



# GENROCKET ROI ANALYSIS

## The Impact of Synthetic Test Data Automation

GenRocket's *Synthetic Test Data Automation* provides multiple benefits to DevOps teams and delivers increased ROI at every stage of adoption and deployment. Its flexible self-service platform allows developers and testers to generate any volume or variety of synthetic data for any testing requirement on-demand. GenRocket technology has been successfully deployed across the full spectrum of testing categories.

- Component testing
- Functional testing
- API testing
- Performance testing
- Integration testing
- Regression testing

***Synthetic Test Data Automation eliminates the need for manual data creation and drastically reduces provisioning time when compared to testing with sensitive production data.***

## Real-World Impact of the GenRocket Platform

To project the ROI that your organization can achieve with GenRocket, consider the recent experience of a global financial services company that has deployed the platform. They implemented a GenRocket technology adoption program across their Card, Bank, Payments, Analytics and Enterprise teams to achieve the following goals through the broader use of synthetic data:

- Assured data privacy versus testing with sensitive production data
- Accelerated test data provisioning time for fully automated testing
- Greater testing accuracy leading to lower software defect rates
- Improved agility in responding to new application requirements

During a six-month initiative they were able to increase the adoption of scrum teams to 23 total teams and realized a dramatic increase in operational efficiency.

- Annual time savings of 1,212 hours (over 30 person-weeks per year)
- A 98% reduction in test data provisioning time (days reduced to minutes)
- A 67% improvement in test coverage (more testing with greater data variety)

Here are some examples of the test data use cases implemented by scrum teams:

- Increase combinatorial testing to cover 6,000 data permutations
- Ensure PCI compliance with synthetic data for 532 functional tests
- Simulate shopping transactions to train machine learning algorithms
- Test over 60,000 mapping rules across multiple general ledger data feeds
- Validate the rules used by a Data Loss Protection tool to ensure its effectiveness

## Reduction in Test Cycle Time

To project an ROI for your organization, the experience of the financial services company described above provides a baseline assumption for calculating annual cost savings as GenRocket is rolled out to internal DevOps teams. The table below contains average annual salaries and hourly labor costs for various DevOps staff members who typically use the GenRocket platform. Based on the mix of staff members provided in the table, a weighted labor cost of \$37 per hour will be used to project the annual cost savings.

DevOps Staff	Annual Salary	Hourly Labor Cost	Percentage of Staff	Weighted Labor Cost
Software Engineer	\$100,000	\$50	20%	\$37
Test Engineer	\$80,000	\$40	30%	
Tester	\$60,000	\$30	50%	

The financial services company cited above realized a reduction in test cycle time of 1,212 hours across 23 scrum teams during the first year. That means that each scrum team averaged just over 52 hours in time savings during the year. Based on the average time saved, and the average hourly labor cost, we can project the annual cost savings for various stages of deployment for your organization. They are given in the table below.

## Savings From Reduced Cycle Time

Number of Scrum Teams	Hours Saved per Team	Annual Time Savings	Average Hourly Cost	Annual Cost Savings
25	52	1,300	\$37	\$48,100
50		2,600		\$96,200
75		3,900		\$144,300
100		5,200		\$192,400
250		13,000		\$481,000
500		26,000		\$962,000

We believe these assumptions are conservative. **GenRocket routinely accelerates days of test data provisioning time down to just a few minutes.** This creates a dramatic reduction in test cycle time, while at the same time, increasing the number of tests that can run with a greater volume and variety of test data.

# Reduction in Software Defects

While GenRocket technology can significantly accelerate the testing process, an even greater ability is increasing the coverage and quality of testing. This can drastically reduce the number of software defects that must be corrected, either during the development and release cycle or after bugs are introduced into a production environment. In this regard, **GenRocket truly delivers on the promise of *quality at speed*.**

Here's how GenRocket can deliver significant cost savings by reducing the number of software defects. There are many estimates circulating in the enterprise software arena about the average number of bugs that occur in 1,000 Lines of Code (LOC).

In the book "[How Google Tests Software - Help me test like Google](#)," the authors describe a mid-size software development project and present Google's own numbers for code size (200K LOC), the defect rate (8 bugs per 1,000 LOC) and the average cost to fix a bug (\$1,500). Other sources estimate the average defect rate per 1,000 LOC in the range of 5 to 15 bugs. Developers in [Stack Overflow](#) report lower defect rates in the range of 1-15 per 1,000 LOC. The actual number of defects depends on the type of code, programming language, the stage of development, and the level of test automation rigor and sophistication.

