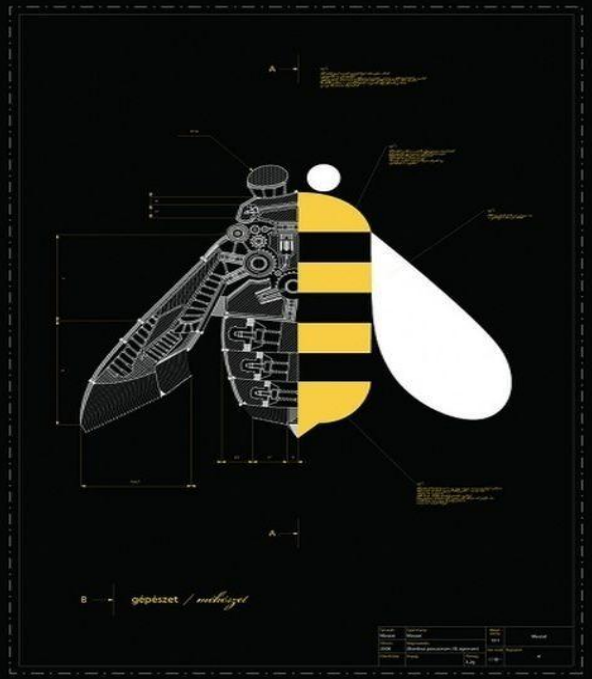
SYSTEM SOFTWARE FUSE



PRESENTED BY,

KARTHIK SIBI.M

PROBLEM STATEMENT

- In all technologies, the main base is process. When the process is affected, it will also affect the software so, the hardware also downed. Now software fuse helps to recover.

PROBLEM PREDEFINED

- Arthurx.sys(problem).
- CPU bottle neck.
- Memory leak.
- Multitasking.
- Defragmenting files.
- Outdated softwares.

To solve those problems they have some predictions,

- We have to include some additional hardwares like, RAM,Hard- disks Etc.,
- Doesn’t provide huge Environmental supporting.
- Cost is too expensive.
- Even a firewall rules fails.

Better performance of FUSE

- If we use this software we can save a costly system.
- It will freezes the unwanted processes.
- It will controls the overloaded processes and outdated processes.
- Saves all hardwares and softwares and prevents system heating.
- Intimation is provided by our software.
- After using our software, system will be fast, avoids hanging and safe.

OBJECTIVE OF FUSE PERFORMANCE

How does our solution is effective towards the problem,

- It doesn’t require external hardwares.
- Cost Effectiveness is less.
- Provides high Environmental Support.
- It can even rectify what firewall cannot.
- It will controls all process mainly unwanted processes and maintains the process which affects the system.

