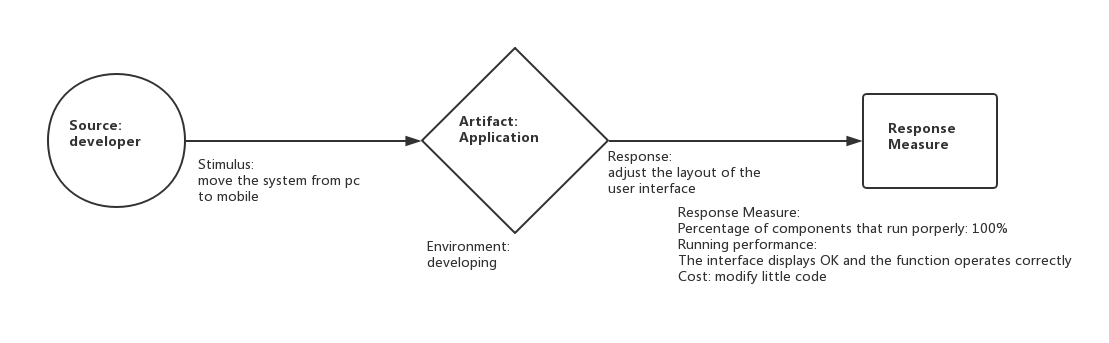
# Assignment 1

## Portability

### General Scenario

|  |  |
| --- | --- |
| Portion of Scenario | Possible Values |
| Source | developer, user |
| Stimulus | Move the system to a different terminal to run |
| Artifact | Application or component |
| Environment | developing, test, runtime |
| Response | adapt to different terminal display environment  UI to make responsive adjustments  The system in different environments should be able to run all the normal functions |
| Response Measure | Cost(manpower, resources, time)  Percentage of components that can run properly after portability  The error and crash rate after portability  Running performance |

### Concrete Scenario

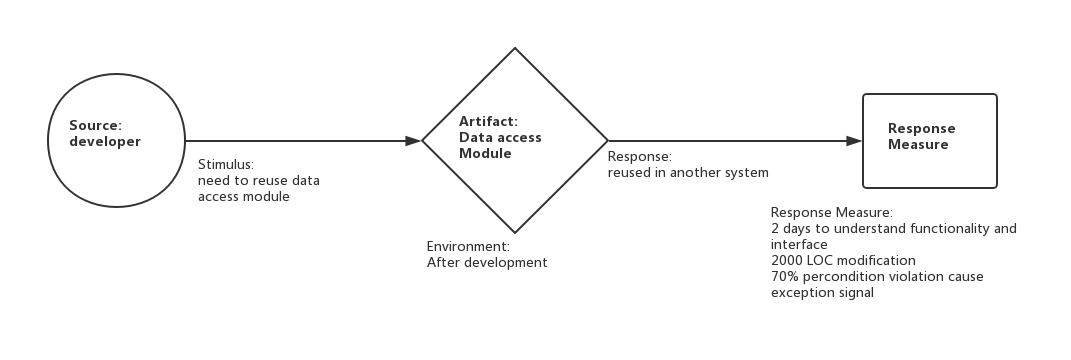


## Reusability

### General Scenario

|  |  |
| --- | --- |
| Portion of Scenario | Possible Values |
| Source | Software, people |
| Stimulus | Have partial of an existing program used in another program |
| Artifact | Framework or component that is going to be reused |
| Environment | After development |
| Response | Component is reused in another system |
| Response Measure | Time to understand the functionality of a component.  Modification needed to adapt one component to the specific functional requirement in a new system.  Proportion of precondition violation get handled by exception signaling. |

