# PreEmptive Analytics Sample Application Overview

The following sample app highlights select use case/patterns for PreEmptive Analytics users using the PreEmptive Analytics API embedded inside a C# mobile app within a Xamarin Visual Studio project.

The sample application can be used as:

- Presentation and instructional material,
- Code samples for developers

Content includes:

- The application itself as both a finished executable (Android initially with iOS coming soon) and as a Visual Studio development project.
- PreEmptive Analytics Workbench extensions as a development project that can be applied to user/dev-managed Workbench instance (either in a private Azure VM or on-premises)
- Instructions on how to access a PreEmptive Solutions Azure hosted instance of both the extended Workbench and an instance of PreEmptive Analytics for TFS (for demonstration purposes only).

## The application

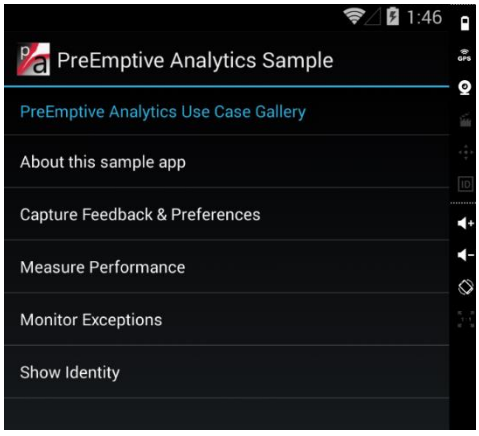
The PreEmptive Analytics Sample App lets users select preferences, time their actions, and throw different types of exceptions inside the app. All of this "user and application behavior" including the specific features exercised during each session are tracked using PreEmptive Analytics instrumentation. Additionally, a "user identity" consisting of a role and a user key is synthesized on a session-by-session basis – you can see your value at any time by visiting the Identity page inside the app.

*NO* actual *Personally Identifiable Information (PII) is ever collected.*

The resulting telemetry is transmitted to a PreEmptive Analytics™ endpoint where it is ingested and processed using the PreEmptive Analytics Workbench and PreEmptive Analytics for TFS.

For questions or to request access to hosted versions of the PreEmptive Anlaytics Workbench and PreEmptive Analytics for TFS, email sampleanalyticsapp@preemptive.com. For more information on PreEmptive Analytics (on *and beyond* mobile), visit www.preemptive.com/pa.