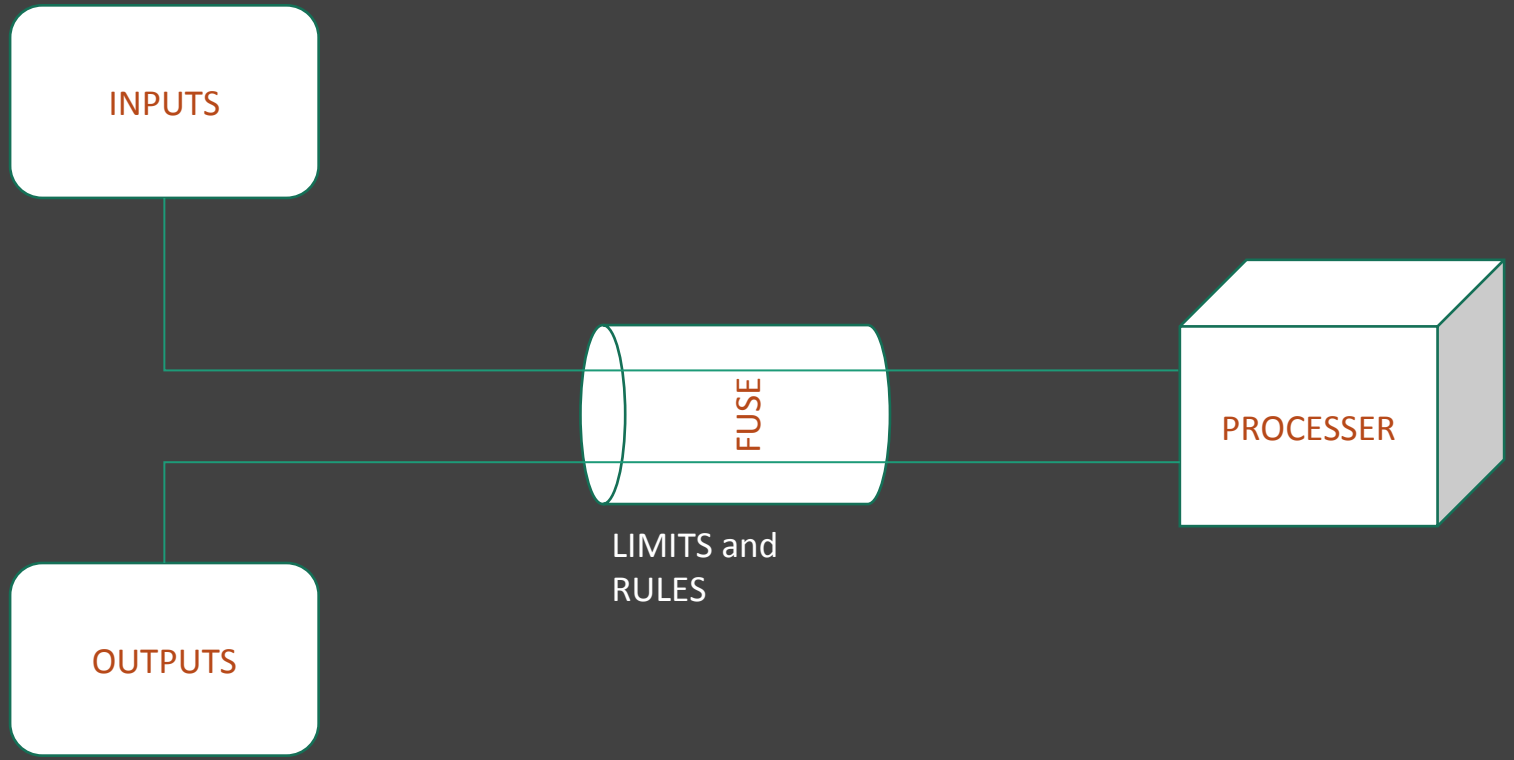
LIFE CYCLE MODEL

- We had chosen a **Agile Model**, here the changes are always welcomed at anytime in our software modules.
- This model is the combination iterative and incremental process model with duration of two weeks to two months.
