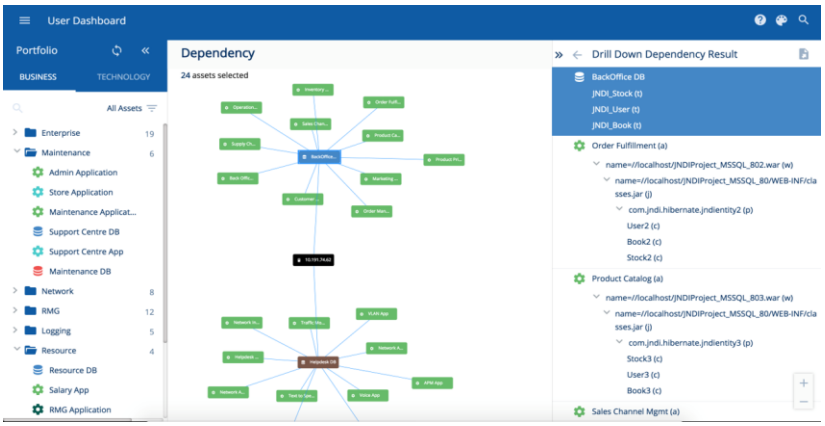
Crosscode Panoptics

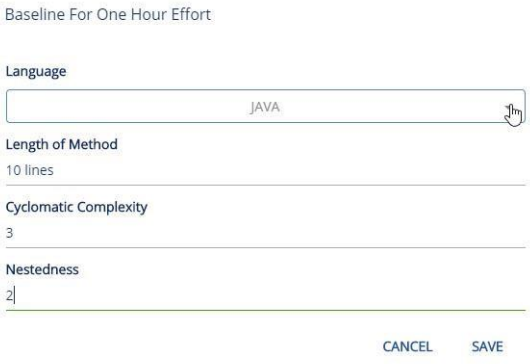
Automated Change Estimation

- Calculate Change Effort Quickly and Accurately
- Get More Proposals Approved
- Avoid Time and Cost Overruns
- Improve Margins / Budget Efficiency

Crosscode Panoptics introduces a compelling automated solution to the challenge of estimating software change quickly and accurately.

First, Panoptics discovers the applications and databases and maps the dependencies between them at every level of detail, down to method and column. It presents a high-level view showing which systems depend on each other.

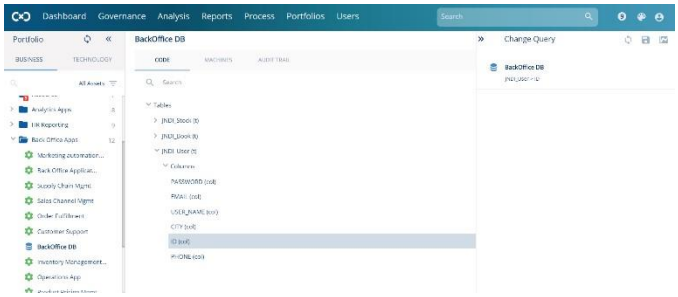




To calculate accurately the effort required for a given change, Panoptics needs to baseline the user's productivity. The user tells Panoptics how many lines of code a member of his team can change in an hour. This metric can be defined for each application language supported by Panoptics. In this screenshot, we see that in one hour, the average team member can change 10 lines of JAVA code, each having on average a nestedness of 2 and a complexity of 3.

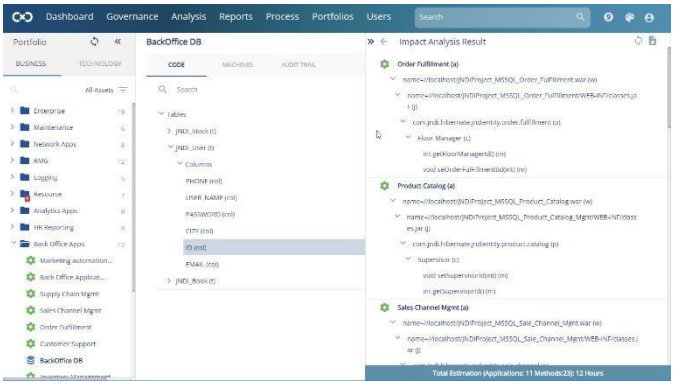
Now it's time to define the scope of change. Panoptics automatically groups systems into technology portfolios according to application language or database type. In addition, the user can easily assign an application or database to one or more business portfolios. Panoptics makes it easy to see at a glance which systems will be impacted by a change. In the screenshot below, the solutions architect has created a portfolio for all the systems that depend on the system that must be changed – in this case, the BackOffice database.

Panoptics automatically decomposes each application or database, showing the structure of all the elements within it. By clicking on the elements that are to be changed, the solutions architect creates a Change Query. A Change Query can include any number of elements from any number of applications and databases. Panoptics' Change Estimation feature calculates the effort required for any change, no matter how complex.