### Concrete Scenario



### Tactics for reusability

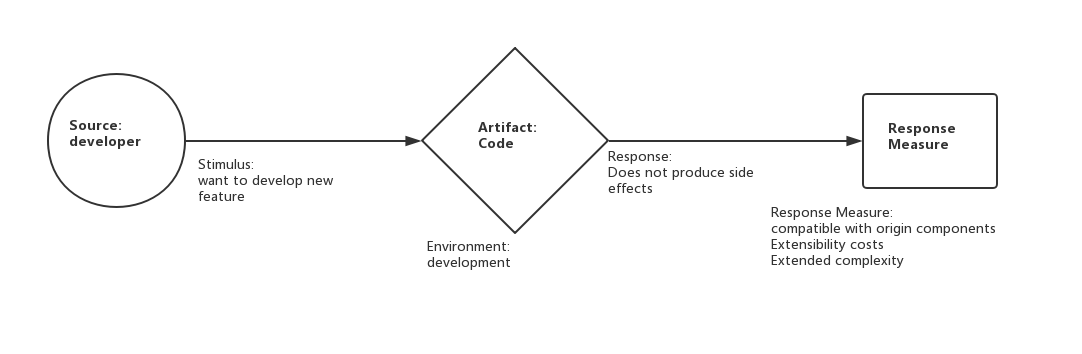
Modularization: Modularization is the process of encapsulating complexity of related classes and exposing their abstract interface. After modularization, program is divided into components, which have high intra-component cohesion and low inter-component coupling. Components have relatively simple and independent function. They can be composed to perform complex tasks. So they are quite suitable for code reuse.

## Extensibility

### General Scenario

|  |  |
| --- | --- |
| Portion of Scenario | Possible Values |
| Source | Developer, System Administrator, User |
| Stimulus | On the basis of the origin system, add new features and plugins  Add new quality attributes  Expand capacity |
| Artifact | System user interface, source code, interface, component |
| Environment | Runtime, build, design |
| Response | Find the location of the system needs to be extended, without affecting the expansion of other functions, the expansion of the test, the deployment of the expansion. Ensures the availability of extensions. |
| Response Measure | Cost, funds, Complexity  The impact of the original system components  Compatibility |

### Concrete Scenario

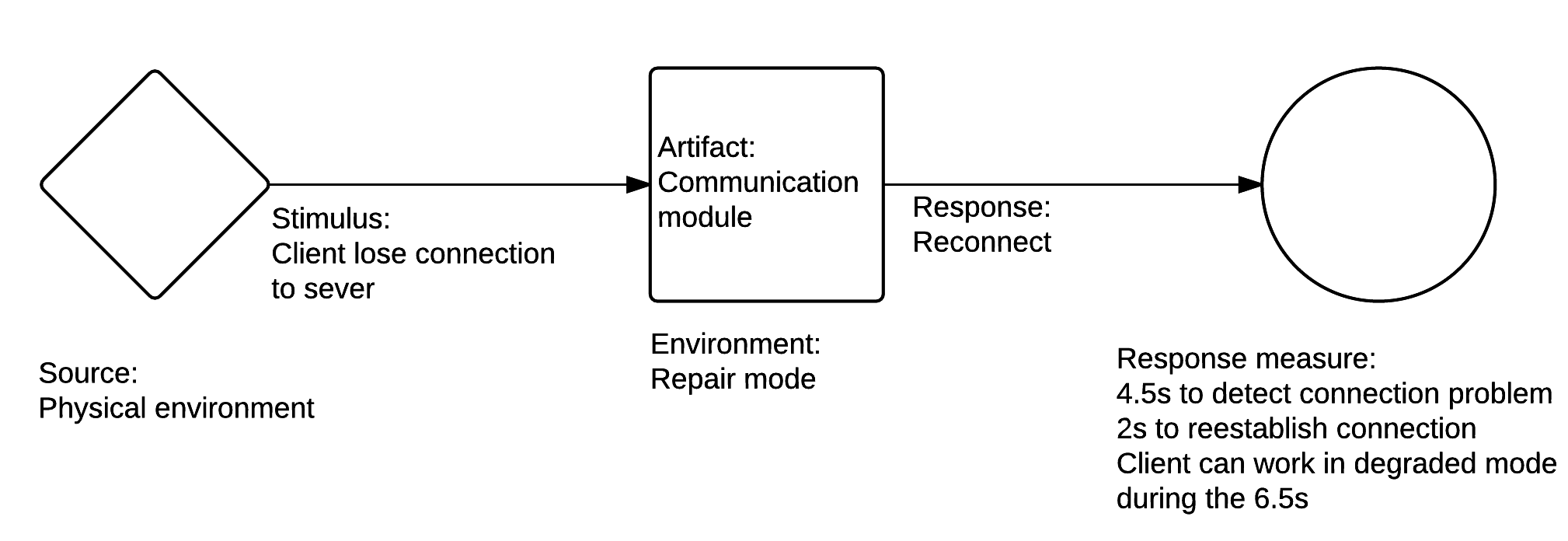


## Recoverability

### General Scenario

|  |  |
| --- | --- |
| Portion of Scenario | Possible Values |
| Source | Software, physical infrastructure, physical environment, people |
| Stimulus | Process crash, power off, hardware failure |
| Artifact | Process, persistent storage |
| Environment | Overloaded operation, degraded operation |
| Response | Log the fault  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 |

