Contents

[Introduction 2](#_Toc441759310)

[Vision and Goals of the Project: 2](#_Toc441759311)

[Users of the Project: 3](#_Toc441759312)

[Scope and Features of the Project: 3](#_Toc441759313)

[Solution Concept 3](#_Toc441759314)

[Architecture of the Project 3](#_Toc441759315)

[Minimum Acceptance criteria: 4](#_Toc441759316)

# Introduction

Releasing software with a Continuous Integration (CI) Pipeline can greatly increase the speed with which code gets released. CI Pipelines can largely replace manual processes of releasing. When it comes to the content of a release however, most tools focus on relating each build to a set of changes in source control. This is great for developers to determine the source of a bug in a particular build, but the source control does not hold much value when it comes to communicating the business content of the build. When a build gets deployed, the business wants to know what the content of the build is, so that they know what to test, as well as what can be released to production. This often means a manual step in the CI process is necessary: the developer must look up the work items in the work item tracking software, and enter them manually into the release tracking software. The goal of this project is to eliminate that manual step of entering release notes by integrating the build pipeline with Work Item Tracking software.

# Vision and Goals of the Project:

“Automated Release Notes” involves the development of a Java plugin for Teamcity (Build Pipeline) which would integrate it with Work Item Tracking software(in this case Visual Studio) for generating a useful text file containing work item information which would be used by Business/Quality Analysts(QA) for testing purposes. This would eliminate the manual step of creating release notes by the developers while checking in their changes in source control software.

High-level goals of Automated Release Notes include:

* Providing an efficient tool to eliminate the manual step of entering release notes integrating the build pipeline (Teamcity) with work item tracking software (Visual Studio).
* Providing QA with a better understanding of the latest build and test the critical content of the build.
* Eliminating manual errors committed by developers while manually creating the release notes.
* Enabling the improvement over the efficiency of releasing software with a continuous integration

# Users of the Project:

Automated Release Notes will mainly be used by software developers to create automated release notes notifying the Quality Analysts regarding description of the recently checked in work items for quality and testing purposes. Both – Developers and Quality Analysts will be the beneficial users of this product.

# Scope and Features of the Project:

* An efficient and user friendly interface within Teamcity for automatic release notes generation corresponding to each check in.
* This approach should be configurable for multiple deployment servers such as Octopus Deploy.
* The end-product could be configurable for multiple tracking servers such as VSO
* Scalability: can scale to large number of users, projects and services

# Solution Concept

Described below is the high level outline of the solution.

## Architecture of the Project

Below is the description of the system components that are the building blocks of the architectural design:

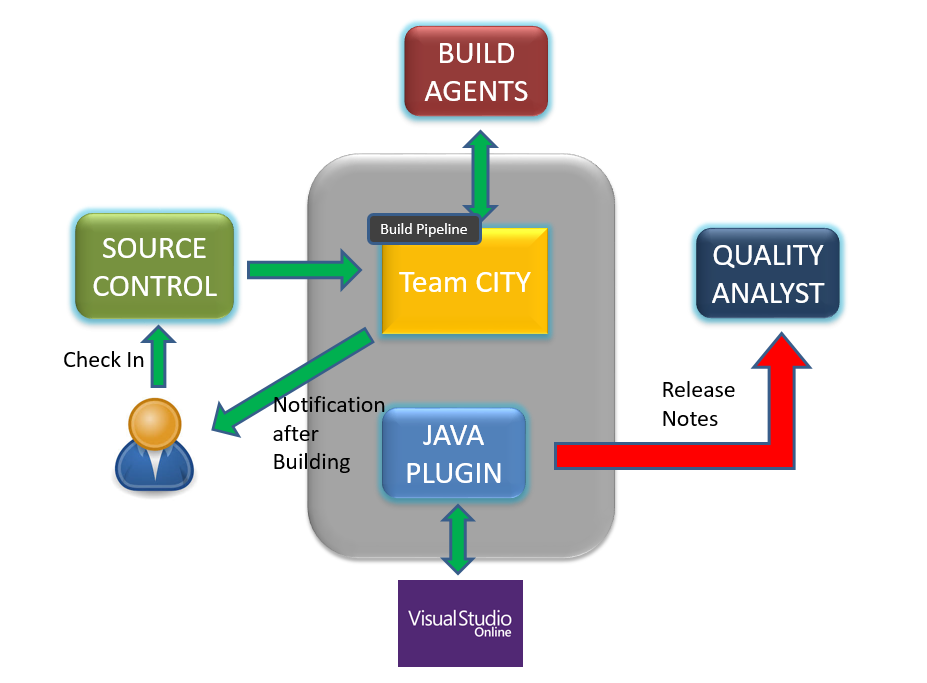
**Build Pipeline (Team City):** Team City is an enterprise of JetBrains which is a Java based platform that implements Continuous Integration. It has the following features:

* The commits are [Gated](https://en.wikipedia.org/wiki/Gated_Commit)  (this feature prevents developers from breaking sources in a [version control system](https://en.wikipedia.org/wiki/Revision_control) done by running the build remotely for local changes prior to commit).
* Build [Grid](https://en.wikipedia.org/wiki/Grid_computing). Allows running multiple builds and tests under different platforms and environments simultaneously.
* Integrated [code coverage](https://en.wikipedia.org/wiki/Code_coverage), [inspections](https://en.wikipedia.org/wiki/Software_inspection) and [duplicates](https://en.wikipedia.org/wiki/Duplicate_code) search.
* Integration with [IDEs](https://en.wikipedia.org/wiki/Integrated_development_environment): [Eclipse](https://en.wikipedia.org/wiki/Eclipse_(software)), [IntelliJ IDEA](https://en.wikipedia.org/wiki/IntelliJ_IDEA" \o "IntelliJ IDEA), [Visual Studio](https://en.wikipedia.org/wiki/Visual_studio).
* Platforms supported: [Java](https://en.wikipedia.org/wiki/Java_(programming_language)), [.NET](https://en.wikipedia.org/wiki/.NET_Framework) and [Ruby](https://en.wikipedia.org/wiki/Ruby_(programming_language)).

**Source Control:**

**Work Item Tracking Software (Visual Studio):**

**Java Plugin: Needs to be discussed**



**Architectural Design**

# Minimum Acceptance criteria:

Minimum acceptance criteria is a TeamCity Plugin in the form of a Java API that could integrate the Teamcity and Visual Studio to generate a text file containing the work item description of the checked in code which could serve as the release notes for the Quality Analysts. Ideally the end-product will be configurable for Visual Studio Team Services/Visual Studio Online (VSO).

# Release Planning

Needs to be discussed