Costs/Factor	Effect	Quantified Data	Effect Size	Study	Qualitative Support
		Clone & own Strategy			
Development	Decreased	35% of reused code	-35%	[29]	[3, 36, 56]
		50–80% reused code		[31]	
Adaptation	Increased				[18]
Maintenance	Increased				[18, 36, 42]
Maintenance	Decreased				[2, 3, 36]
Productivity	Improved		+37%	[29]	[47]
Productivity	Indecisive			[23]	
Bugs identified	Reduced		-35%	[29]	
			-66.7%	[47]	
Time to market	Reduced		-30%	[47]	[2, 19, 32]
		Factors of three to five		[31]	
		Migration			
Adoption	Investment	4 PM pilot → 2 products; 12 PM → 23 products	0.5 PM/product	[5, 30]	
		36 months 80% code → 12 months 100% code → 24 months platform (tools, FM)	—	[22]	
		>3 months; tool train. 8.67%, FM 8.67%, DA 17.83%, FM 5.67%, IMPL 47.5%, OTH 11.67%		[45]	
		$235,200; 14.5 PM ($336,000) beforehand		[21]	
		4.2 PY		[57]	
		break even after eight months		[35]	
Staffing	Reduced		-66.6%	[42]	
		20 instead of 150	-86.7%	[43]	
			-75%	[48]	
Staffing	Increased				[37]
Feature development	Reduced		-50%	[5, 30]	[20, 53]
		<300 a year over 1,600 to >2,500	-88–-81%	[22]	
		40-70% reuse		[33, 34]	
		70% reuse		[48]	
			-67%	[48]	
Feature development	Equal				[37]
		42-60% code reuse		[31]	
Bugs identified	Reduced	∼80 a year down to ∼40	-50%	[22]	
			-23.6%	[47]	
Testing costs	Reduced	-$908.100		[21]	
		Saved $80 + $39 million + integration testing costs over 3 years		[27]	
			-96%	[48]	
Maintenance	Reduced	Code base from 11,985 to 10,584 SLOC	-17%	[39]	[10, 20, 41]
		-$1.98 million (removed code redundancy/cloned units)		[21]	
		Code size for component from 91,106 LOC to 31,932	-65%	[57]	
Maintenance	Equal				[37]
Derivation	Reduced	1 day → 1 hour	-95.83%	[28]	
		-$630.900 (build cost), -$334.400 (fewer build fails), -$4.28 million (distribution/scoping)		[21]	
Release	Increased	2.5 PM ($58,800) per product release (slight increase)		[21]	[27, 42]
Integration	Reduced				[28]
Productivity	Increased	improvement of 3 to 5	+200–400%	[48]	[28, 42]
Time to market	Reduced	24 → 10 months, 19 → 8, 17 → 5	-57.9%–-70.6%	[20]	[37]
		90 days estimated to be 2.5 times faster	-60%	[33, 34]	
		2 years → 3 months	-87.5%	[42]	
		3 years → 1.5 years → 3.5 months	-90.3%	[43]	
Overall	Reduced	-$3.67 million → -$4.23 million a year	-7–-3%	[21, 47]	
		Savings of 8.8 PY, ROI: 110%	-52.3%	[57]	
		break even after 3 products		[34]	
		savings of $166 million over 4 years		[26]	

