

DevAnalytics

User Guide — Last update: 19 January 2021

Basis Technologies

Table of Contents

- Introduction 1
- DevAnalytics..... 2
 - Key Features 3
 - Technical Specification 4
- Key Areas & Metrics 5
 - Velocity 6
 - Cycle Times 8
 - Rework 9
 - Work In Progress 10
 - Approval Times 11
- Visualization 12
- Reporting 13
 - Reporting classifications 14
 - Running DevAnalytics to view and extract data for reporting 15

Introduction

This document provides instructions for the user guide and information of DevAnalytics to provide detailed analysis of the performance of SAP development and delivery processes.

DevAnalytics

DevAnalytics: What Is It?

DevAnalytics provides data-driven insight that can transform delivery of SAP change. Its suite of detailed metrics can be used to identify waste and inefficiency, and to benchmark performance. Near-realtime insight based on actual SAP change and transport data is delivered automatically, allowing you to measure and assess performance and deliver continuous improvement. A comprehensive set of metrics identifies waste and inefficiency, helping businesses to save time, reduce costs and respond more quickly. DevAnalytics provides detailed information about your SAP development and delivery processes but also delivers the means to create at-a-glance dashboards.

DevAnalytics provides:

- Insight that is unavailable elsewhere
- A view of current and historical data
- A unique means to benchmark performance (both internally and against peers)

Using ActiveControl to manage SAP change enables comprehensive meta-data to be gathered around changes so that they can be categorised by:

- Transport path/SAP module (e.g. ECC, BW)
- System (e.g. Dev, QA, Pre-prod, Production)
- Supplier (analyse internal and external teams)
- Project (assess specific performance)
- Functional area (e.g. Finance, Sales)
- Type of change (e.g. enhancement, bug fix)
- Priority (e.g. standard, emergency)

Without ActiveControl, DevAnalytics will provide transport level metrics based on SAP transport logs but will not be able to:

- Report at change level (e.g. business requirement, change request)
- Provide rework and waste metrics
- Provide approval time metrics

Key Features

1. Detailed analysis of SAP development and change processes

Identify bottlenecks, waste and re-work;

No additional software required;

Deeper insight is available on systems running ActiveControl;

2. Around 35 pre-defined and configurable metrics across 5 key areas

No need to create definitions and metrics from scratch – start building reports immediately.

3. Near real-time insight

Data can be updated daily to allow up-to-the-minute decision making

4. No historical time limit

Analyse old project data for insight and comparison – if there's data, DevAnalytics will break it down for you.

5. Selective, granular reporting

Mine data by date range, aggregation period, SAP module, project, team & more

6. S/4 & HANA support

Connect to SolMan on any SAP system; no other tools required

7. Variety of output types to suit user preference

Choice of numeric and graphical output;

Output data in format suitable for 3rd-party visualisation tools.

Technical Specification

Below are some technical specification guidelines for DevAnalytics

Metric categories	5 – velocity, cycle times, rework and waste, work in progress, approval times
Pre-defined metrics	35 ¹ (see DevAnalytics Feature Sheet for full list)
Reporting classification	8 – date, transport path, system, supplier, project, functional area, type of change, priority
Data refresh frequency	Daily, or on demand
User interface	External visualization tool (via data export) or SAP GUI (graphical and tabulated data)
Data export format	*.csv, *.xls, *.xlsx; Data exports can be filtered according to required time period
Transport Espresso compatibility	Transport Espresso v5.30 or above is required for full functionality
SAP Server Requirements	DevAnalytics requires SAP Solution Manager NetWeaver 7.01 or newer
Supported SAP applications	ABAP, Java and HANA applications managed via transports
S/4HANA compatibility	Yes (see above)
Disk space requirements (typical)	5GB-20GB ² ; data archiving function supplied

Key Areas & Metrics

DevAnalytics delivers 35 pre-defined, configurable metrics across 5 key areas: velocity, cycle times, rework and waste, work in progress and approval times.

