# Parasoft DTP Documentation

About Parasoft Development Testing Platform (DTP)

Parasoft Development Testing Platform (DTP) eliminates the business risk associated with faulty software, accelerates delivery, and facilitates continuous process improvement. DTP monitors and measures the application of quality practices, such as static analysis, unit testing, coverage analysis, runtime error detection, etc. Data generated throughout the SDLC is collected, correlated, and analyzed in order to deliver intelligent, actionable findings that enable you to focus on the impact of changed code and demonstrate full compliance traceability. DTP applies practices according to your development policies, unobtrusively bridging the gap between business expectations and dev/test activities.

Key Features

- **Integration with open source analysis tools and testing frameworks**: Download extensions for DTP from the Parasoft Marketplace that enable you to integrate with popular open source tools and systems. More at https://marketplace.parasoft.com.

- **DTP Engines for Java, C/C++, and .NET**: Automate the collection of development testing data by configuring and deploying DTP's native code analysis and test execution tools into your infrastructure.

- **Interactive explorer views**: Dig deeper into static analysis violations, metrics analysis, test results, and coverage through detailed explorer views that show findings in context of the source code, while enabling you to manually execute remediation paths, mark tests to be rerun, and more.

- **Intelligent metadata**: Flag violations for remediation and tests to re-run or assign actions to improve the efficiency of the development organization and help bridge the gap between technical and business risk.

- **Customizable Web-based interface**: Leverage built-in or custom dashboards, widgets, and reports for at-a-glance information about the status of development activities.

- **Robust REST and Java APIs**: Access defects, requirements, tasks, and other data to integrate with other tools and applications that extend DTP functionality.

- **Access code from source control directly from the browser**: Leverage out-of-the-box integrations with commonly-used SCMs, such as Git, Subversion, AccuRev, Mercurial, Microsoft TFS, and Perforce, or publish sources directly from analysis tools.

- **Integration with requirements and defect tracking systems**: Create links across systems within your development infrastructure, such as TFS, JIRA, Bugzilla, and Rational DOORS to enable customizable DTP workflows. Tests can also be linked via reference numbers.