In this case, the business requirement is to extend the User ID column, so the Change Query consists of the element JNDI_User > ID:

With a single click, Panoptics instantly assesses the impact and calculates the effort required. It displays all the applications and methods that will be impacted by a change to the ID column. A change to the User ID column will impact 23 methods in 11 applications. Based on the number of lines per method and their nestedness and complexity, this change will require twelve hours.



Panoptics lets the user export this information so that it is easy to reference both the impact assessment and the change estimation. This information can be used to create an SOW. It also guides Testing and QA once the change is complete.

	A	B	C	D	E	F
2	Total Estimated Hours	12				
3	BackOffice DB	JNDI_Book > id				
4	Order Fulfillment (a)	JNDIProject_MSSQL_Order_FulFillment.war (w)	JNDIProject_MSSQL_Order_FulFillment/WEB-INF/classes.jar (j)	jndi.hibernate.jndientity.order.fulfillment (p)	OrderFulFillmentBook (c)	void setFloorManagerId(int) (m)
5						int getOrderFulFillmentId() (m)
6	Product Catalog (a)	JNDIProject_MSSQL_Product_Catalog.war (w)	JNDIProject_MSSQL_Product_Catalog_Mgnt/WEB-INF/classes.jar (j)	jndi.hibernate.jndientity.product.catalog (p)	ProductCatalogBook (c)	void setProductCatalogId(int) (m)
7						int getSupervisorId() (m)
8	Sales Channel Mgmt (a)	JNDIProject_MSSQL_Sale_Channel_Mgnt.war (w)	JNDIProject_MSSQL_Sale_Channel_Mgnt/WEB-INF/classes.jar (j)	jndi.hibernate.jndientity.sale.channel (p)	SaleChannelBook (c)	int getAdministratorId() (m)
9						void setAdministratorId(int) (m)
10	Operations App (a)	JNDIProject_MSSQL_Operation.war (w)	JNDIProject_MSSQL_Operation/WEB-INF/classes.jar (j)	jndi.hibernate.jndientity.operations (p)	OperationBook (c)	void setSuperUserId(int) (m)
11						int getSuperId() (m)
12	Supply Chain Mgmt (a)	JNDIProject_MSSQL_Supply_Chain.war (w)	JNDIProject_MSSQL_Supply_Chain/WEB-INF/classes.jar (j)	jndi.hibernate.jndientity.supply.chain (p)	SupplyChainMgntBook (c)	int getChairmanId() (m)
13						void setChairmanId(int) (m)
14	Order Management (a)	JNDIProject_MSSQL_Order_mgnt.war (w)	JNDIProject_MSSQL_Order_mgnt/WEB-INF/classes.jar (j)	jndi.hibernate.jndientity.order (p)	OrderMgntBook (c)	void setAuthorizedUserId(int) (m)
15						int getAuthorizedUserId() (m)
16	Customer Support (a)	JNDIProject_MSSQL_Customer_Support.war (w)	JNDIProject_MSSQL_Customer_Support/WEB-INF/classes.jar (j)	jndi.hibernate.jndientity.customer.support (p)	CustomerSupportBook (c)	int getDomainId() (m)
17						void setDomainUserId(int) (m)
18	Marketing automation (a)	JNDIProject_MSSQL_Marketing_Automation.war (w)	JNDIProject_MSSQL_Marketing_Automation/WEB-INF/classes.jar (j)	jndi.hibernate.jndientity.market.automation (p)	MarketingMgntBook (c)	int getAuditorId() (m)
19						void setAuditorId(int) (m)
20	Back Office Application (a)	JNDIProject_MSSQL_Back_office.war (w)	JNDIProject_MSSQL_Back_office/WEB-INF/classes.jar (j)	jndi.hibernate.jndientity (p)	Book (c)	int getId() (m)
21						void setId(int) (m)
22	Product Pricing Mgmt (a)	JNDIProject_MSSQL_Product_Price_Mgnt.war (w)	JNDIProject_MSSQL_Product_Catalog/WEB-INF/classes.jar (j)	jndi.hibernate.jndientity.product.price (p)	ProductPriceMgntBook (c)	void setProductPricingId(int) (m)
23						int getActiveUserId() (m)
24	Inventory Management (a)	JNDIProject_MSSQL_Inventory_Mgnt.war (w)	JNDIProject_MSSQL_Inventory_Mgnt/WEB-INF/classes.jar (j)	jndi.hibernate.jndientity.inventory (p)	InventoryMgntBook (c)	int getFileManagerId() (m)
25						void setFileManagerId(int) (m)

Panoptics makes it easy for enterprises and the professional services organizations that serve them estimate quickly and accurately the effort required for software change. Accurate change estimation results in fewer overruns, better planning and budgeting, improved odds of getting a proposal approved, and happier customers.