Example Metric Visualisation

	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11
Business Task / Requirement	Cycle 1							Cycle 2			
Transport 1	D	D	D	Q	Q	Q	Q	Q	Q	Q	P
Transport 2		D	D	Q	Q	Q	Q	Q	Q	Q	P
Transport 3								D	D	Q	P
Elapsed time in Dev	D	D	D	D	D	D	D	D	D		9
Elapsed time in QA				Q	Q	Q	Q	Q	Q	Q	7
Total time in Dev	D	D	D					D	D		5
Total time in QA				Q	Q	Q	Q			Q	5
Total time to Completion	P	P	P	P	P	P	P	P	P	P	10
Cycles from Dev to QA	1	1	1	1	1	1	1	2	2	2	2
Daily WIP in DEV	WIP	WIP	WIP					WIP	WIP		
Daily WIP in QA				WIP	WIP	WIP	WIP			WIP	
Daily WIP in DEV & QA	WIP	WIP	WIP	WIP	WIP	WIP	WIP	WIP	WIP	WIP	

The Metric example above shows you the logic of the DevAnalytics metrics and how it provides detailed level analysis at transport and change level. It has transport 1 as an example going through the cycle from development(D) through to Production(P) and showing how long it took to complete the change, and whilst it is going through this cycle there is also 5 other key areas that contain 9 metrics (bottom half of the screenshot) being analysed and extracted.

Velocity

Velocity metrics identify how much new and updated functionality is being delivered into your SAP systems, by reviewing the number of transports and changes being deployed.

Velocity measures how much work has been done in a given time period. In the context of DevAnalytics, that means it's a measure of the amount of change that is actually making it to QA and production – and hence to testers and end users – in the time period that you choose to monitor.

Velocity is a clear measure of how much work has actually been finished, regardless of how efficiently that work was carried out. It's important during development planning as it provides a guide to the work that teams might realistically complete in each phase based on past performance.

Velocity is higher as the number of changes deployed increases and is lower when the deployment rate reduces.

DevAnalytics provides 12 velocity metrics:

- Deployments – Total (Transports) = The total number of transport deployments into each target system on any selected date
- Deployments – Success (Transports) = The total number of successful transport deployments into each target system on any selected date.
- Deployments – Error (Transports) = The total number of transport deployments in error into each target system on any selected date.
- Deployments – Percentage in error (Transports) = The percentage of transport deployments in error on any selected date.
- Deployments – Total tasks (Tasks) = The total number of task deployments into each target system on any selected date.
A task is considered to be deployed if any of its component transports have been deployed. For example, If a task has transports that have been imported into 3 different target systems on a specific date this will count as 3 task deployments.
- Deployments – Success tasks (Tasks) = The total number of successful task deployments into each target system on any selected date.
- Deployments – Error (Tasks) = The total number of task deployments in error into each target system on any selected date.
- Deployments – Percentage in error (Tasks) = The percentage of task deployments in error on any selected date.
- Deployments – Emergency (Tasks) = The total number of emergency task deployments into each target system on any selected date. A task is considered to be an emergency if it's priority is set to Emergency / Urgent.
- Deployments – Emergency percentage (Tasks) = The percentage of emergency task deployments on any selected date.
- Users involved in unreleased transports (Average) = The average number of users with transports that are unreleased on any selected date including sub-task users.
- Users releasing transports (Average) = The average number of users releasing transports on any

selected date.

Cycle Times

Cycle times are a measure of the total time it takes to deliver functionality to the business. They tell you the duration of development and testing activity from the moment work is started until its completion. That is, how long it actually takes to get each 'job' done (i.e. change delivered).

Where a change consists of multiple inter-connected pieces of development, some of which require more rework than others, cycle times tell you how long all this activity is taking.

Cycle times are a critical measure of how efficiently your development and delivery processes are functioning. It's the metric to look at when you want to know 'if we start this now, when can the business expect to see it?'. A reduction in cycle times can equate directly to faster time to market.

DevAnalytics allows you to measure the number of days being spent in Development and QA along with the total time from the start of Development to Completion.

DevAnalytics provides 8 cycle times metrics:

- Time in DEV – Average (Transports) = The average amount of time in days that transports have spent in development.
- Time in QA – Average (Transports) = The average amount of time in days that transports have spent in QA.
- Time to Completion – Average (Transports) = The average amount of time in days that transports have taken to be completed. (E.g. deployed into Production or Merged).
- Elapsed time in DEV – Average (Tasks) = The average amount of time in days that tasks have spent in development – from the first transport creation until the deployment of all transports for the task into QA.
- Total time in DEV – Average (Tasks) = The average total time in days that tasks have spent in development only (excludes overlapping time in QA).
- Elapsed time in QA – Average (Tasks) = The average amount of time in days that tasks have spent in QA – from the first import of a transport for a task into QA to the time all transports for the task are completed. (E.g. deployed into Production or Merged).
- Total time in QA – Average (Tasks) = The average total time in days that tasks have spent in QA only (excludes overlapping time in development).
- Total time to Completion – Average (Tasks) = The average amount of time in days that tasks have taken to be completed – from the first transport creation to the time all transports for the task are completed. (E.g. deployed into Production or Merged).

