**Project SYNOPSIS**

**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** |

**Details of the organization**

|  |  |
| --- | --- |
| **Details of the organization**  **(with postal address):** | **Philips Innovation Campus  Manyata Tech Park,  Nagavara,  Bangalore - 560045  India** |
| **Name of Guide with contact details and email address:** | **Sriranga Natha**  **+91 9886623382**  **sriranga.natha@philips.com** |
| **Date of commencement of the project:** | **12th January, 2016** |

1. **Introduction**

Magnetic Resonance Imaging (MRI) is one of the youngest and most innovative imaging technologies and is growing in applications but also in complexity of the system. MR has a large software presence in the Philips Innovation Campus, Bangalore, producing state of the art software driving MR workflow, post-processing, reconstruction and serviceability.

1. **Objective**

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.

1. **Need for the project**

An essential part of the Philips development framework is its association with Research hospital sites. To facilitate co-development in such sites, MR offers a software patching mechanism that enables the sites to explore and test new pulse sequence techniques. As a continuous improvement, MR is building an intuitive and fast Software patching framework with a long term view of single-click enabling of patches.

In the MR systems being used currently, the main activity involves identification of the MR applications that need to be restarted to activate this patch. In these systems, if an issue is detected with a particular DLL, the user, after rectifying the issue, needs to manually search all over the remote repository for dependent DLLs, check for the compatibility, and then rebuild the entire system, which usually takes around 6hours, and then install the system again, which takes some more hours. In addition, all system processes need to be suspended in order to check for the dependent processes (executables).

1. **Proposed Solution**

To resolve the above mentioned problem, an automated patch mechanism is proposed whose 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. Implementation of this patch mechanism shall aid in improving the efficiency of the system along with a reduction in the user effort.

1. **Methodology**

The project shall be carried out as follows:

* Identify and build the dependent DLLs on the developer’s system (local system)
* Identify and stop the dependent processes on the remote test system
* Target all the DLLs to their respective locations on the remote test system
* Start only the impacted processes