## Inside the Application: Usage, behavior, and quality measurement



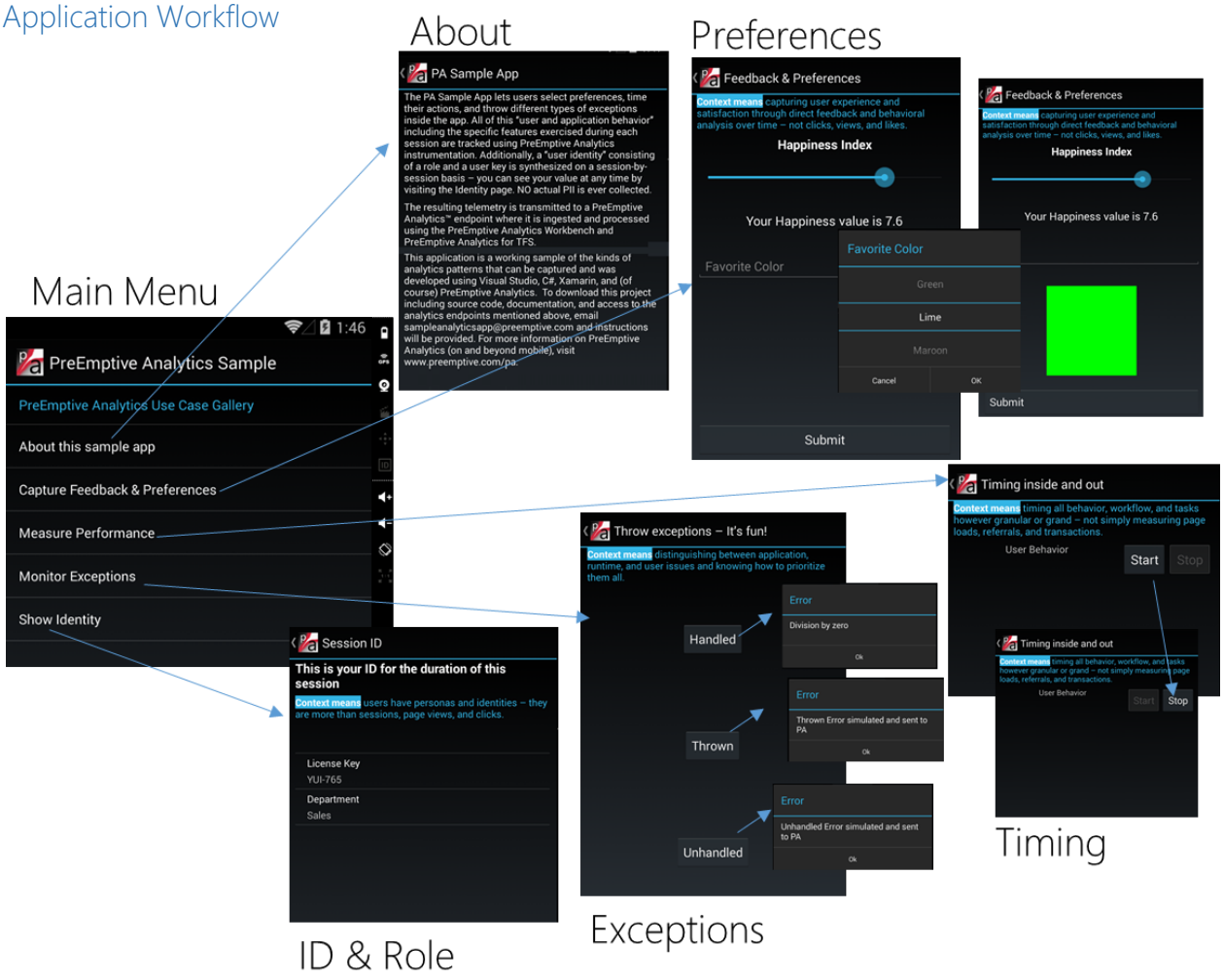

**Capture feedback, preferences, and custom data:** Users share their mood and their preference for a color – these values can represent any dynamic state of an app or any combination of user behaviors.

**Measure performance:** Users start and stop a timer whose duration can span pages and operations – these values can represent complex user workflows, system calls, or any other meaningful event span.

**Monitor exceptions:** Users can trigger thrown, caught, and unhandled exceptions – these events can represent application defects, system failures, malicious usage, user error or any other material production incident.

**Identity**: Each session will have a generated "license key" and role – no other context is captured at the client. All other context will be generated at the analytics endpoints through analysis and data enrichment.

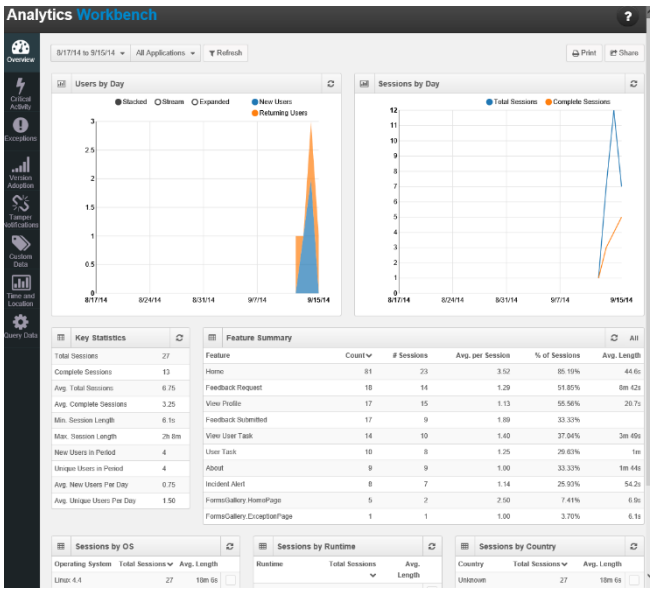
## Application Workflow




The application is simple to navigate making runtime data generation easy and transparent.

## Workbench Experience

The data transmitted from the sample app will be immediately available inside the associated Workbench in both "standard" and "full context" formats, where the "full context" format utilizes the extensions included with this sample environment.