Rework

Rework metrics measure the efficiency of your development and delivery processes, and can highlight bottlenecks and cost caused by inferior quality.

It's not often that an SAP change makes its way from development to production at the first attempt. That may be understandable given the complexity of many systems and the high level of interdependency that may exist between different transports, but rework is a significant drain on resources.

Insight into levels of rework provides perhaps the clearest guide as to how you can optimise delivery of SAP change. Reducing rework means you can deliver faster, increase the volume of change that makes its way to production, and lower development and QA costs. You may even manage all three.

Rework is essentially a measure of the number of additional cycles it takes to deliver changes to completion. For example, a perfect change deployment would be done in one cycle. If a change failed testing twice before completion then that would add two rework cycles giving a total of three cycles to completion. In essence when a change has gone backwards and forwards in the change life cycle from development through to another system further in the path, it will evaluate the cycles and see how many times its gone through the delivery to deployment phase.

DevAnalytics provides 5 rework metrics:

- Cycles from DEV to QA – Average (Tasks) = The average number of cycles from Development to QA for tasks. A cycle represents the deployment of all the transports for a task into QA for testing. Each time there is a test failure another development cycle will be required to fix it. More than 1 cycle indicates a rework.
- Percentage of business tasks with reworks from DEV to QA = The percentage of tasks with more than 1 cycle from Development to QA indicating that there were reworks.
- Cycles from DEV to PROD – Average (Tasks) = The average number of cycles from Development to Production for tasks. A cycle represents the deployment of all the transports for a task into Production. Each time there is a Production failure another development cycle will be required to fix it. More than 1 cycle indicates a rework.
- Percentage of business tasks with reworks from DEV to PROD = The percentage of tasks with more than 1 cycle from Development to Production indicating that there were reworks.
- First pass percentage in QA (Tasks) = The percentage of tasks with only 1 cycle from Development to QA indicating that there no were reworks.

Work In Progress

Work in progress metrics provide visibility of the total amount of on-going development and testing activity for work that has already begun.

Some changes never make it to production and may stack up due to dependencies on other tasks or due to a changes in business requirement or priorities. The cumulative total can be significant – each work in progress change represents an investment that has yet to deliver any return.

Work in progress can help you to judge the efficiency of your processes and provides a means to track the on-going opportunity cost of change that has yet to be delivered.

DevAnalytics provides the ability to count the number of changes and transports that are currently in progress either in Development or QA.

DevAnalytics provides 6 work in progress metrics:

- WIP in DEV – Daily Average (Transports) = The average number of transports in Development on any selected date where no deployment into QA has taken place.
- WIP in QA – Daily Average (Transports) = The average number of transports in QA on any selected date where they have been deployed into QA but are not yet completed. (E.g. deployed into Production or Merged).
- WIP in DEV & QA – Daily Average (Transports) = The average number of transports in Development and QA on any selected date where they are not yet completed. (E.g. in Production or Merged).
- WIP in DEV – Daily Average (Tasks) = The average number of tasks in Development on any selected date where no deployments into QA for any transports in the tasks have taken place.
- WIP in QA – Daily Average (Tasks) = The average number of tasks in QA on any selected date where transports for the tasks have been deployed into QA but are not yet completed. (E.g. deployed into Production or Merged).
- WIP in DEV & QA – Daily Average (Tasks) = The average number of tasks in Development and QA on any selected date where they are not yet completed. (E.g. in Production or Merged).

Approval Times

The effectiveness of approval processes, and the bottlenecks they can cause are difficult to measure – whether due to processes or people – and can be hard to prove.

The approval time metrics supplied by DevAnalytics provide a clear view of where approvals are being delayed, and so can offer an easy starting point for any attempts at improving speed and efficiency.

DevAnalytics provides the ability to understand the number of days that changes and transports have been awaiting approval.

DevAnalytics provides 4 approval time metrics:

- Approval time into QA – Average (Transports) = The average amount of time in days it takes for transports to be approved for deployment into QA.
- Approval time into PROD – Average (Transports) = The average amount of time in days it takes for transports to be approved for deployment into Production.
- Approval time into QA – Average (Tasks) = The average total amount of time in days it takes for the transports for tasks to be approved for deployment into QA.
- Approval time into PROD – Average (Tasks) = The average total amount of time in days it takes for the transports for tasks to be approved for deployment into Production.

