

Revolution or Regression?

A Comparatively Empirical Study of Two Agile
development process



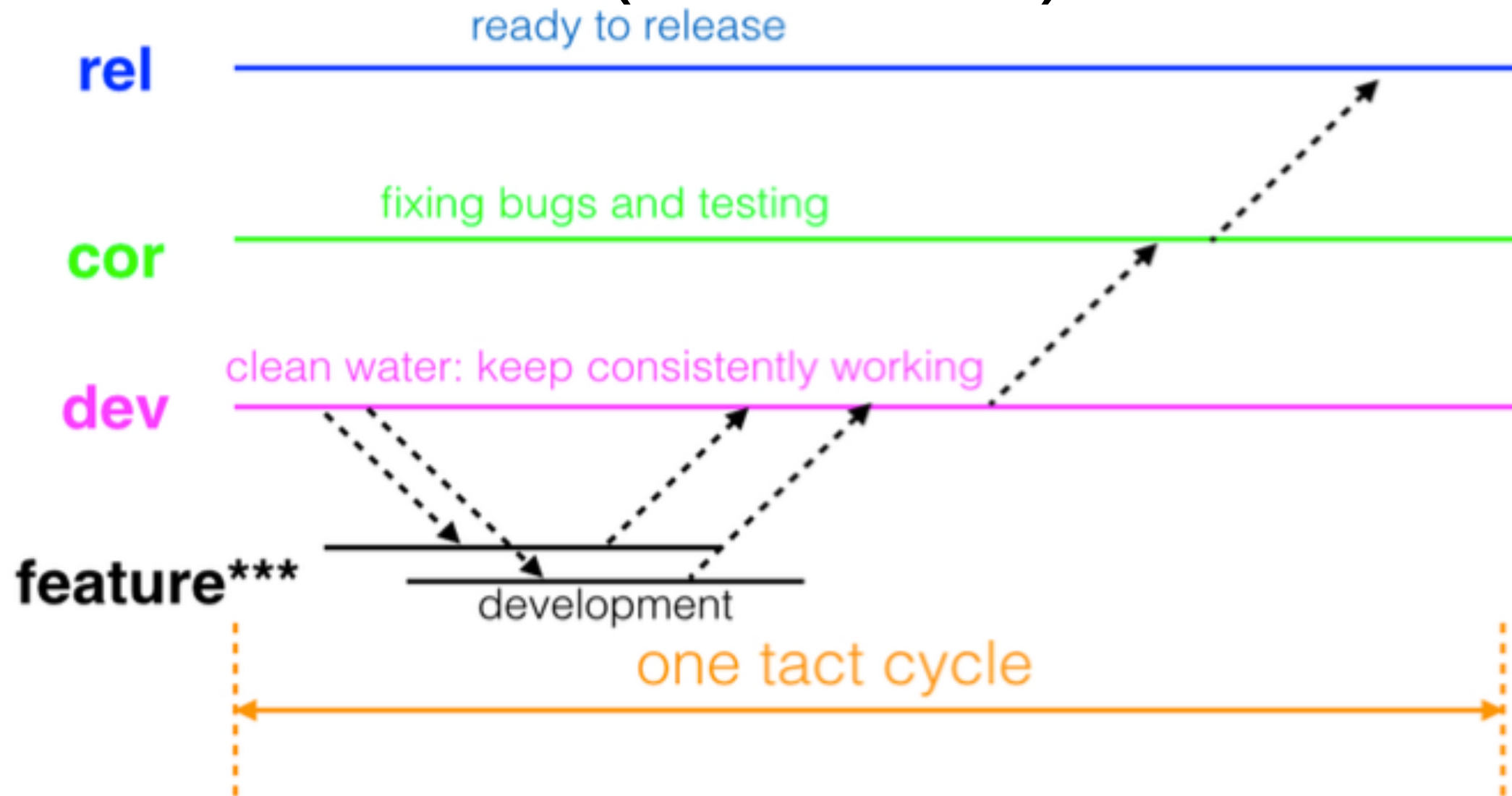
POLYTECHNIQUE
MONTREAL

LE GÉNIE
EN PREMIÈRE CLASSE



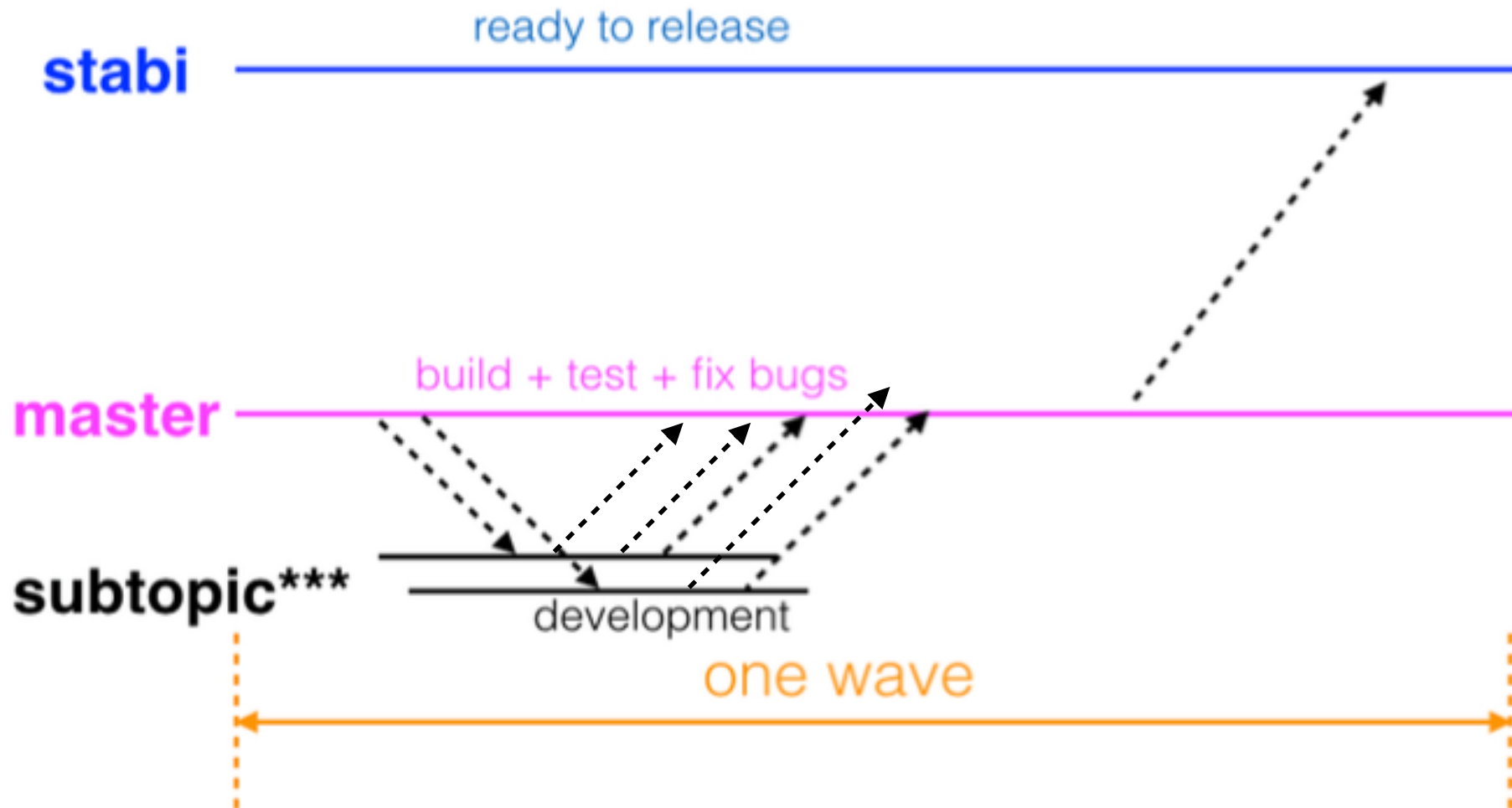
Yujuan Jiang, Bram Adams

Branch-based approach (Lumira)



Clean water development: each feature branch got merged back to dev branch only after the whole feature is completed. Only bug-fixes will be cherry-picked to cor/rel branch.

Toggle-based approach (Orca)



Each feature branch check in master branch whenever they want. Merge small, merge faster. Yet as long as the toggle of this feature is not turned “on”, the code will not be integrated into release branch.

