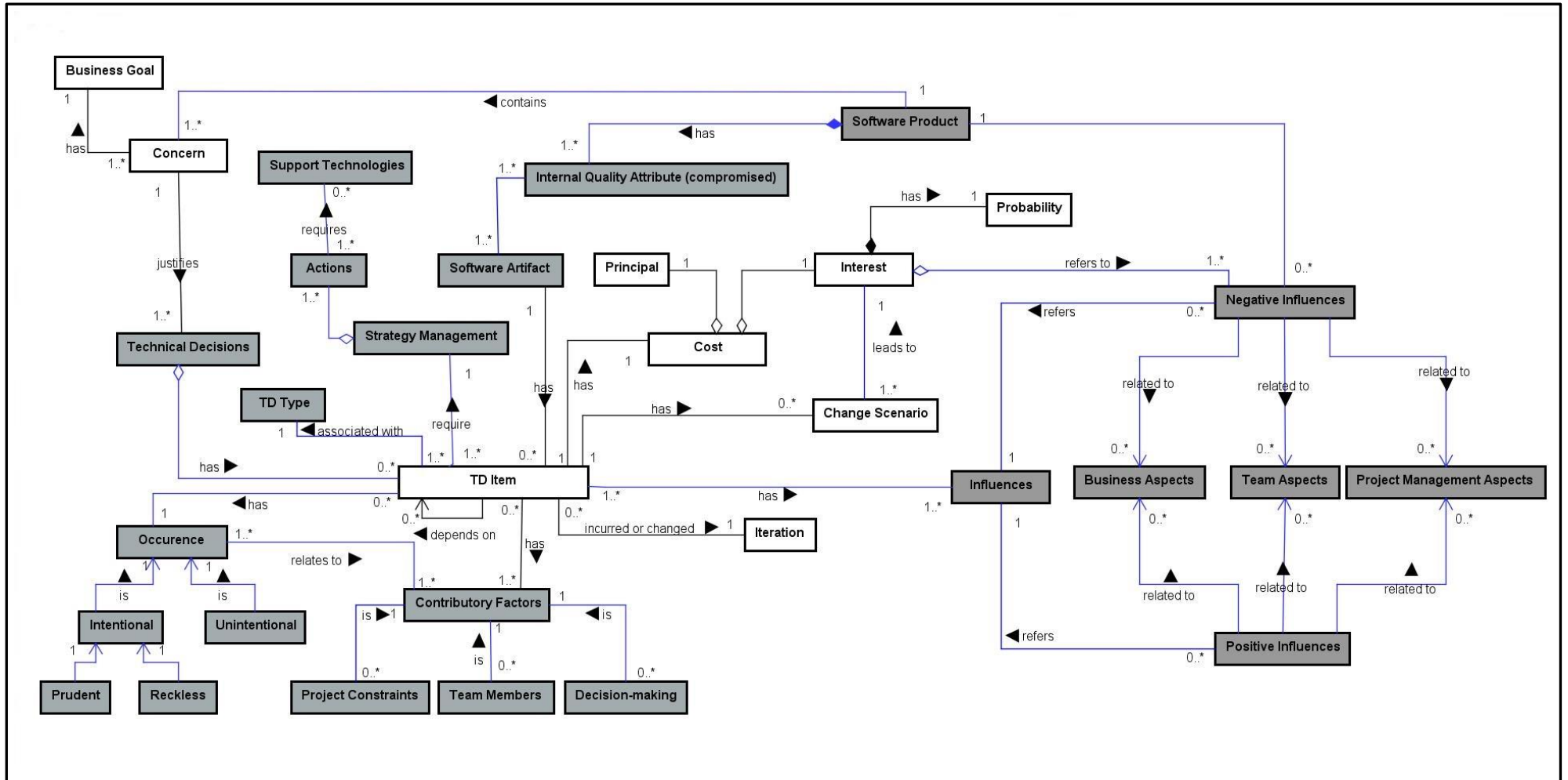
Figure 2.7 - An evolved Conceptual Metamodel of Technical Debt