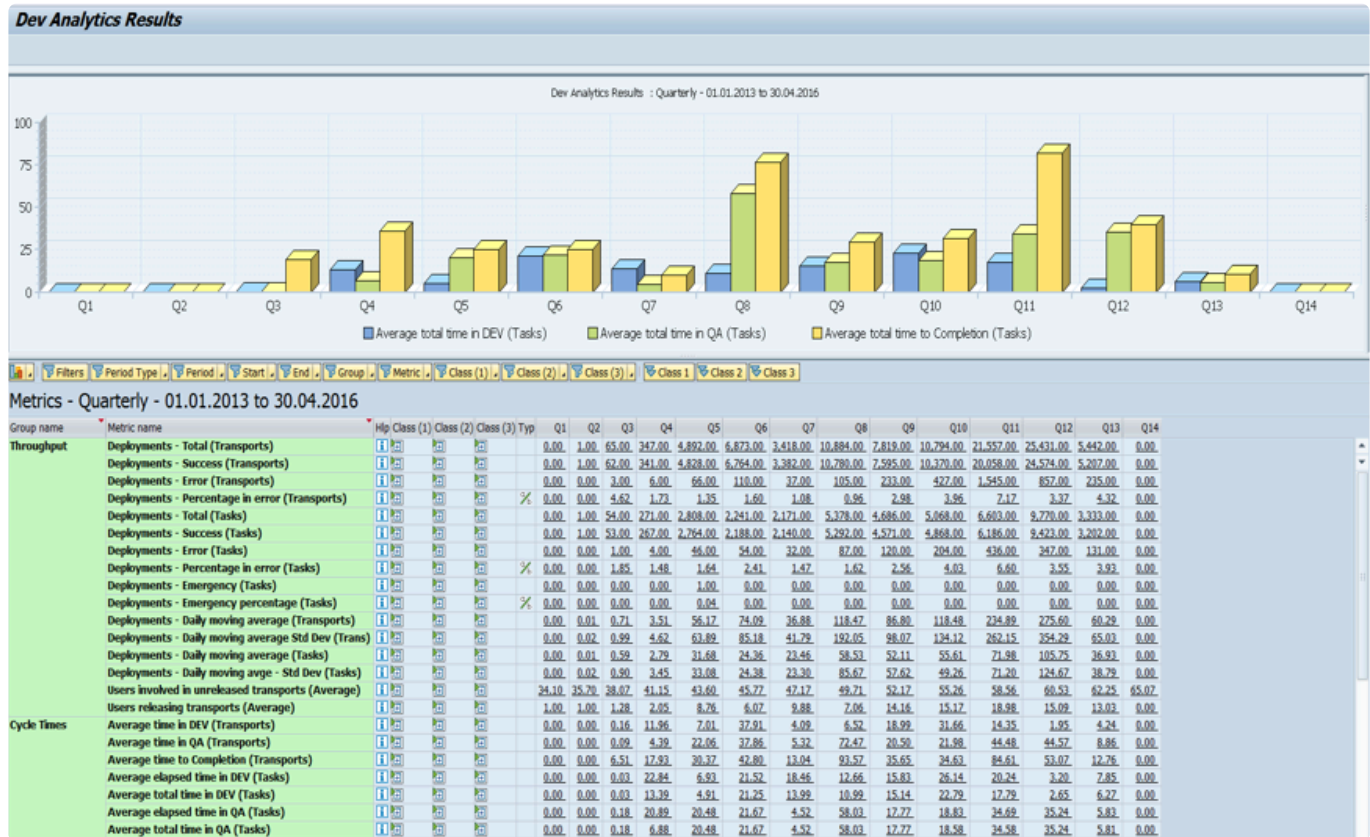
Visualization

SAP GUI

No additional software is required in order to view output from DevAnalytics.

Results can be presented in the standard SAP GUI in either tabular or graphical format.