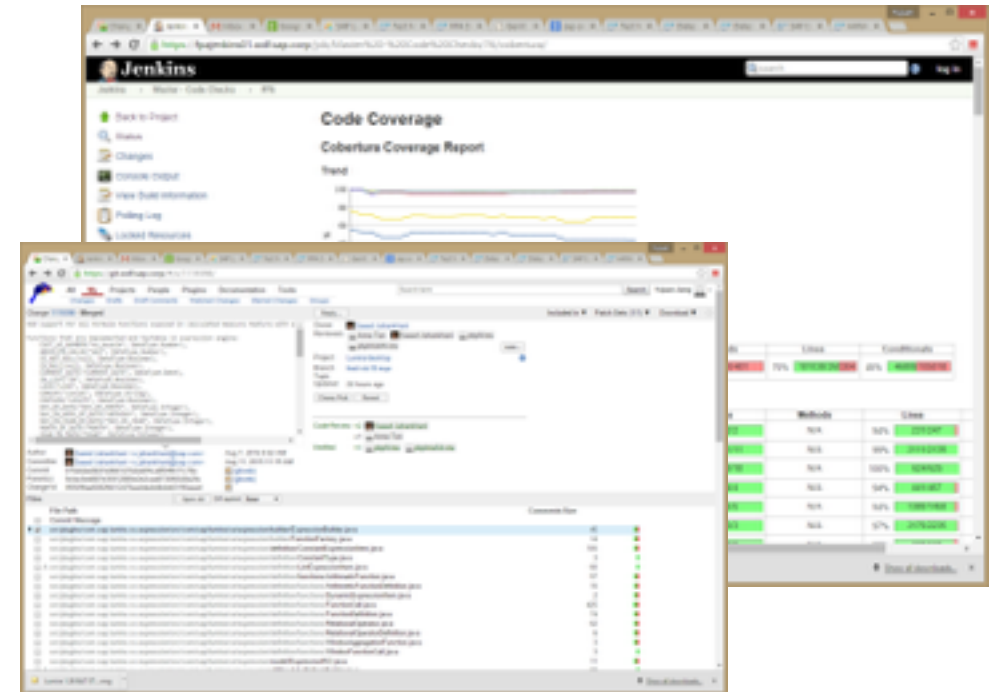
Key Performance Indicator (KPI)

- One Code Line: **full/avg TAT**, success rate, avg build time per merge commit
- High Velocity in Development: **# merged merge commit into master**/subtopic branch, avg # LOC per commit, success rate, avg time for regression voter
- Frequent Delivery: **# of cherry-picks** etc...

** Each group has a main KPI (highlighted) and a bunch of supporting KPIs.*

Fetching Data

- From Jenkins (build information)
- From Gerrit (review information)
- From JIRA (bug tracking information)



- * JSON + Python + Perl + .xsjs (javascript for SAP)
- * e.g., for test coverage ratio, we use Python script to fetch the data from Jenkins through JSON API, process locally to get the required data then send it to js script at HANA server, which insert them into HANA database.

Implementation & Modelling

- Modelling (visualizing the data):
 - For simple model:
 - * HANA studio graphical views
 - For complex model:
 - * procedure, i.e., the SQL script
 - * e.g., TAT (Turn Around Time), is a time span within which each feature branch checks in at least once.