- We have to work simultaneously on planning requirements, analysis, design, coding and testing.
- This model break the model into small incremental builds.

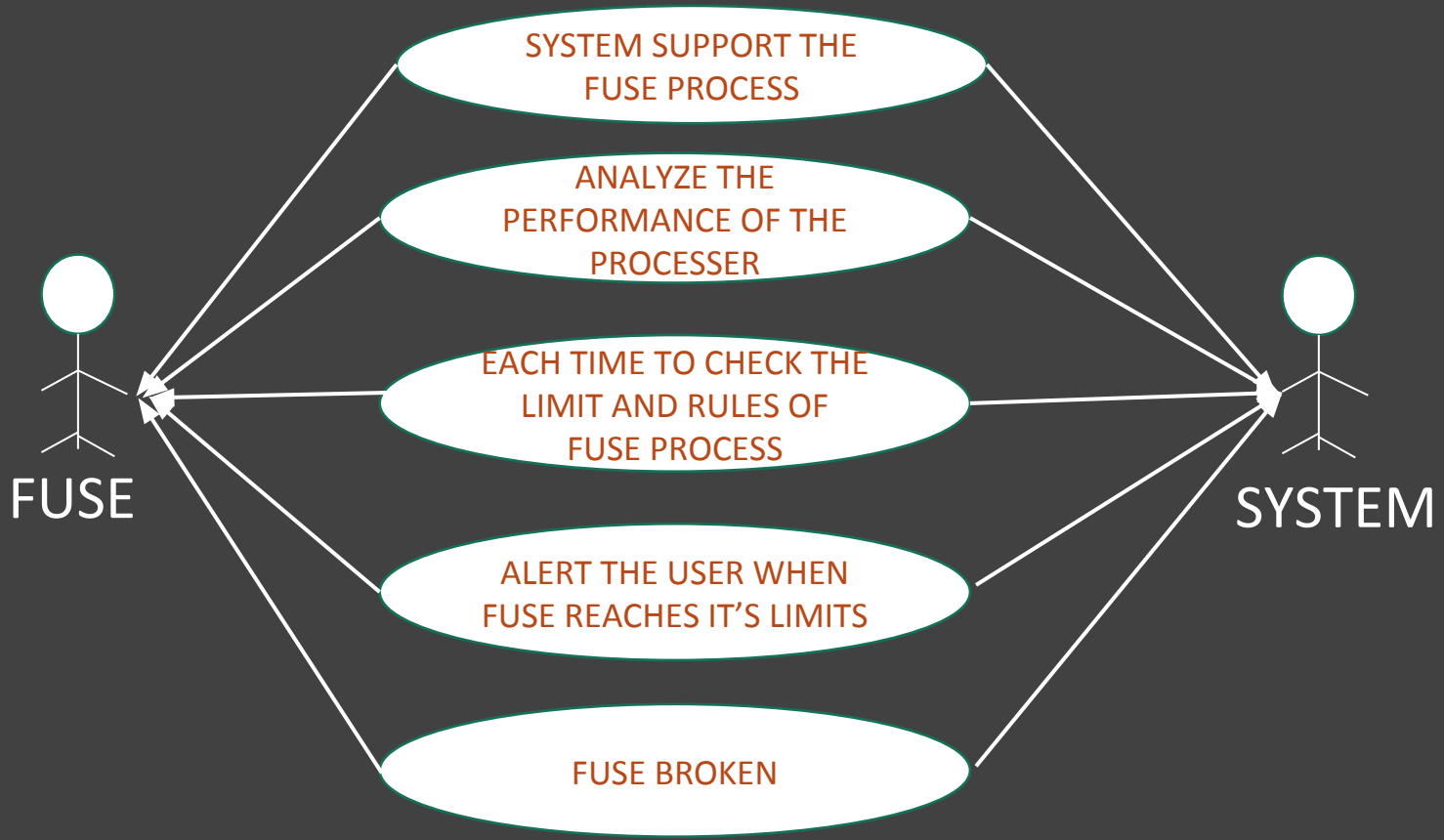
Why we Choose AGILE MODEL?

