**General**

|  |  |
| --- | --- |
| **Business Transactions:**   * E2E path used to fulfill a service or a business function provided by the app * Critical business functions of an application * I consider my app to be Golden / Healthy if my users can do these 10 things. * Top 5/10/15/20 things that your users must be able to do efficiently in order for the app to be considered healthy * Non-negotiables for an application to be considered “Golden” * Key interactions that your users perform within the app that needs be monitored & evaluated continuously to ascertain if the app is healthy & performing as expected | **Why monitor Micro-services?**   * BT’s provide the only consistent measurement across code & architectural changes * SEP’s provide a service focused perf measurement * E2E is critical in micro-services since each service will be a business function   Correlation across languages will be key since each micro-services can be diff language |
| Why APM for **DevOps**?   * **Adaptive Monitoring:** As code & architecture changes, monitoring can automatically adapt * **Quantifying Change:** Quantify if the change was good or bad. * **Proactive Detection:** Detect degradations. Measure BT’s (business function). Ultimately DevOps or not, the app is servicing the users by providing specific “business functions” and BT’s are the only consistent measurement point for application performance * **Business Impact:** Understand & quantify impact of degradation | **What are EUM injection methods available: (https://docs.appdynamics.com/display/PRO44/Overview+of+Injection+Types)**   * Manual injection into application pages (choose JS hosting option, then add script to each page) * Auto-injection * Assisted Injection (via Injection Rules or Attribute Injection JS\_HEADER & JS\_FOOTER) * Web server (Apache or Nginix) mod\_substitue, mod\_filter or mod\_rewrite rules   F5 I-Rules (iRules can modify network traffic passing through BIG-IP) |
| **How can we incorporate APM into DevOps Cycles:**   * Deployment automation using Chef & Puppet * Integration with CI pipelines to push release notifications into AppD * SNOW Integration * Release Comparison & Business Performance visualization | **What is Business IQ:**   * Extension of Core APM * Provide business context * Alert on business health (not just technical health) * Collect performance & business meta-data for 100% of the transactions * Give actionable insights on the performance and business health of applications in real time |

**Competitive**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Dynatrace Differentiators**   |  |  | | --- | --- | | **Core APM** | * Built primarily for dev, Overhead Limiter * 4 diff products with 5 diff UI’s * Not scalable & needs more infra * BT’s require manual config | | **DT SaaS** | * OneAgent is a security risk (root access, ldpreload) * Modifies AppServer process to run as DT user. Spawn multiple (up to 5) processes * Pulls down updates & Auto-updates agent binaries (not desirable in enterprise prod) * Breaks the DevOps paradigm of immutable containers | | **Analytics**  **(Business Insights)** | * Pre-configured widgets with fixed metrics (42) * No capability to add custom metrics * No Query based UI | |
| **NewRelic Differentiators**   |  |  | | --- | --- | | **Core APM** | * Only SaaS * No baselines only Apdex * Only instrument standard framework code, Custom code needs to be identified explicitly * No Tag & Trace * Random sample every 100 ms -> means you will miss problems * BT’s require manual config * No DB or server monitoring | | **Analytics (Insights)** | * Does not include machine data * Extensive code changes are needed to include Custom metrics * Query Language based (Vs AppD Drag-drop). No explanation of data dictionary   **AD Advantages**   * Better context for complex use cases linkage between analytics results, Core APM and RUM snapshots * Analytics metrics for business alerting * Deployment flexibility (On-prem & Cloud) * Analytics includes Logs * Better RBAC * Flexibility to collect data with or without code changes   **NR advantages:**   * Very easy to get going * Developers sometimes don’t mind changing code * Interface is nice & fast. Query language is slightly ahead of ADQL | |
| **Why not DT Business Insights**   * Pre-configured widgets with fixed metrics (42) * No capability to add custom metrics * No Query based UI |
| **Why not NR Insights**   * Does not include machine data * Extensive code changes are needed to include Custom metrics * Query Language based (Vs AppD Drag-drop). No explanation of data dictionary |
| **Marketing Analytics**  Competitor Advantages (Adobe & Google)   * Social Analytics: What content drives Likes & Tweets * Traffic Source Analytics: Which campaigns are effective * Site Flow Analytics: Funnel with diff paths to see how customers are navigating the app * A/B Testing with both versions running simultaneously * Campaign Building Tools * Content Management & Media Optimization   Position AppD as complementary to Marketing Analytics. |