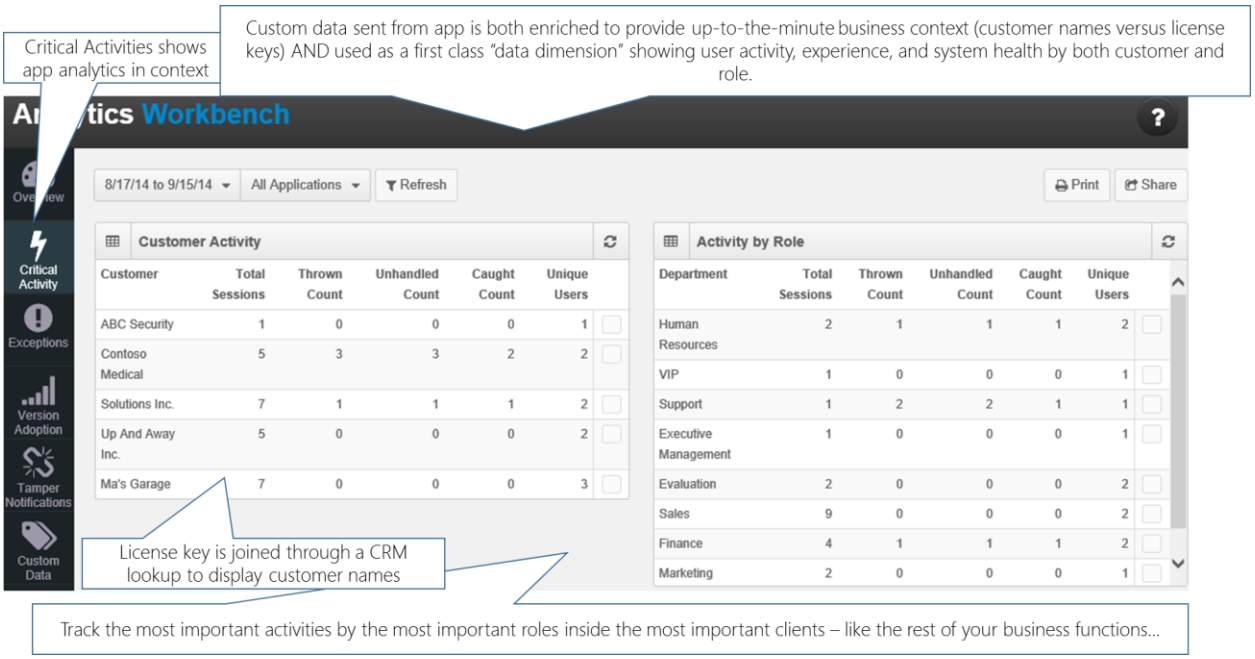



### Standard format

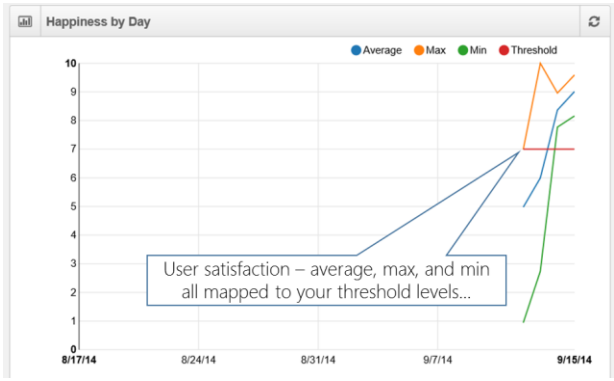
The standard format captures "one-size-fits-all" metrics including users, sessions, features, exceptions, basic timing, runtimes and OS data.

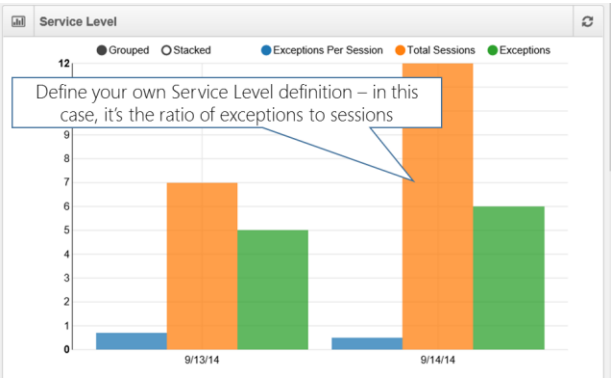
Filtering on equally generic dimensions including apps, versions, date ranges, runtimes, etc. is also supported.