Costs	Effect	Quantified Data	Effect Size	Study	Qualitative Support
		Platform Strategy			
Adoption	Investment	26 man-months (6 products), $0.3 million		[44]	[8]
		107 man-months, $1.3 million		[44]	
		2 PM		[9]	
		$3.5 million		[15]	
		break even after 2 years		[14, 24]	
		break even after 5 products		[55]	
Integration	Reduced	$0.07 million compared to $0.36 million of developing anew	-80.56%	[44]	[7, 12, 51]
		Builds in weeks instead of months		[11, 14]	
	Increased	119% of developing not for reuse	+19%	[44]	
		$12.56 per LOC compared to $10 (estimated)	+20.4%	[15]	
Development	Reduced	6,728 SLOC saved from 217 KSLOC	-3.1%	[7, 14]	[3, 4, 12, 49, 52]
		31% & 38% code reuse for one product	-38% & -31%	[44]	
		5–10% of single system (190 PM)	-90–95%	[9, 13]	
		Code size reduced by 34–88% (76% for concrete product)	-34–88%	[11]	
		Almost 80% reuse, cost ratio of 20% compared to 65%	-45%	[8]	
			-50%	[15]	
			-10–30%	[14]	
		over 75% reuse	-25%	[14]	
			-66.7%	[51]	
		effort reduced by a factor of 6 on average	-83.3%	[54]	
Development	Indecisive	Saved 43.3 man-days, lost 31.5 days (drastic exception)		[44]	
Development	Increased	111% of costs developing for reuse	+11%	[44]	
		160% of costs developing for reuse	+60%	[54]	
		150% developing assets	+50%	[4]	
		25% costs of reusing	+25%	[4]	
Maintenance	Reduced	54 man-months, $0.7 million		[44]	[3, 7, 12, 48, 55]
		99 man-months, $1.3 million		[44]	
			-60%	[54]	
			-60%	[55]	
Staffing	Reduced	10x products, 5x developers	-50%	[22]	[21, 38]
		25% of the staff	-75%	[1]	
		15 compared to 100	-85%	[11, 14]	
			-75%	[16]	
			-90%	[1]	
		50 developers in contrast to 200	-75%	[14]	
			-75%	[14]	
Dependencies	Increased				[6]
Productivity	Improved	0.7 & 1.1 KSLOC/man-month	+40% & +57%	[44]	[8]
		3.6 instead of 1 product	360%	[14]	
Productivity	Indecisive	100% LOC/man-month → 135% (1 year) → 78% (5 years)	-22–+35%	[46]	
Bugs identified	Reduced	1.3 & 2.0 bugs/KSLOC	-24% & -51	[44]	[12, 49]
			-96%	[1]	
			-90%	[11, 14]	
		defects cut in half	-50%	[1]	
			-90%	[1]	
Time to market	Reduced	21 instead of 36 months	-42%	[44]	[4, 14, 25, 52, 55]
		1/3 of the time	-66.7%	[1]	
			-50%	[11, 14]	
		From years to months		[15]	
		1 year down to 1 week	-98.1%	[1]	
		over 50% reduction	-50%	[1]	
			-71.7%	[51]	
			-50%	[54]	
		2-4 more times would be needed otherwise	-50–75%	[55]	
Quality	Reduced	Savings in second year of $0.06 million		[44]	[51]
Release	Reduced	Installation from 16 hours → 15 min	-98.4%	[38]	[14]
Overall	Reduced	80 man-months, $1 Mil; saved 328 man-months, $4.1 Mil; ROI 410%, break even: 2 years	-80.39%	[44]	[50]
		206 man-months, $2.6 Mil; saved 446 man-months, $5.6 Mil; ROI 216%, break even: 6 years	-53.81%	[44]	
			-50%	[11, 14]	
		Savings of over $300 million		[12]	
		Savings of $15 million		[15]	
		12 instead of 3 products	-75%	[1]	
		$4 million of savings a year		[1]	
		savings of $340 million over eight years		[17, 40]	

References

[1] L. Bass, P. Clements, and R. Kazman, *Software Architecture in Practice*, 1999.

[2] V. Bauer, J. Eckhardt, B. Hauptmann, and M. Klimek, "An exploratory study on reuse at Google," in *SER&IP*, 2014.

[3] V. Bauer and A. Vetro, "Comparing reuse practices in two large software-producing companies," *Journal of Systems and Software*, vol. 117, 2016.

[4] J. Bergey, S. Cohen, L. Jones, and D. Smith, "Software product lines: Experiences from the sixth DoD software product line workshop," Carnegie Melon University, Tech. Rep., 2004.

[5] BigLever, "BigLever software case study: Engenio," Tech. Rep. 2005-06-14-1, 2005.

[6] C. Bogart, C. Kästner, J. Herbsleb, and F. Thung, "How to break an API: Cost negotiation and community values in three software ecosystems," in *FSE*, 2016.

[7] G. Bowen, "An organized, devoted, project-wide reuse effort," *Ada Letters*, vol. XII, no. 1, 1992.

[8] L. Brownsword and P. Clements, "A case study in successful product line development," Carnegie Mellon University, Tech. Rep., 1996.

[9] R. Buhrdorf, D. Churchett, and C. Krueger, "Salion's experience with a reactive software product line approach," in *PFE*, 2003.

[10] P. Clements and J. Bergey, "The U.S. Army's common avionics architecture system (CAAS) product line: A case study," Carnegie Mellon University, Tech. Rep. CMU/SEI-2005-TR-019, 2005.

[11] P. Clements, S. Cohen, P. Donohoe, and L. Northrop, "Control channel toolkit: A software product line case study," Carnegie Mellon University, Tech. Rep. CMU/SEI-2001-TR-030, 2001.

[12] P. Clements, S. Gregg, C. Krueger, J. Lanman, J. Rivera, R. Scharadin, J. Shepherd, and A. Winkler, "Second generation product line engineering takes hold in the DoD," *CrossTalk – The Journal of Defense Software Engineering*, vol. 27, no. 1, 2014.

