

GuardIEn Savings for CA Gen Development



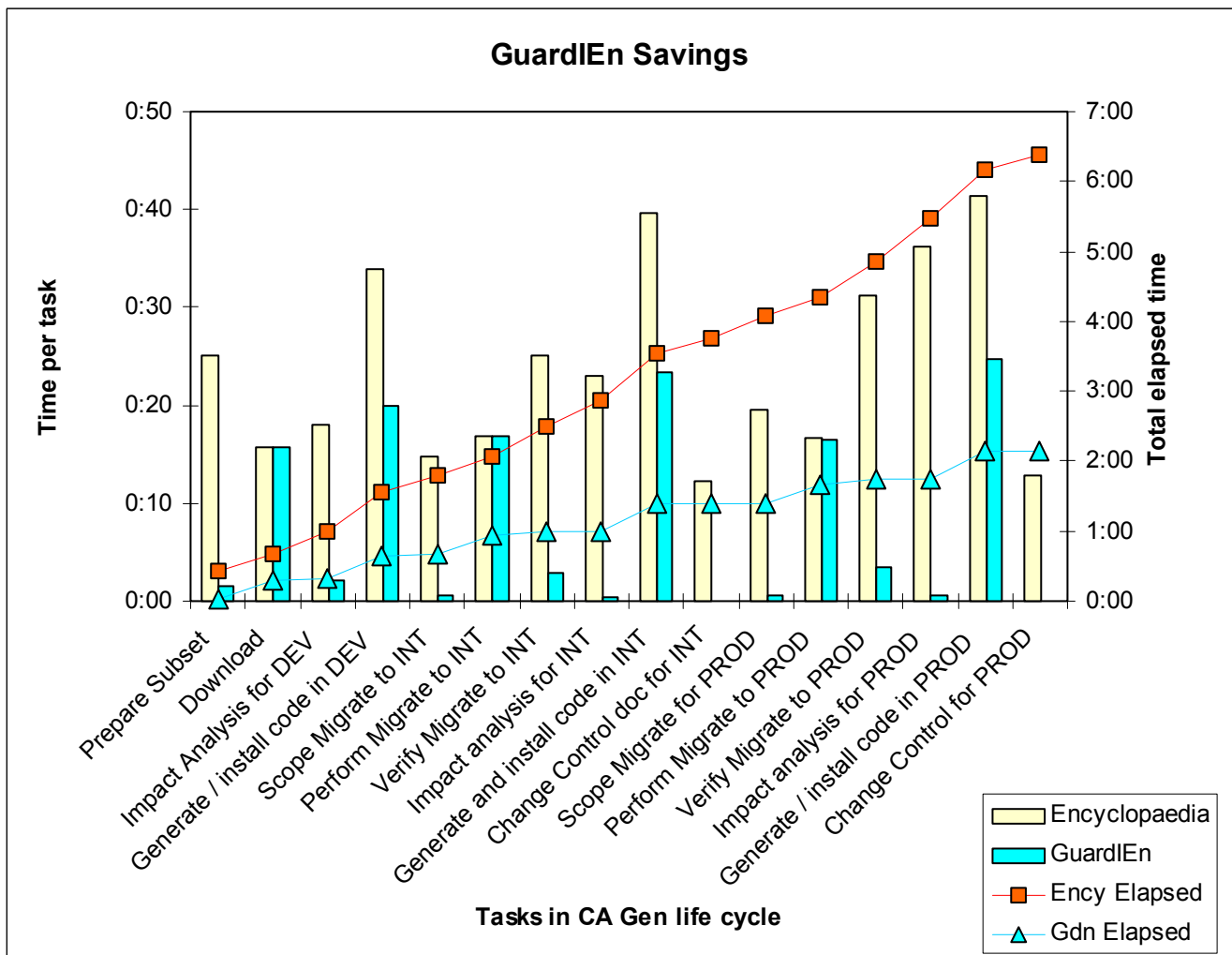
This paper quantifies the savings that can be achieved by using GuardIEn for managing the CA Gen development process.

Common tasks associated to application development using CA Gen have been performed, and timed, using the standard CA Gen tools and reports, and then repeated using GuardIEn. All of the key Gen processes (i.e. scope subset, download, impact analysis, migrate, generate and install) were managed by GuardIEn.

The results show that GuardIEn provides significant savings in time for CA Gen development.

In the example change performed, GuardIEn provided a 66% saving in elapsed time.

This supports savings reported by GuardIEn customers of approximately 50% in the effort required to implement changes when using GuardIEn.



Task	Encyclopaedia	GuardIEn	Savings
Prepare Subset & determine downgrades	0:25:11	0:01:27	94%
Perform Download	0:15:43	0:15:42	0%
Perform Impact Analysis for implementation to DEV	0:18:02	0:02:12	88%
Generate and install code in DEV	0:33:54	0:19:53	41%
Scope Migrate for promote to INT	0:14:47	0:00:36	96%
Perform Migrate to INT	0:16:51	0:16:49	0%
Verify Migrate to INT	0:25:02	0:02:51	89%
Impact analysis on INT environment	0:23:01	0:00:23	98%
Generate and install code in INT	0:39:45	0:23:19	41%
Change Control documentation for INT	0:12:16	0:00:04	99%
Scope Migrate for promote to PROD	0:19:34	0:00:40	97%
Perform Migrate to PROD	0:16:36	0:16:32	0%
Verify Migrate to PROD	0:31:08	0:03:29	89%
Impact analysis on PROD environment	0:36:12	0:00:35	98%
Generate and install code in PROD	0:41:25	0:24:42	40%
Change Control documentation for PROD	0:12:52	0:00:05	99%
Total	6:22:19	2:09:19	66%

Notes:

- 1) This example uses a simple 3-tier model architecture of DEV, INT and PROD.
- 2) The times in the table are taken from a worked example. Where necessary the reports available on the Encyclopaedia were used to cross reference data about objects. In reality many CA Gen sites do not go to this trouble and take the approach of “risk by degrees”. E.g. if unsure of the impact of a change, it is “safer” to generate the whole load module (or in some cases model) rather than run the risk of leaving something un-generated.
- 3) The list includes tasks where there is some interaction with the Encyclopaedia. Clicking changes on the Toolset and application testing are not included as measuring these tasks depends greatly on the nature of the change and the extensiveness of the testing. Some of the tasks included in the list above may experience varying degrees of savings due to “over scoping” e.g. Downloads scoped by inexperienced staff members often contain more objects than are needed, and therefore run longer. GuardIEn helps resolve this problem through the use of subsetting wizards and greater visibility of objects and associations.
- 4) The change that was measured to obtain these results was a typical change. It involved changing 12 Action Blocks and 2 Procedures to use a new Common Action Block to perform the read of an Entity Type, rather than do the read directly.
- 5) If a different model architecture is used, such as CBD, parallel development etc, then the savings to be achieved are even greater. GuardIEn includes features that specifically help with these approaches to CA Gen development, e.g. cross release migrations, shared object distribution and automated generations that source objects from multiple models within the same logical generation unit.
- 6) The savings measured are purely based on the time taken. In practice, the difficulty of reliably performing these tasks using the standard encyclopaedia utilities means that further savings are possible through the avoidance of errors and re-work when using GuardIEn.
- 7) The details of the worked example that was used to prepare this paper are available on request.