**To keep the ROI analysis very conservative, we'll use an average of 2 bugs per 1,000 LOC.**

The next important factor to consider is the average cost to fix a bug. Again, the estimates cover a wide range depending on the severity of the bug and its stage in the development process. Google's average cost estimate for fixing a bug is \$1600.

[Capers Jones](#) describes a range of 3 hours to 10 hours depending on the type of bug. We believe a conservative rule of thumb is ½day of development work (4 hours) for each bug. This is the view of [Jeff Sutherland](#), one of the inventors of Scrum.

**To estimate the average cost of fixing bugs, this ROI analysis will use a conservative \$200 per bug based on 4 hours of a developer's time at the rate of \$50/hour.**

The following table shows potential cost savings for code ranging in size from 1,250K LOC to 25M LOC (see the first column). This range assumes an average scrum team contains 7 individuals, 5 of whom are developers with each developer generating 10K LOC per year. This range of application sizes allows the reader to match the projected cost savings to the scope of their own environment. The next three columns calculate the number of bugs that must be resolved each year, the time to fix them, and the labor cost measured in developer time.

## Saving From Reduced Software Defects

Size of Application	Defect Rate Bugs/ 1000 LOC	Time to Fix Hours	Developer Rate Hourly Rate	Projected Defect Reduction
	2	4	\$50	50%
LOC	Bugs	Time to Fix	Cost to Fix	Cost Savings
1,250,000	2,500	10,000	\$500,000	\$250,000
2,500,000	5,000	20,000	\$1,000,000	\$500,000
3,750,000	7,500	30,000	\$1,500,000	\$750,000
5,000,000	10,000	40,000	\$2,000,000	\$1,000,000
12,500,000	25,000	100,000	\$5,000,000	\$2,500,000
25,000,000	50,000	200,000	\$10,000,000	\$5,000,000

The last column presents the projected cost savings from defect reduction after deploying GenRocket's Synthetic Test Data Automation solution.

The GenRocket platform can easily reduce any defect rate by 50% or more through the use of advanced and intelligent data-driven testing methods. GenRocket is routinely used to test scenarios that are simply not possible with conventional test data sourced from a production database. **GenRocket allows developers and testers to design and generate any type of synthetic data to maximize test coverage.**

- All combinations and permutations of data
- Positive and negative test data values
- Boundary and edge case conditions
- Patterned and rules-based data values
- Stateful and dynamic workflow data
- Fresh data uncorrupted by prior usage
- Billions of rows of data for load and performance testing

And as stated previously, the accelerated cycle times enabled by GenRocket allow for more testing at all stages of the software development lifecycle.

# The Combined Benefits of Cycle Time Reduction and Software Defect Reduction

The table below summarizes the combined benefits of **Cycle Time Reduction** and **Defect Reduction**. It illustrates the total annual savings GenRocket can deliver for various levels of scrum team deployment and across a range of application code sizes. It illustrates the potential ROI for the *Corporate Edition* at \$50,000 (the blue portion of the table) and the *Enterprise Edition* at \$100,000 (the green portion of the table).

## Summary of Cost Savings and Projected GenRocket ROI

		Cycle Time Reduction		Defect Reduction		Total Annual Savings	ROI
	GenRocket Investment	Scrum Teams	Labor Cost Savings	LOC	Labor Cost Savings		
Corporate Edition	\$50,000	25	\$48,100	1,250,000	\$250,000	\$298,100	496%
		50	\$96,200	2,500,000	\$500,000	\$596,200	1092%
		75	\$144,300	3,750,000	\$750,000	\$894,300	1689%
Enterprise Edition	\$100,000	100	\$192,400	5,000,000	\$1,000,000	\$1,192,400	1092%
		250	\$481,000	12,500,000	\$2,500,000	\$2,981,000	2881%
		500	\$962,000	25,000,000	\$5,000,000	\$5,962,000	5862%

The annual cost savings in the table demonstrate how rapidly the return-on-investment in GenRocket technology increases with each stage of scrum team deployment and progressively larger application sizes. The GenRocket solution easily pays for itself in a matter of weeks or months. The faster and broader the GenRocket deployment grows, the greater the return-on-investment your organization will realize.