|  |
| --- |
| 南京大学 |
| 体系结构 |
| Assignment-1 |

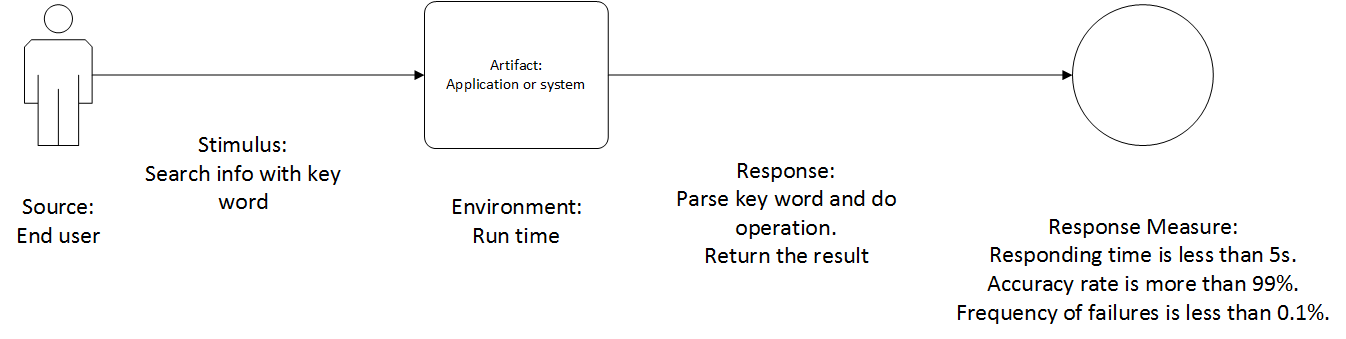
|  |
| --- |
| 管登荣 MF1632020  2016-12-26 |

# Reliability

1. General Scenario

|  |  |
| --- | --- |
| Portion of Scenario | Possible Values |
| Source | Developer, tester, system administrator and end user |
| Stimulus | Request from a user or external system, an operation, emergency |
| Artifact | Code, application, component, configurations, system |
| Environment | Design time, deploy time, startup, shutdown, run time, repair mode, overloaded operation, continuous operation |
| Response | Estimate the risk of failure:   * History data and experiences * Design pattern * Evaluation and review * Comprehensive test   Prevent the failure from happening  Handle the failure  Record and manage the failure:   * Log the fault * Notify appropriate entities( people or systems) |
| Response Measure | Frequency of failures or the probability of success  Time of failures  Time to respond  Accuracy rate of response  Accuracy rate of the operation’s result  Continuous operation hours  General indexes are as follows:   * MTBF: Mean Time Between Failure * MTTR: Mean Time To Repair * MTTF: Mean Time To Failure |

