

Parabuild Demo
at
Bay XP

Slava Imeshev
vimeshev@viewtier.com
www.viewtier.com

Agenda

- Introduction
- Continuous Integration
- Best practices
- Parabuild introduction
- Continuous Integration for Junit in 10 minutes
- Q&A

Introduction

Slava Imeshev

- XP practitioner
- President of Viewtier Systems, makers of Parabuild

Continuous Integration

1. Important eXtreme Programming practice
 - Check in your changes often
 - Don't check in hourly
 - Check in when you are ready
 - Long check in intervals may mean that you are digging yourself down
2. Supporting Infrastructure
 - Version control system (VCS)
 - Versioned storage as in time machine
 - Continuous Integration server
 - Watches for changes and notifies when the build is broken
 - Lets fix the breakage immediately

Benefits

- Most of the software projects fail because they:
 - Miss deadlines
 - Cannot stay in budget
 - Unable to meet the requirements
- Continuous Integration allows to reduce risk of failure by:
 - Reducing slipping of schedule caused by code base breakage
 - Increasing team morale and helping to maintain capable engineers on board
 - Reducing development time and hitting the market before competitors

Introducing Continuous Integration

- Introducing Continuous Integration to organization, with all benefits it brings, may face some challenges. Here are the ways to address some them:
 1. Release Engineering AKA “Mr. Nightly Build” resists
 - RE has to become an owner of the infrastructure
 - Show that this is a relief, not a burden (lesser chances of daily build breakage or practically unbreakable ones with Parabuild)
 - Get them production-grade tools
 2. Builds are still constantly broken
 - Educate your team to respond to breakage immediately
 - See, I have broken the build, fixed it and no one even noticed the breakage.
 - Follow best practices in the next section

Best Practices

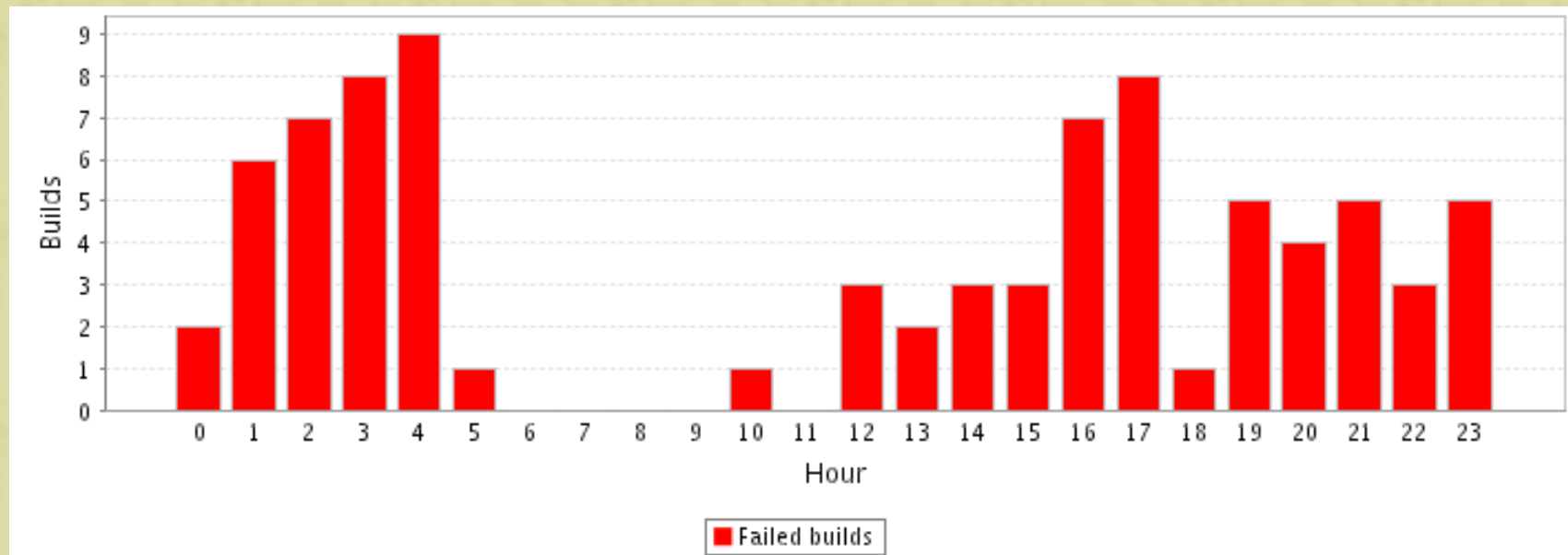
Infrastructure

- Get a **production-grade software build management system**. A build server that need CPR every hour is no good.
- Dedicate the **fastest hardware money can buy**. Faster hardware produces quicker response about quality of changes.
- Make sure your **product can be build from command line**. It should be able to build in unattended, batch mode.
- Build scripts and dependencies are **under version control**. Everyone should be able to reproduce build breakage at given time.

People

- Remember - **you don't break the build if you don't do anything**. It OK to break a build from time to time. It is NOT OK to leave it broken.
- **Always claim build breakage**. People will appreciate knowing that the breakage is being taken care about, and will do same for you.
- **Avoid build breakage patterns**: Five O'clock Checkin, Spoiled Fruit and Small Change. This alone will reduce build breakage by 50%.
- Establish **fun, non-insulting ritual for hailing build breakers**. Buying donuts to the team works well!

5PM Check In



Neat Build and Release Management System

- Continuous Integration
- Scheduled builds
- Manual builds
- Multiplatform builds
- Parallel builds
- Version control and issue tracking support
- Web user interface
- Searchable build logs and results archive
- Build telemetry
- Automatic merges
- Group-based security and LDAP integration
- Ease of installation and administration
- Four years old. Current version is 3.2. Modified XP approach.

Killer Feature

Problem Of Continuous Integration

1. Notifications regarding build breakage are perfectly ignorable
2. Daily or nightly (scheduled) builds are still broken

Parabuild's Solution

- Innovative patent-pending technology
- Provides practically unbreakable scheduled builds
 - Scheduled builds are successful even if the head of the code base (AKA latest) is broken

Continuous Integration and Daily Builds for JUnit

- Install Parabuild
- Set up Continuous Integration
- Set up daily build
- 10 minutes from zero to done!

Q&A

Thank you.

Parabuild downloads:

<http://www.viewtier.com/downloads.htm>