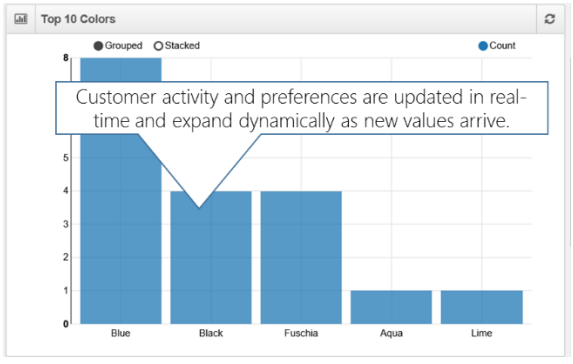
## Full context format



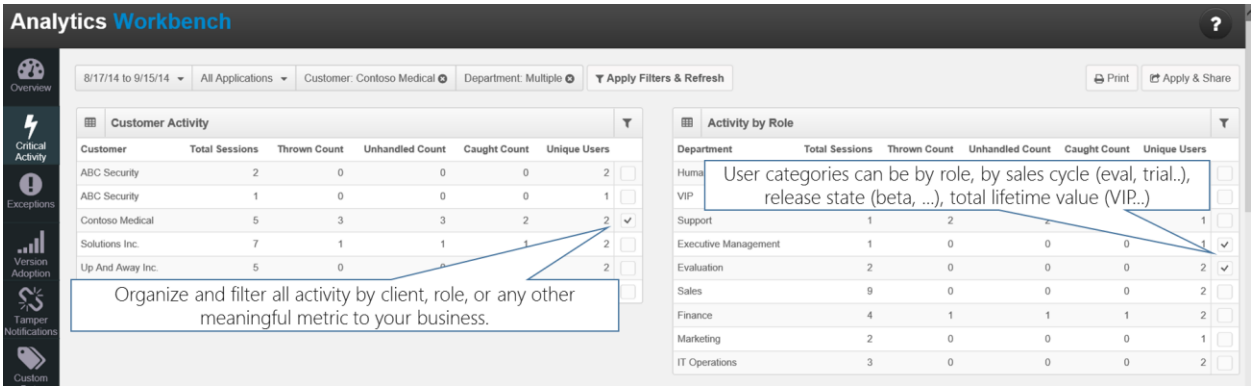

And this perspective is extended to reflect feature use, preferences, and other business and usage-specific activities.