- Our use cases only supports agile model, because of certain changes .
- Highly Interdependent.
- User Colloboration.
- Large scale, offshore and distributed.
- Quality Assurance.

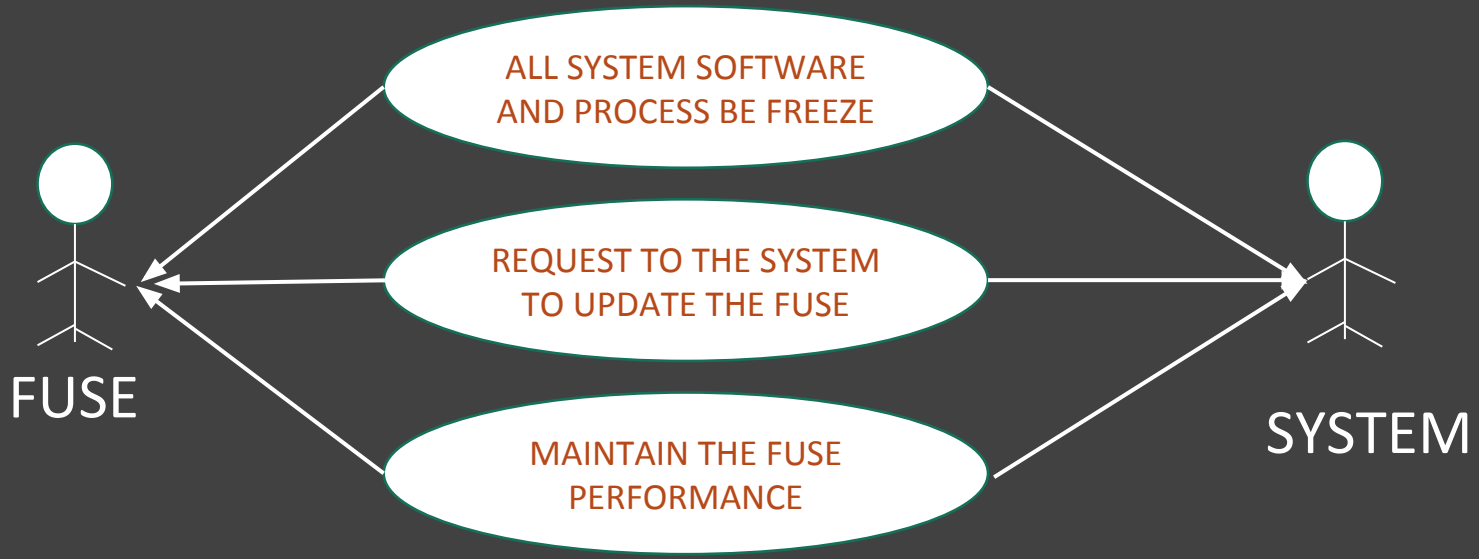
Destination of FUSE



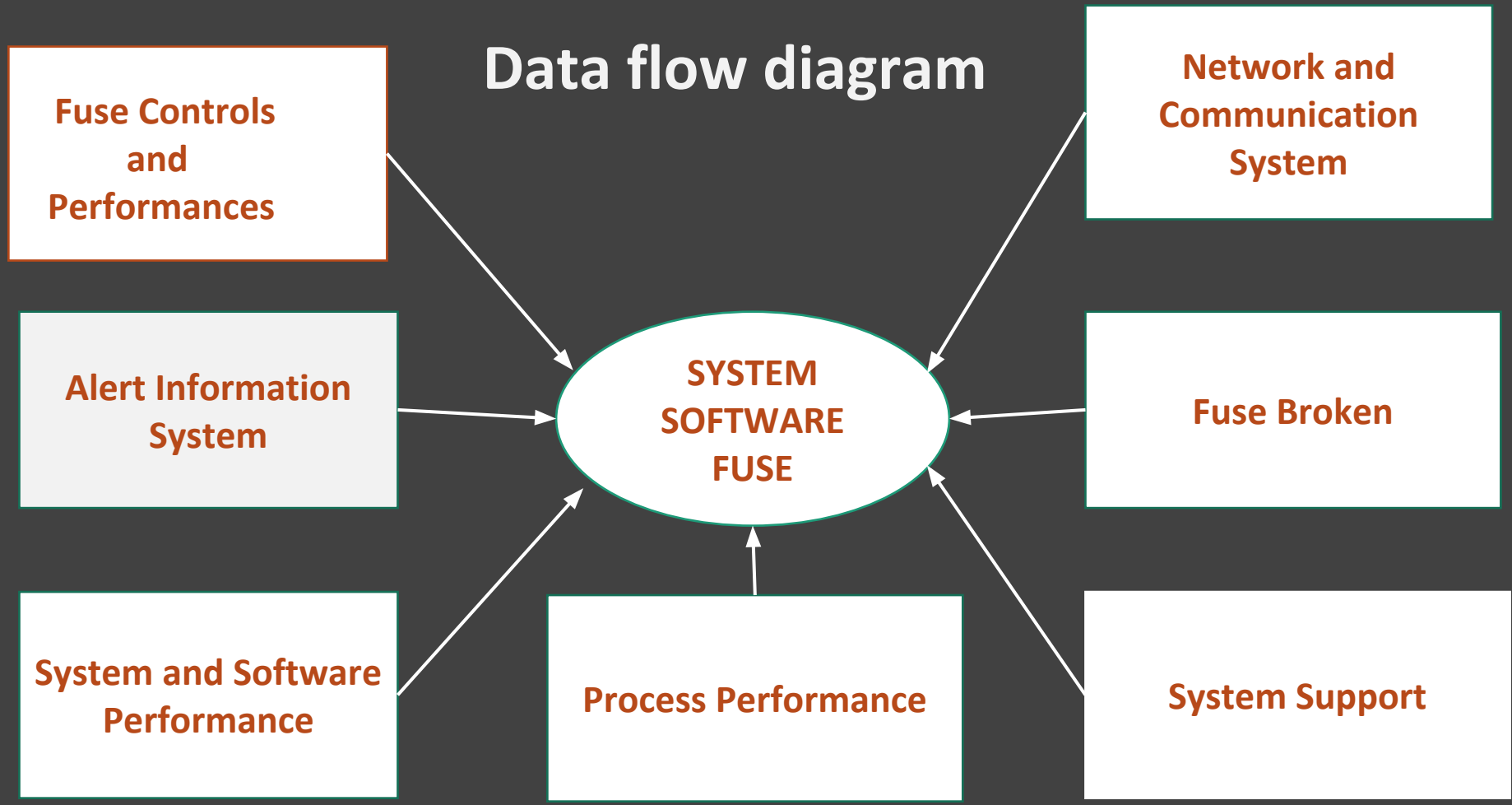
BEFORE FUSE PERFORMANCE



AFTER FUSE PERFORMANCE



Data flow diagram



Database Table Design

Process History
Process Id
Process name
Process Signature
Load performance
Process destination
Process type
Time for running

Cost Estimation

- It includes 14 parameters to calculate cost for our software such as reliable backup, communication, processing function, performance, run at existence, online data entry, multiple screen, master file, i/p o/p are complex, internal processing complexity, code is reusable, conversion, installation, easy of use.
- Formula:

  FP=count total*[0.65+0.01* ∑(fi)]

  =50*[0.65+0.01*55]

  = 60.

Cost Estimation Models

- E=5.2*(KLOC) –Walson-Felix Model
- E=5.5+0.73*(KLOC) – Bailey Basili Model.
- E=3.2*(KLOC) – Boehm Model.
- E=5.288*(KLOC) – Doty Model for KLOC >9.
- E=91.4+(0.355*FP) –Alberecht Model.
- E= -37+0.96FP – Small Project.

Test Cases

- Verify, whether the system accepts fuse as its software
- Verify, whether the fuse analyse processor performance.
- Check/test whether, the limit of fuse is intimated to us by the message.
- Check whether, we can use a new fuse after the burnt of existing fuse.


Thank You!