Example



Example of drilled down details within each metrics:

Dev Analytics Results

Request ID	Current Date	Time	Name	Ret. code	Path	Target	Group	Project	Type	Priority	Supplier	N
D01K916508	21.08.2017	06:42:14		8	ECC BAU / Production Support	I01 ECC Production - BAU	HR	BAU - Weekly Release (Week #1)	Incident Fix	Normal	Internal	
D01K929237	21.08.2017	06:42:15		0	ECC BAU / Production Support	I01 ECC Production - BAU	FICO	BAU - Weekly Release (Week #2)	Business Change	Normal	Internal	
D01K933211	14.08.2017	03:22:07	Ross McLanachan	0	PI BAU / Production Support	D01 PI Development - BAU	SD	BAU - Weekly Release (Week #1)	Business Change	Normal	Internal	
D01K933211	14.08.2017	08:23:04	Ross McLanachan	0	PI BAU / Production Support	T01 PI Test - BAU	SD	BAU - Weekly Release (Week #1)	Business Change	Normal	Internal	
D01K933211	14.08.2017	08:23:04	Ross McLanachan	0	PI BAU / Production Support	R01 PI Production - BAU	SD	BAU - Weekly Release (Week #1)	Business Change	Normal	Internal	
D01K933211	21.08.2017	06:50:34		4	PI BAU / Production Support	R01 PI Production - BAU	SD	BAU - Weekly Release (Week #1)	Business Change	Normal	Internal	
D01K933211	21.08.2017	06:55:41		4	PI BAU / Production Support	R01 PI Production - BAU	SD	BAU - Weekly Release (Week #1)	Business Change	Normal	Internal	
D01K933219	14.08.2017	03:22:07	Ross McLanachan	0	PI BAU / Production Support	D01 PI Development - BAU	HR	BAU - Weekly Release (Week #1)	Operational Change	Normal	Internal	
D01K933219	14.08.2017	08:23:04	Ross McLanachan	0	PI BAU / Production Support	T01 PI Test - BAU	HR	BAU - Weekly Release (Week #1)	Operational Change	Normal	Internal	
D01K933219	16.08.2017	06:31:26		0	PI BAU / Production Support	R01 PI Production - BAU	HR	BAU - Weekly Release (Week #1)	Operational Change	Normal	Internal	
D01K933219	16.08.2017	08:51:13		4	PI BAU / Production Support	R01 PI Production - BAU	HR	BAU - Weekly Release (Week #1)	Operational Change	Normal	Internal	
D01K933219	21.08.2017	06:55:43		4	PI BAU / Production Support	R01 PI Production - BAU	HR	BAU - Weekly Release (Week #1)	Operational Change	Normal	Internal	
D01K933221	14.08.2017	03:22:07	Ross McLanachan	0	PI BAU / Production Support	D01 PI Development - BAU	HR	BAU - Weekly Release (Week #1)	Operational Change	Normal	Internal	
D01K933221	14.08.2017	08:23:04	Ross McLanachan	0	PI BAU / Production Support	T01 PI Test - BAU	HR	BAU - Weekly Release (Week #1)	Operational Change	Normal	Internal	
D01K933221	21.08.2017	16:03:16		0	PI BAU / Production Support	R01 PI Production - BAU	HR	BAU - Weekly Release (Week #1)	Operational Change	Normal	Internal	
D01K933223	14.08.2017	03:22:07	Ross McLanachan	0	PI BAU / Production Support	D01 PI Development - BAU	HR	BAU - Weekly Release (Week #2)	Normal Change	Normal	Internal	
D01K933223	14.08.2017	08:23:04	Ross McLanachan	0	PI BAU / Production Support	T01 PI Test - BAU	HR	BAU - Weekly Release (Week #2)	Normal Change	Normal	Internal	
D01K933307	14.08.2017	03:22:07	Ross McLanachan	0	PI BAU / Production Support	D01 PI Development - BAU	HR	BAU - Weekly Release (Week #1)	Operational Change	Normal	Internal	
D01K933307	14.08.2017	04:17:40	Ross McLanachan	0	PI BAU / Production Support	D01 PI Development - BAU	HR	BAU - Weekly Release (Week #1)	Operational Change	Normal	Internal	
D01K933307	14.08.2017	04:47:50	Ross McLanachan	0	PI BAU / Production Support	D01 PI Development - BAU	HR	BAU - Weekly Release (Week #1)	Operational Change	Normal	Internal	
D01K933307	14.08.2017	08:23:04	Ross McLanachan	0	PI BAU / Production Support	T01 PI Test - BAU	HR	BAU - Weekly Release (Week #1)	Operational Change	Normal	Internal	
D01K933307	21.08.2017	13:01:00		0	PI BAU / Production Support	R01 PI Production - BAU	HR	BAU - Weekly Release (Week #1)	Operational Change	Normal	Internal	
D01K900734	14.08.2017	03:22:07	Ross McLanachan	0	PI BAU / Production Support	D01 PI Development - BAU	Basis	KSE test	Normal Change	Normal		
T01K900734	14.08.2017	08:23:04	Ross McLanachan	0	PI BAU / Production Support	T01 PI Test - BAU	Basis	KSE test	Normal Change	Normal		
T01K900780	14.08.2017	03:22:07	Ross McLanachan	0	PI BAU / Production Support	D01 PI Development - BAU	Basis	KSE test	Normal Change	Normal		
T01K900780	14.08.2017	08:23:04	Ross McLanachan	0	PI BAU / Production Support	T01 PI Test - BAU	Basis	KSE test	Normal Change	Normal		
T01K900782	14.08.2017	03:22:07	Ross McLanachan	0	PI BAU / Production Support	D01 PI Development - BAU	Basis	KSE test	Normal Change	Normal		
T01K900782	14.08.2017	08:23:04	Ross McLanachan	0	PI BAU / Production Support	T01 PI Test - BAU	Basis	KSE test	Normal Change	Normal		