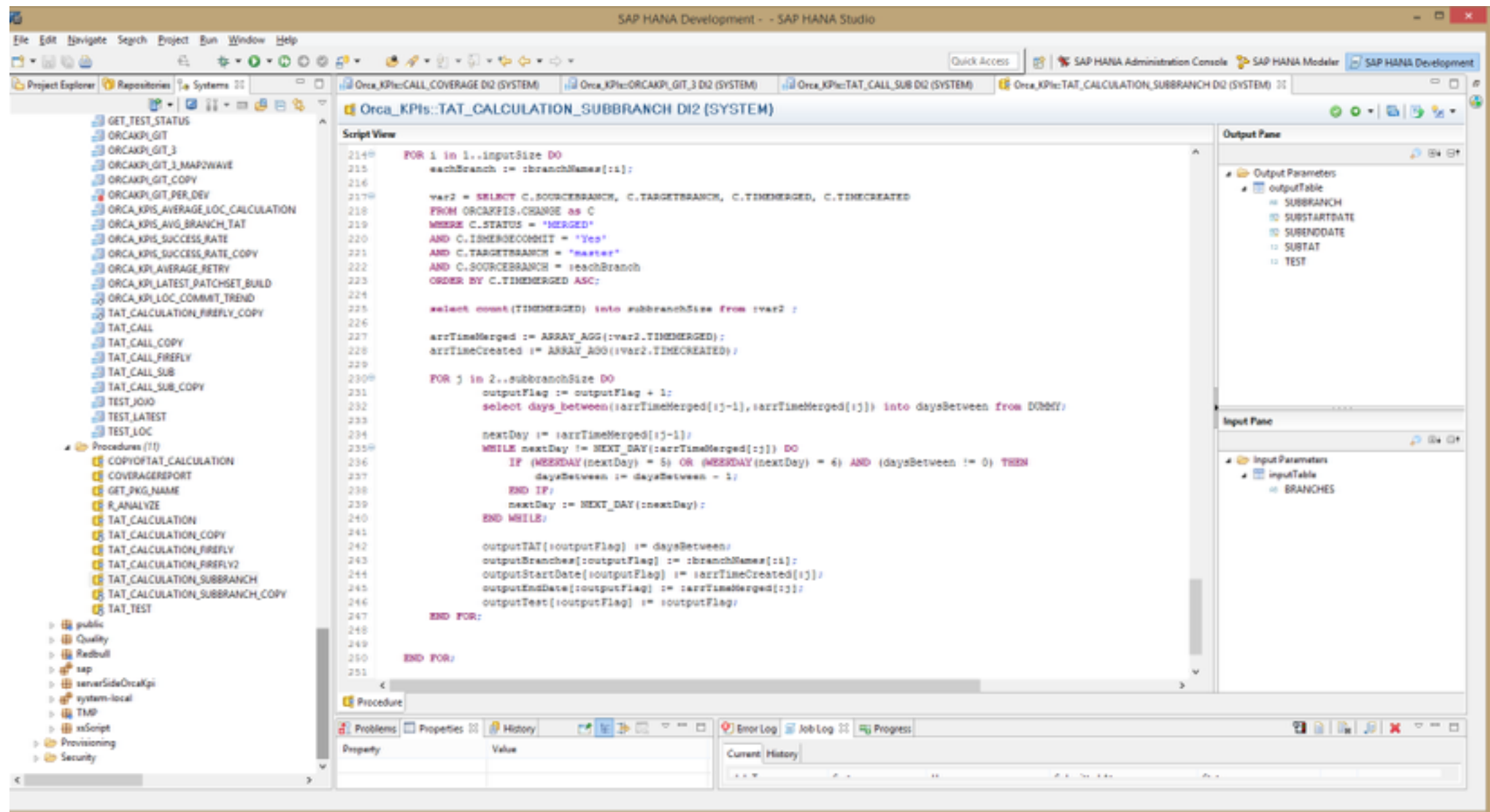
Example: Graphical View

The screenshot displays the SAP HANA Studio interface for a project named 'Orca_KPIs:ORCAKPI_GIT_3 D12 (SYSTEM)'. The main workspace shows a graphical data model (Scenario) with a flow diagram. The diagram starts with a 'Semantics' node, followed by an 'Aggregation' node, then a series of 'Join' nodes (Join_4, Join_3, Join_2, Join_1) and 'Projection' nodes (Projection_2, Projection_1). The 'Details' pane on the right lists columns with their properties, including Type, Key, Name, Label, Aggregation, Variable, Label Column, Hidden, and Value help View/Table. The 'Properties' pane at the bottom shows the general properties of the selected object, such as Name, Label (English), Package, and Activated By.

Type	Key	Name	Label	Aggregation	Variable	Label Column	Hidden	Value help View/Table
		dateCreated	dateCreated					
		dateMerged	dateMerged					
		numberCo...	numberCom...	Formula				
		numPatches...	numPatchset	Formula				
		numBuilds...	numBuilds	Formula				
		TotalLOCN...	TotalLOCNoD...	Formula				
		testDuration	testDuration	Formula				
		CHANGEID	CHANGEID					
		TARGETBR...	TARGETBRAN...					
		SOURCEBR...	SOURCEBRAN...					
		STATUS	STATUS					
		USERNAME	USERNAME					
		TIMEMERG...	TIMEMERGED					
		TIMECREA...	TIMECREATED					
		TOTALLIN...	TOTALLINES...					
		ISMERGE...	ISMERGE COM...					
		ISTRIVAL	ISTRIVAL					
		BUILDTIME...	BUILDTIMEID					
		BUILDSTAR...	BUILDSTARTTL...					
		BUILDIND...	BUILDENDTIME					
		PATCHSET...	PATCHSETRE...					
		BUILDTIME...	BUILDTIMEID...					
		NAME	NAME					
		EMAIL	EMAIL					
		numBuilds...	numBuilds	Formula				

A GUI tool easily play around the data with basic functionalities such as union/join/conversion/filtering...

Example: Procedure



SQL script, more flexible to deal with complex logic.

Implementation & Modelling

- Deploy to Orca