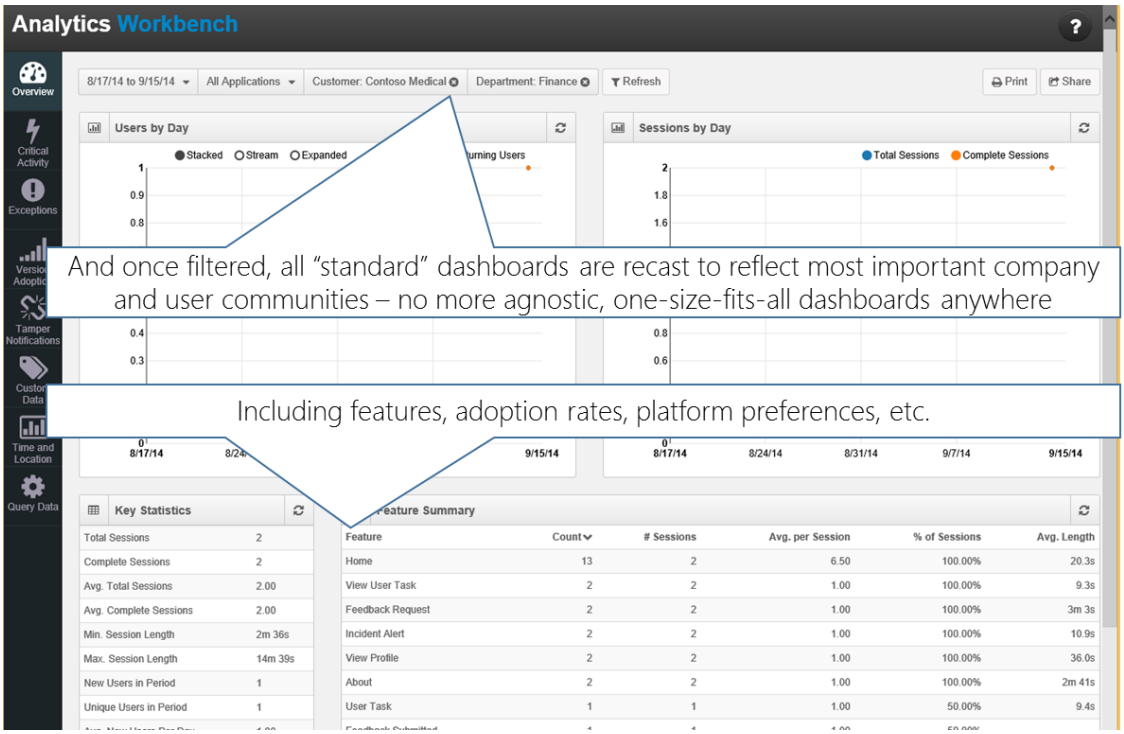



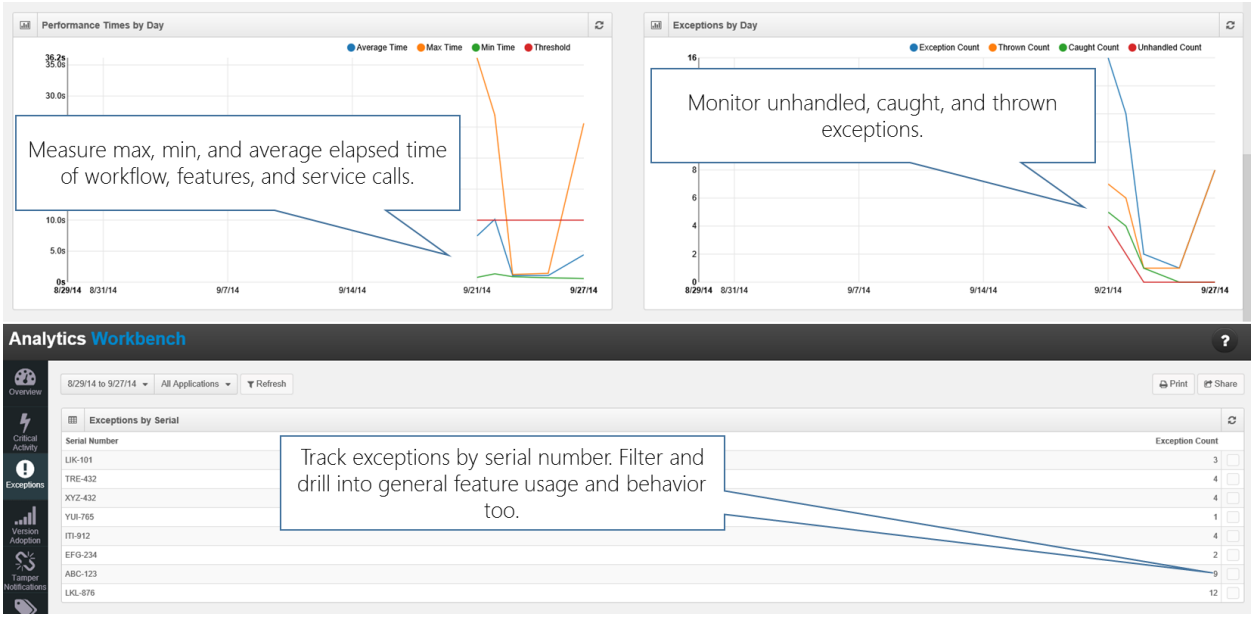
## Understanding the most important activities by the most important people inside your most important clients and communities




## And that business critical filter can be applied back to all of the "standard" dashboards and KPIs



Timing of complex tasks, system calls, and user behavior – as well as monitoring of all class of exceptions can also be filtered by user, organization, and any other dimension you can define.




…And for those exception patterns that are already understood to be material, exceptions and other supporting data can ALSO be sent into PreEmptive Analytics for TFS for automatic analysis. Once identified as important, TFS work items are created.