### Concrete Scenario



### Tactics for reusability

Replication: Replication can be used on server or database, usually with a master/slave relationship between the primary and the copy. Primary responds to inform the copy of state updates it must make. When the primary is down, the copy takes over and continues to provide service.

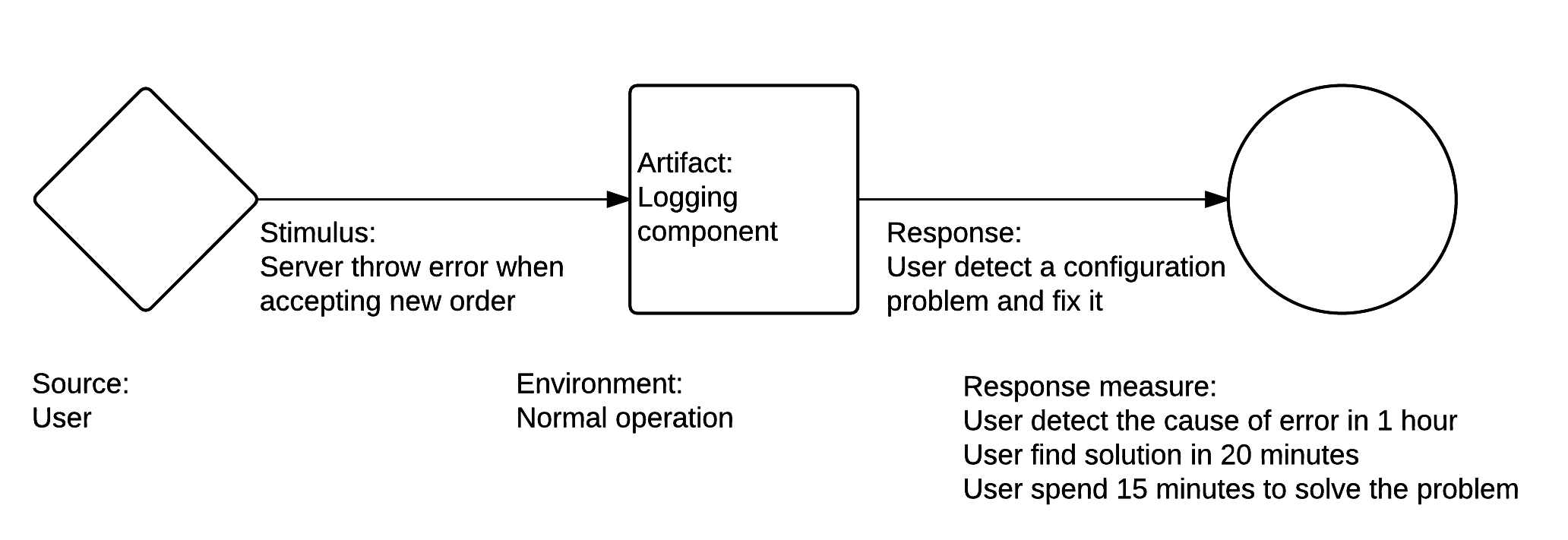
Checkpoint and operation log: Checkpoint is periodical dumping of consistent state. Operation log records important operation performed by system, typically data updating operation. When a failure occurs, system loads latest checkpoint and replays operation log to restore to the state before failure.

## Supportability

### 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 |

### Concrete Scenario



### Tactics for reusability

Graceful degradation: Maintain limited function when part of the software failed to work properly due to hardware failure or exhausted resource. Degradation allows recovery from exception without intervention by technical support staff.

Event logging: Event logging provides system administrators with information useful for diagnostics and auditing. Log high-level information, especially failure information.