Reporting

Delivering a lot of change in your SAP systems generates a very large amount of associated data.

DevAnalytics allows you to create detailed reports and metrics so you focus on the areas of your processes that require the most attention.

Reporting classifications

Thanks to the 35 pre-defined metrics provided by DevAnalytics, you can begin building the reports you need almost immediately.

Data can be presented across any date range, and the results refined using eight reporting classifications (see right). This ability to benchmark selected areas – supplier performance, for example – means that future performance can be tracked and provides the means for you to focus on the particular SAP systems where there is most room for improvement.

Classifications can even be combined to provide precise insights, so that you can hone in on specifics. Want to find out which SAP system in which module had the longest approval times for a given project over the last six months? Well now you can.

DevAnalytics supports the following classifications for data reporting:

- Date (any period plus aggregation by year, quarter, month, week or day)
- Transport path/SAP module (e.g. ECC, BW)
- System (e.g. Dev, QA, Pre-prod, Production)
- Supplier (analyse internal and external teams)
- Project (assess specific performance)
- Functional area (e.g. Finance, Sales)
- Type of change (e.g. enhancement, bug fix)
- Priority (e.g. standard, emergency)

Running DevAnalytics to view and extract data for reporting

Once the data for DevAnalytics has been generated (see Setup and config section) it can be analysed in SAP or exported for analysis in an external visualization tool.

Note that program **/BTR/DEVA_MAIN_N** must have been run with the required variant in order to generate the data. Any previously generated data can be analysed or exported.

Execution notes:

- **Output mode:** Choose how you want to output the data
 - **Hierarchical (in SAP GUI)** – This displays the metric data output in the SAP GUI via a grid and allows the user to drill down, filter, show graphs, etc.
 - **Report (ALV)** – Output the data as a flat ALV report
 - **Download to server** – Download the data to the specified server folder and filename in tab delimited format. The filename will automatically be appended with a date and timestamp
 - **Download to PC** – Download the data to the specified PC folder and filename in tab delimited format. The filename will automatically be appended with a date and timestamp
- **Analysis period:** Chose the date period that you wish to view the data for
- **Data source:** Always choose “Use instance results”
- **Metric selection:** Choose which metrics you wish to view and the period you wish to have the data aggregated by (E.g. Monthly)
- **Data restrictions:** Enter any specific selection criteria to narrow down the data selection for a more targeted view (E.g. only show data for specific systems and projects)
- **Transformation run options:** This allows specific data extract runs to be selected. In most cases the **Latest run** would be used but previous extract runs (if available) can also be selected if required

Notes

Hierarchical (in SAP GUI)

When the data has been output in SAP there are options to be able to manipulate the data on screen:

- Metrics can be added to / removed from the graphic section
- Different filters, periods, date ranges can be specified
- Data can be expanded to view a breakdown for each classification
- The data visible on screen can be exported to spreadsheet / Excel

Report and Download options

The data extracted in these modes split up by metric and the associated classifications so you can obtain lower level figures across all available classifications. These modes are best used when extracting the DevAnalytics data to be viewed in external data visualization tools.

* The Setup and configuration document has a step by step process on setup and extraction of DevAnalytics content ([Setup and Configuration](#))