2. Concrete scenario

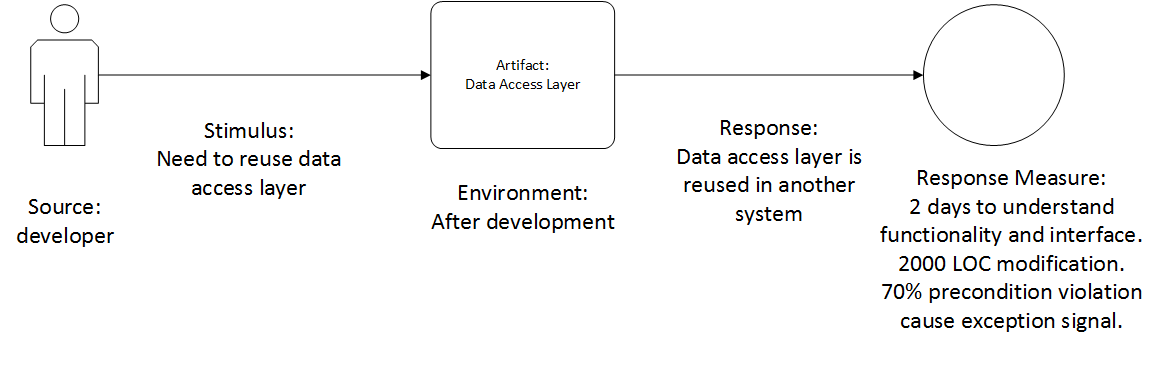


# Reusability

1. General scenario

|  |  |
| --- | --- |
| Portion of Scenario | Possible Values |
| Source | Developer, software architect |
| Stimulus | Develop similar portion, extend system functionality, develop utility class or class for public use |
| Artifact | Framework or component that is going to be reused |
| Environment | After during development |
| Response | Component or framework is reused in another system |
| Response Measure | * Time to understand the functionality of a component or framework. * Modification needed to adapt one component to the specific functional requirement in a new system. * Proportion of precondition violation get handled by exception signaling. |

2. Concrete scenario

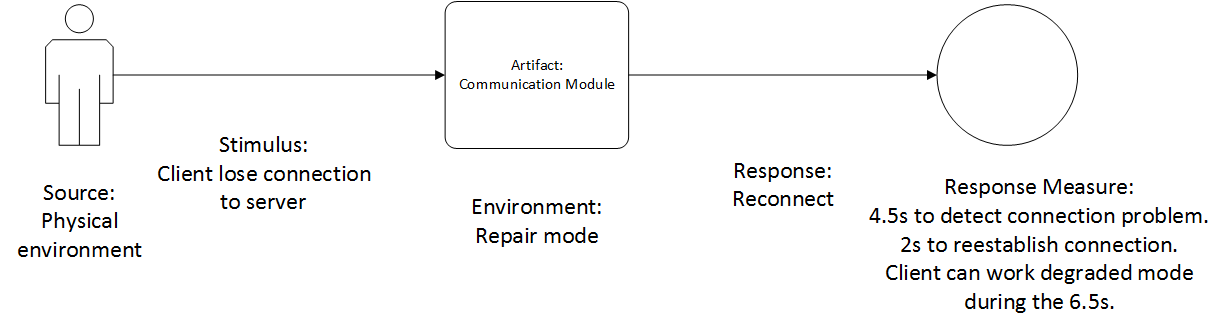


# Recoverability

1. General scenario

|  |  |
| --- | --- |
| Portion of Scenario | Possible Values |
| Source | Developer, system Maintenance Engineer, physical environment, end user |
| Stimulus | Process crash, power off, hardware failure |
| Artifact | Process, persistent storage |
| Environment | Overloaded operation, degraded operation |
| Response | * Log the fault or error message * Restore system to a consistent state |
| Response Measure | * Time to detect the fault * Time to recover from the fault * Time in which system can work in degraded state |