DTP Workflow
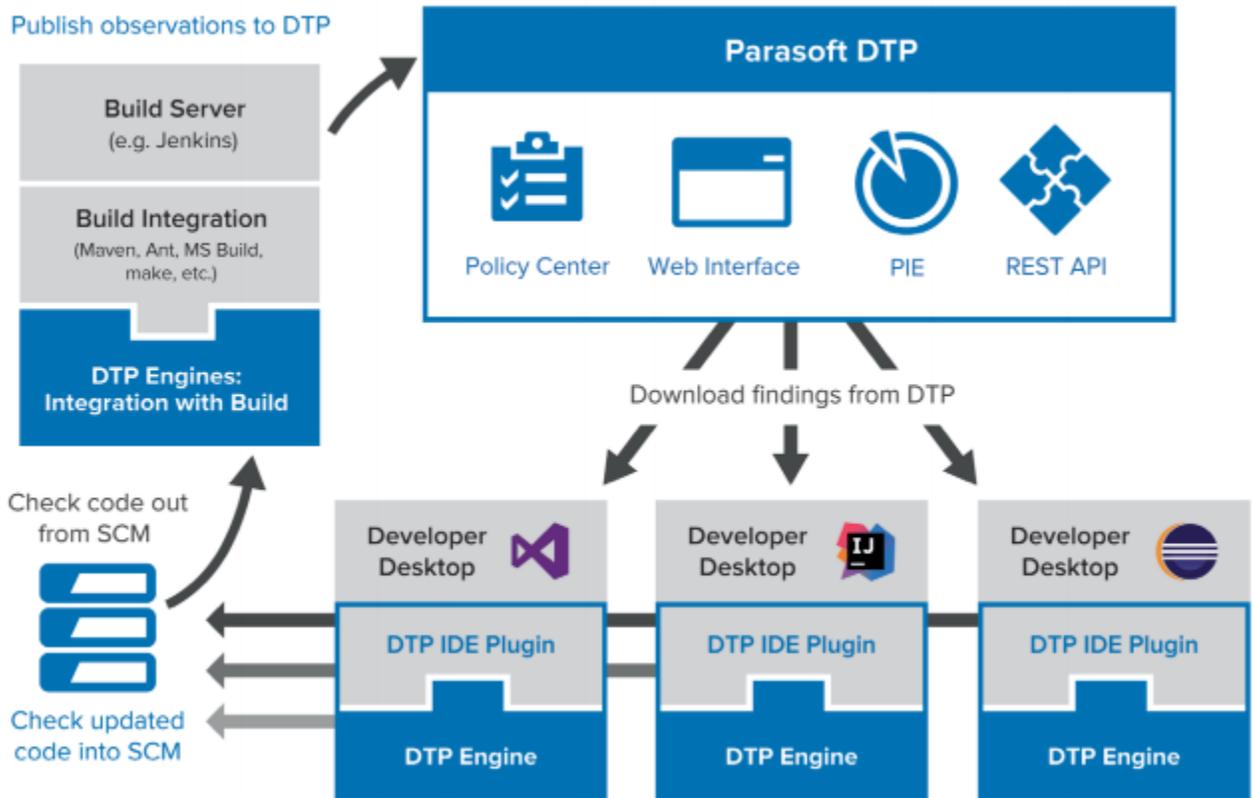
Associated Modules

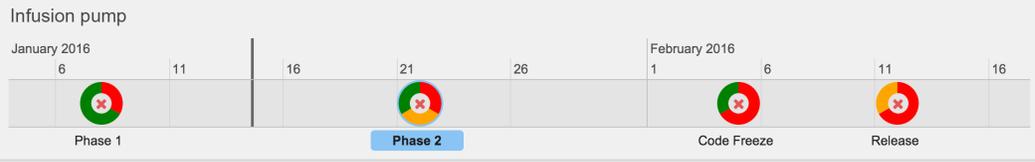### Process Intelligence Engine: Predictive Analytics for the SDLC

