

# Parasoft Development Testing Platform

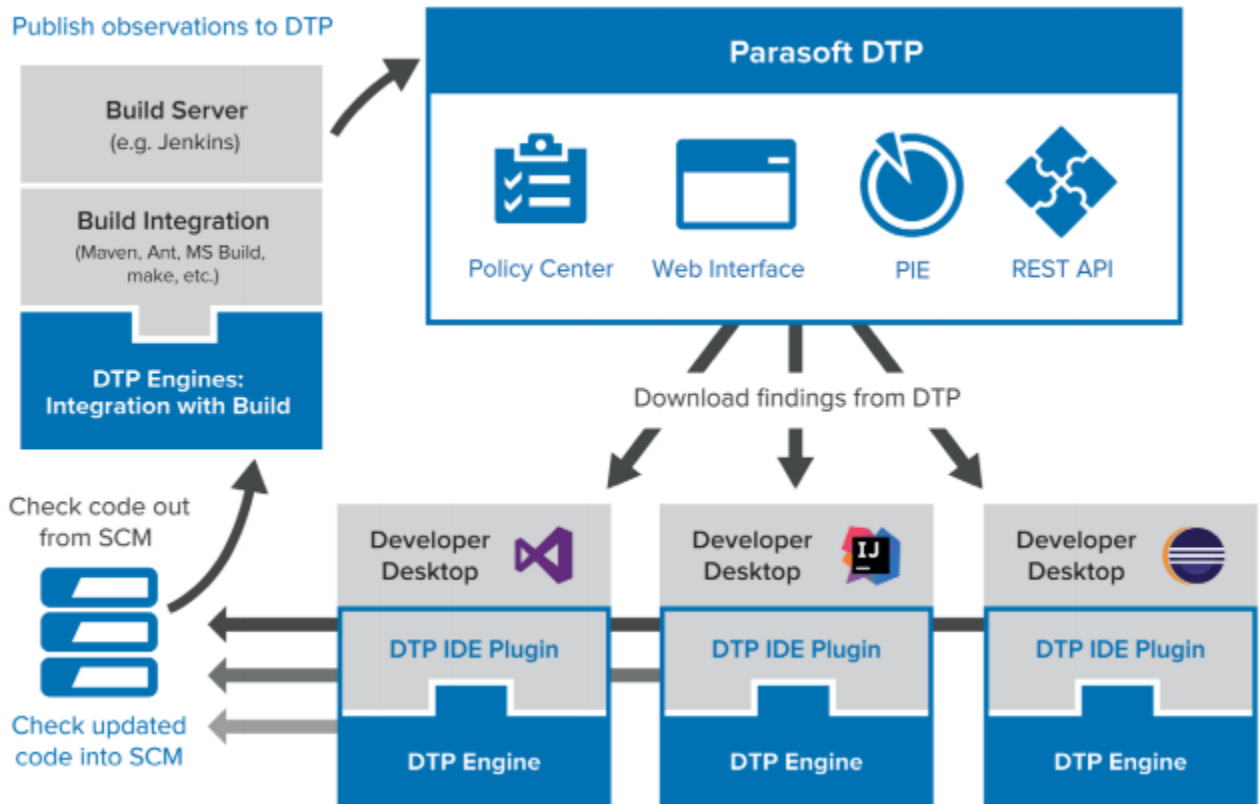
## About Parasoft Development Testing Platform (DTP)

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

### DTP Enterprise Pack

- Improve software quality and continuously improve development processes by rooting out risky patterns buried deep in the SDLC data.
- Increase DevTest productivity with prioritized, actionable findings that are only triggered when coding patterns violate your policies.
- Enhance your DevTest infrastructure with Parasoft Marketplace extensions for popular systems or leverage the open API to seamlessly integrate with any SDLC component.

### Extension Designer: Predictive Analytics for the SDLC










Extension Designer is the interface for creating and configuring logic flows that apply advanced analysis techniques to expose application hotspots. For example, DTP can collect coverage from unit testing, functional testing, and manual testing activities so that a Extension Designer flow 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.

**PARASOFT** Extension Designer ? admin

Services Model Profile **Configuration** Upload Artifact

Server Settings  
Proxy Settings  
SMTP Settings  
Marketplace Settings  
**Artifact Manager**

Powered by Parasoft Marketplace®

<input type="checkbox"/> Select All	Search Artifacts		
<input type="checkbox"/>		<b>Change Based Testing</b> — 2.0.0 <span style="background-color: #28a745; color: white; padding: 2px;">2.0.1 available</span> <a href="#">Documentation</a> <a href="#">Parasoft License</a> The Change Based Testing slice calculates change based testing metrics that can be displayed in multiple forms (pie chart, table report, and DTP test case explorer). It gathers all the files in a code base and analyzes the test cases associated with each file. If the file has been modified within the given date range and the test cases associated with that file have not been run since the change, then the test case "Action" will be set to "Retest." Otherwise, the test case "Action" will not be modified. <span style="background-color: #28a745; color: white; padding: 2px;">PIE Slice</span> <span style="background-color: #28a745; color: white; padding: 2px;">Marketplace</span>	 
<input type="checkbox"/>		<b>DTP Build Review Workflow</b> — 2.0.0 <a href="#">Documentation</a> <a href="#">Parasoft License</a> A "review" in Parasoft is a session in which a file from one build is compared to the same file in a different build. Reviews are commonly used to compare the current build with the most recent build. Each review is associated with "findings," which are annotations added by the reviewer. Build reviews enable developers and testers to call out potential risk in the application as the code evolves. Reviews are normally created manually in DTP, but the DTP Build Review workflow enables you to automate the creation of reviews, as well as specify a user who will receive automated email notifications about the reviews created. An email will automatically be sent when reviews are created, when findings are added, and when an open build review contains no open or in-progress findings. <span style="background-color: #ffc107; padding: 2px;">DTP Workflow</span> <span style="background-color: #28a745; color: white; padding: 2px;">Marketplace</span>	
<input type="checkbox"/>		<b>DTP File Based Licensing Report</b> — 2.0.0 <a href="#">Documentation</a> <a href="#">Parasoft License</a> This workflow artifact provides detailed information about your Parasoft DTP license consumption/usage. This helps you determine if the number of files being scanned for Static Analysis, Functional Testing, Unit Testing, or Metrics is close to reaching your licensed limit. <span style="background-color: #ffc107; padding: 2px;">DTP Workflow</span> <span style="background-color: #28a745; color: white; padding: 2px;">Marketplace</span>	
<input type="checkbox"/>		<b>Export DTP Explorer Data To CSV</b> — 2.0.0 <a href="#">Documentation</a> <a href="#">Parasoft License</a> This workflow accepts any parameter URL from DTP's explorers (Test Explorers and Violations Explorers) and passes its parameters to DTP REST API and the developer's tool to generate a CSV output. <small>Powered by Parasoft Development Testing Platform. Copyright © 1999-2017.</small>	

## 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.

- Infusion pump
- Glucose sensor
- Defibrillator
- Pacemaker

### Infusion pump



### Phase 2

1/22/2016

Filter: Infusion pump

Delete Edit



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