2. Concrete scenario

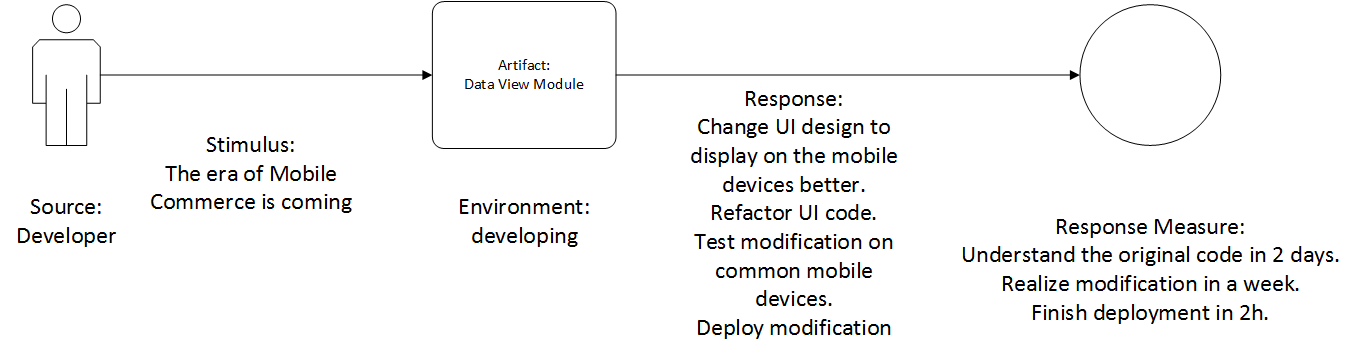


# Maintainability

1. General scenario

|  |  |
| --- | --- |
| Portion of Scenario | Possible Values |
| Source | Developer who works on first, second or muti-development , system administrator and end user |
| Stimulus | Defect, new requirement, changed environment, software evolution |
| Artifact | Code, data, interfaces, components, resources, configurations |
| Environment | Run time, failure even break down |
| Response | Handle the defect:   * Locate defects or their cause * Isolate defects or their cause * Correct defects or their cause, repair or replace faulty or worn-out components without having to replace still working parts   Meet the new requirements:   * Confirm the new requirement * Locate the addition portion * Integrate the addition portion   Cope with a changed environment:   * Read and understand the source code * Design and refactor code   Test modification/addition, and then deploy modification and update the version |
| Response Measure | Cost in the process of maintain, general indexes are as follows:   * MI (Maintainability Index) -- lines-of-code, McCabe and Halstead complexity * WMC (weighted methods per class) * DIT (Depth of Inheritance Tree) * NOC, CBO, RFC, LCOM, etc. |

2. Concrete scenario

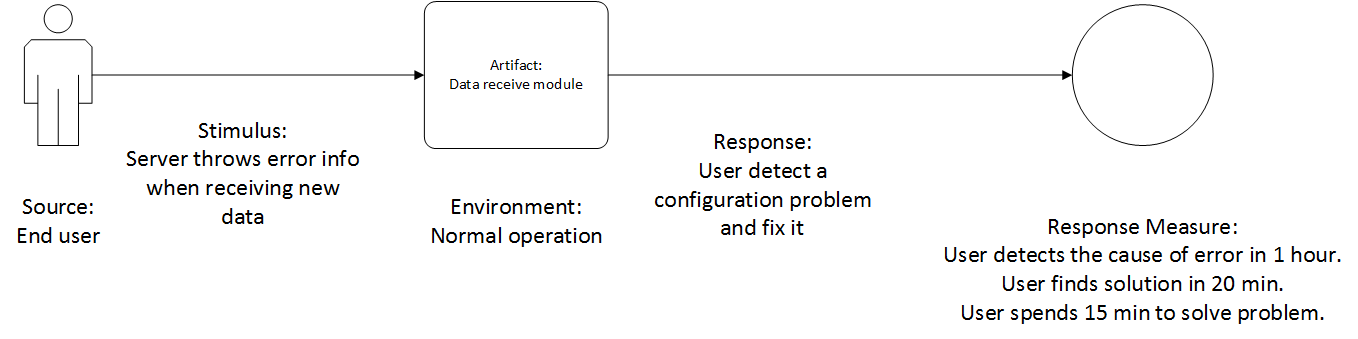


# Supportability

1. General scenario

|  |  |
| --- | --- |
| Portion of Scenario | Possible Values |
| Source | End user, technical support staff |
| Stimulus | Maintainer also need to install, configure and upgrade the program. Maintainer needs to identify and resolve issues when the program works incorrectly. |
| Artifact | monitoring component, logging component |
| Environment | installation, upgrading, normal operation |
| Response | Log the fault, together with global variables and execution path |
| Response Measure | * Time to find cause of a problem * Time to find solution of the problem in document * Time to actually solve the problem |

2. Concrete scenario



# Internal or External

In my opinion, internal quality attributes can be measured from development view and external quality attributes can be measured from product view. As a result, reliability, supportability, recoverability can be classified as external quality attributes, reusability and maintainability can be classified as internal quality attributes.