The award-winning Process Intelligence Engine (PIE) is a first-of-its-kind predictive analytics engine that identifies risky patterns buried deep in the data and delivers an actionable list of prioritized findings based on your policies. DTP collects the raw observations generated by disparate test and analysis tools throughout the SDLC so that PIE can apply advanced analysis techniques that highlight business risks. For example, DTP can collect coverage from unit testing, functional testing, and manual testing activities so that PIE can analyze and report true coverage. Coverage can also be combined with other metrics to identify application hotspots in areas of the application that present the most significant risk. Contact your Parasoft representative for additional information.
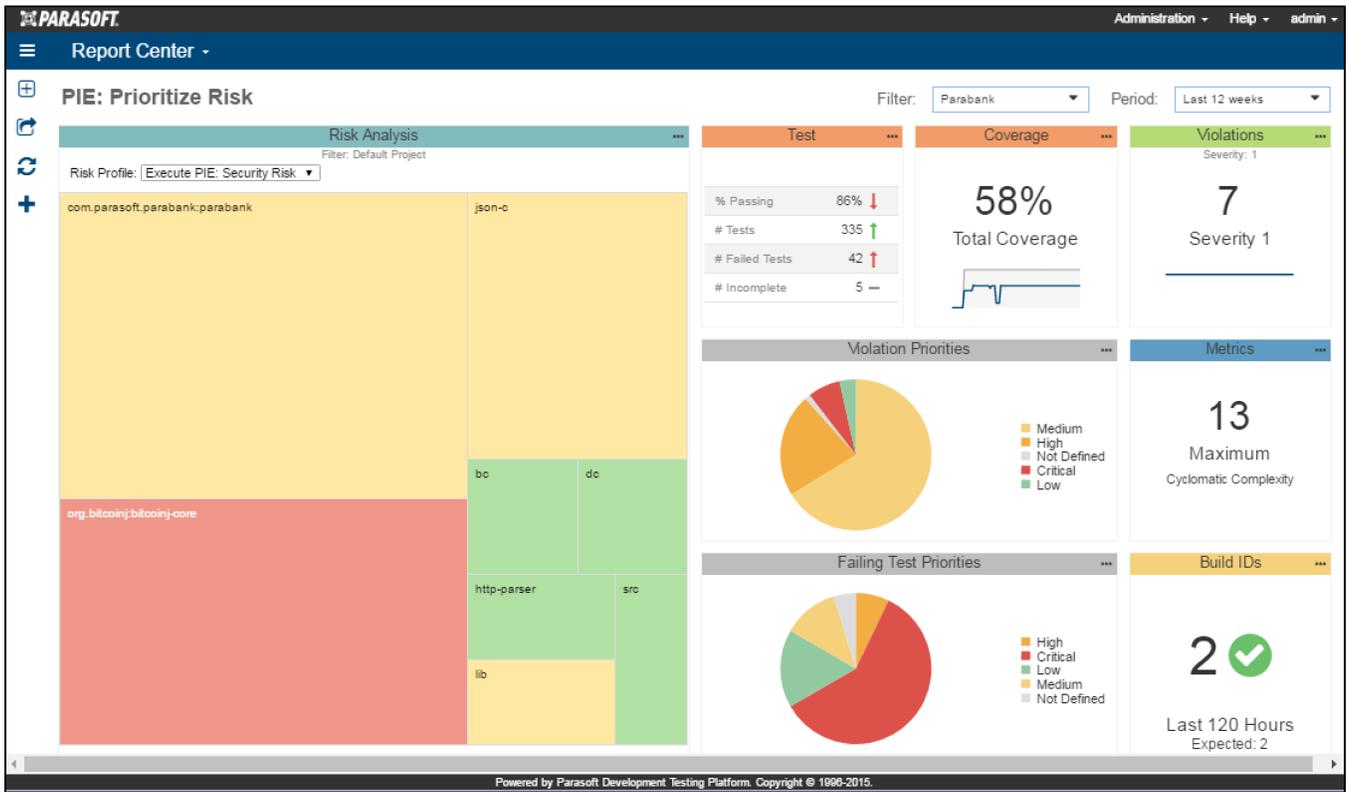
Policy Center: Create and Manage Development Policy and Quality Gates

Ensure that software is being developed according to business objectives by encapsulating expectations into policies and setting quality gates that enable you to clearly understand the risk of a release candidate. A policy can include tolerable levels of violations, defects, complexity, and other criteria. You can also designate how code should be analyzed in central test configurations and distribute them to your development teams to ensure consistency and maintainability. Contact your Parasoft representative for additional information.

Policy Center ▾

| Projects | Policies | Gates | Add Gate | Clone Gate |

All tests
Infusion pump
Glucose sensor
Defibrillator
Pacemaker

Infusion pump

January 2016 · · · · · · · · · · February 2016
6 · · · 11 · · · 16 · · · 21 · · · 26 · · · 1 · · · 6 · · · 11 · · · 16

Phase 1 · · · Phase 2 · · · Code Freeze · · · Release

Phase 2 · ✓ 1 · ! 1 · ✗ 1 · 🗑 Delete · Edit

1/22/2016 · 📅 · Filter: Infusion pump ▾

❌ **Agile Development** · Filter: Infusion pump ▾
⊗ 18 unfinished Development tasks.
⊗ 3 user stories in In Progress state.

✅ **Defect Prevention** · Filter: Infusion pump ▾

⚠️ **Test Regression** · Filter: Infusion pump ▾

   ⚠ **Regression Test**
   31 test failures.
   Select filter: SDM Platform ▾
   Select analysis type: Unit Test ▾
   ✓ # of test failures | % of test failures
   Success < 30 < Unstable < 50 < Error

   ⚠ **Regression Test**

Powered by Parasoft Development Testing Platform. Copyright © 1996-2015.