**WORK PRODUCT(ARTIFACT):DESIGN**

This artifact describes the realization of required system functionality and serves as an abstraction of the source code.

**PURPOSE**

Describe the elements of the system so they can be examined and understood in ways not possible by reading the source code.

**RELATIONSHIP**

|  |  |  |
| --- | --- | --- |
| **Fulfilled Slots** | * [[Technical Design]](http://epf.eclipse.org/wikis/openup/process.openup.base/capabilitypatterns/technical_design_slot_D1C3FAF5.html?proc=_SuWj4dOPEdyqlogshP8l4g&path=_SuWj4dOPEdyqlogshP8l4g,_ieYHIVQwEd2i9JHp7xurvw,_51ewYdOPEdyqlogshP8l4g,_aSrtcdOOEdyqlogshP8l4g,_TV_k8dOKEdyqlogshP8l4g,_gX7xQdOKEdyqlogshP8l4g) | |
| **Roles** | Responsible:   * [Developer](http://epf.eclipse.org/wikis/openup/process.openup.base/capabilitypatterns/developer_82EAB9C9.html?proc=_SuWj4dOPEdyqlogshP8l4g&path=_SuWj4dOPEdyqlogshP8l4g,_ieYHIVQwEd2i9JHp7xurvw,_51ewYdOPEdyqlogshP8l4g,_aSrtcdOOEdyqlogshP8l4g,_TV_k8dOKEdyqlogshP8l4g,_0j1oYKqeEeGLQ_Sv1UBWjg) | Modified By:   * [Developer](http://epf.eclipse.org/wikis/openup/process.openup.base/capabilitypatterns/developer_82EAB9C9.html?proc=_SuWj4dOPEdyqlogshP8l4g&path=_SuWj4dOPEdyqlogshP8l4g,_ieYHIVQwEd2i9JHp7xurvw,_51ewYdOPEdyqlogshP8l4g,_aSrtcdOOEdyqlogshP8l4g,_TV_k8dOKEdyqlogshP8l4g,_0j1oYKqeEeGLQ_Sv1UBWjg) | |
| **Input To** | Mandatory:   * [Implement Solution](http://epf.eclipse.org/wikis/openup/process.openup.base/capabilitypatterns/implement_solution_B7C117A5.html?proc=_SuWj4dOPEdyqlogshP8l4g&path=_SuWj4dOPEdyqlogshP8l4g,_ieYHIVQwEd2i9JHp7xurvw,_51ewYdOPEdyqlogshP8l4g,_aSrtcdOOEdyqlogshP8l4g,_TV_k8dOKEdyqlogshP8l4g,_hVe-oNOKEdyqlogshP8l4g) | Optional:   * [Design the Solution](http://epf.eclipse.org/wikis/openup/process.openup.base/capabilitypatterns/design_solution_7C602F6D.html?proc=_SuWj4dOPEdyqlogshP8l4g&path=_SuWj4dOPEdyqlogshP8l4g,_ieYHIVQwEd2i9JHp7xurvw,_51ewYdOPEdyqlogshP8l4g,_aSrtcdOOEdyqlogshP8l4g,_TV_k8dOKEdyqlogshP8l4g,_frOcQNOKEdyqlogshP8l4g) * [Implement Developer Tests](http://epf.eclipse.org/wikis/openup/process.openup.base/capabilitypatterns/implement_developer_tests_6442995D.html?proc=_SuWj4dOPEdyqlogshP8l4g&path=_SuWj4dOPEdyqlogshP8l4g,_ieYHIVQwEd2i9JHp7xurvw,_51ewYdOPEdyqlogshP8l4g,_aSrtcdOOEdyqlogshP8l4g,_TV_k8dOKEdyqlogshP8l4g,_gXo2UNOKEdyqlogshP8l4g) | External:   * None |
| **Output From** | * [Design the Solution](http://epf.eclipse.org/wikis/openup/process.openup.base/capabilitypatterns/design_solution_7C602F6D.html?proc=_SuWj4dOPEdyqlogshP8l4g&path=_SuWj4dOPEdyqlogshP8l4g,_ieYHIVQwEd2i9JHp7xurvw,_51ewYdOPEdyqlogshP8l4g,_aSrtcdOOEdyqlogshP8l4g,_TV_k8dOKEdyqlogshP8l4g,_frOcQNOKEdyqlogshP8l4g) | | |

**MAIN DESCRIPTION**

This work product describes the elements that will make up the implemented system. It communicates abstractions of particular portions of the implementation.

While architecture focuses on interfaces, patterns, and key decisions, the design fleshes out the technical details in readiness for implementation, or as part of implementation.

This product can describe multiple static and dynamic views of the system for examination. Although various views may focus on divergent, seemingly independent issues of how the system will be put together and work, they should fit together without contradiction.

**TAILORING**

|  |  |
| --- | --- |
| **Impact of not having** | Implementation will proceed with fine-grained, inconsistent tactical decisions that lead to poor-quality software. |
| **Reasons for not needing** | The design typically needs to be represented in some form, although it may be captured in code or tests and not distinct as a separate artifact. In circumstances where a project involves applying well-understood, existing strategies for architecture and design, it is possible that you will not need a *new* design. In those cases, you can simply refer to some existing design. |
| **Representation Options** | It is important that the author of this work product be able to analyze key decisions about the structure and behavior of the system and communicate them to other collaborators. It is also important that these decisions can be communicated at various levels of abstraction and granularity. Some aspects of the design can be represented by source code, possibly with some extra annotations. But more abstract representations of the design will be at a higher-level than source code.  The more abstract representation could use various representation options. UML could be used either strictly or informally; it is a preferred notation based on its rich semantics and broad usage in the industry. Other techniques could be used to communicate the design. Or the design could use a mix of techniques as applicable.  Whether you record these representations on a white board or use a formal tool is not governed by this process. But any representation, whether characterized as formal or informal, should unambiguously communicate the technical decisions embodied by the design. |