**Software Requirements Specification**

**On**

**Automated Patch Mechanism for MR Software Components**

**SUBMITTED BY**

**PRAVEEN K 1MS12CS075**



**M. S. Ramaiah Institute of Technology**

**(Autonomous Institute, Affiliated to VTU)**

**BANGALORE-560054**

**Department of Computer Science & Engineering**

***Under the guidance of***

|  |  |  |
| --- | --- | --- |
| **Sriranga Natha**  **System Architect, PIC ADI Magnetic Resonance**  **Philips Innovation Campus, Bangalore** | **&** | **Sini Anna Alex**  **Assistant Professor**  **Department of Computer Science & Engineering** |

1. **Project Overview**

The aim of this project is to successfully develop an automated patch mechanism for MR software components. This project concerns itself with the workflow aspect of the complex MR software. The patch mechanism’s function will be to fix the logical issues, automatically search for dependent DLLs and ensure compatibility is maintained, and then patch the fix to the system without the need to build the entire system again, or to stop all system processes, except for the dependent ones.

1. **External Interface Requirements**
   1. *Hardware Interfaces:*

The application will run on MR Environment configured test systems.

* 1. *Software Interfaces:*

Inputs

The application takes as argument the files which have been modified.

Outputs

The application restarts only the patch impacted processes.

* 1. *Communication Interfaces:*

The application requires intranet connection to execute several functions such as searching for the dependent processes and DLLs and building the DLLs.

1. **Functional Requirements**
   1. *Identify and build the dependent DLLs on the developer’s system:*

The application takes as argument a list of modified files and the patch location. It finds the DLLs that use the modified file and builds them and adds them to the patch directory.

* 1. *Identify and stop the dependent processes on the remote test system:*

The processes that need to be patched are identified and only these processes are stopped so that the patching can be done on them.

* 1. *Target all the DLLs to their respective locations on the remote test system:*

The built DLLs are copied to their respective locations on the remote test system.

* 1. *Start only the impacted processes:*

The patched processes are restarted on the remote test system.

1. **Software System Attributes**

*4.1* *Reliability:*

The application will meet all of the functional requirements without any unexpected behavior.

*4.2* *Availability:*

The application will be available on demand provided the user has the necessary interfaces for the proper functioning of the application.

*4.3* *Security:*

The application only patches the necessary processes and will not affect the proper functioning of other applications in the user’s system.

*4.4* *Portability:*

The application is designed to run on Microsoft Windows operating system.

*4.5* *Maintainability:*

The application will be written clearly and concisely. The code will be well documented. Particular care will be taken to design the project modularly to ensure that maintenance is easy.

1. **Performance Requirements**
   1. *Real-Time:*

The application requires latest build metadata for current development stream.

* 1. *System Resource Consumption:*

Resource consumption of this application will not reach an amount that affects the normal processing of user’s system.

1. **Design Constraints**

The design constraints that will affect the design of this application are the schedule and the performance.

The project must be completed on schedule and the performance of the application must be very high. Parallelism will be exploited to increase the performance.

1. **Other Requirements(if any)**

Knowledge of C# programming language, domain knowledge of MR are the additional requirements required.