[13] P. Clements and L. Northrop, "Salion, Inc.: A software product line case study," Carnegie-Mellon University, Tech. Rep., 2002.

[14] ——, *Software Product Lines - Practices and Patterns*, 2002.

[15] S. Cohen, E. Dunn, and A. Soule, "Successful product line development and sustainment: A DoD case study," Carnegie Mellon University, Tech. Rep. CMU/SEI-2002-TN-018, 2002.

[16] A. Cortiñas, M. Luaces, O. Pedreira, Á. Places, and J. Pérez, "Web-based geographic information systems SPLE: Domain analysis and experience report," in *SPLC*, 2017.

[17] M. Dillon, J. Rivera, and R. Darbin, "A methodical approach to product line adoption," in *SPLC*, 2014.

[18] Y. Dubinsky, J. Rubin, T. Berger, S. Duszynski, M. Becker, and K. Czarnecki, "An exploratory study of cloning in industrial software product lines," in *CSMR*, 2013.

[19] A. Duc, A. Mockus, R. Hackbarth, and J. Palframan, "Forking and coordination in multi-platform development: A case study," in *ESEM*, 2014.

[20] C. Ebert and M. Smouts, "Tricks and traps of initiating a product line concept in existing product," in *ICSE*, 2003.

[21] D. Faust and C. Verhoef, "Software product line migration and deployment," *Software, Practice & Experience*, vol. 33, no. 10, 2003.

[22] T. Fogdal, H. Scherrebeck, J. Kuusela, M. Becker, and B. Zhang, "Ten years of product line engineering at Danfoss: Lessons learned and way ahead," in *SPLC*, 2016.

[23] W. Frakes and G. Succi, "An industrial study of reuse, quality, and productivity," *Journal of Systems and Software*, vol. 57, no. 2, 2001.

[24] C. Gacek, P. Knauber, K. Schmid, and P. Clements, "Successful software product line development in a small organization – a case study," Fraunhofer IESE, Tech. Rep. 013.01/E, 2001.

[25] C. Ganz and M. Layes, "Modular turbine control software: A control software architecture for the ABB gas turbine family," in *ARES*, 1998.

[26] S. Gregg, R. Scharadin, and P. Clements, "The more you do, the more you save: The superlinear cost avoidance effect of systems product line engineering," in *SPLC*, 2015.

[27] S. Gregg, R. Scharadin, E. LeGore, and P. Clements, "Lessons from AEGIS: Organizational and governance aspects of a major product line in a multi-program environment," in *SPLC*, 2014.

[28] M. Ham and G. Lim, "Making configurable and unified platform, ready for broader future devices," in *ICSE*, 2019.

[29] E. Henry and B. Faller, "Large-scale industrial reuse to reduce cost and cycle time," *IEEE Software*, vol. 12, no. 5, 1995.

[30] W. Hetrick, C. Krueger, and J. Moore, "Incremental return on incremental investment: Engenio's transition to software product line practice," in *OOPSLA*, 2006.

[31] A. Incorvaia, A. Davis, and R. Fairley, "Case studies in software reuse," in *CMPSAC*, 1990.

[32] S. Jansen, S. Brinkkemper, I. Hunink, and C. Demir, "Pragmatic and opportunistic reuse in innovative start-up companies," *IEEE Software*, vol. 25, no. 6, 2008.

[33] P. Jensen, "Experiences with product line development of multi-discipline analysis software at Overwatch Textron Systems," in *SPLC*, 2007.

[34] ——, "Experiences with software product line development," *CrossTalk – The Journal of Defense Software Engineering*, vol. 22, no. 1, 2009.

[35] H. Jepsen, J. Dall, and D. Beuche, "Minimally invasive migration to software product lines," in *SPLC*, 2007.

[36] C. Kapser and M. Godfrey, ""Cloning Considered Harmful" considered harmful: Patterns of cloning in software," *Empirical Software Engineering*, vol. 13, no. 6, 2008.

[37] R. Kolb, I. John, J. Knodel, D. Muthig, U. Haury, and G. Meier, "Experiences with product line development of embedded systems at Testo AG," in *SPLC*, 2006.

[38] C. Krueger, D. Churchett, and R. Buhrdorf, "HomeAway's transition to software product line practice: Engineering and business results in 60 days," in *SPL*, 2008.

[39] E. Kuiter, J. Krüger, S. Krieter, T. Leich, and G. Saake, "Getting rid of clone-and-own: Moving to a software product line for temperature monitoring," in *SPLC*, 2018.

[40] J. Lanman, R. Darbin, J. Rivera, P. Clements, and C. Krueger, "The challenges of applying service orientation to the U.S. Army's live training software product line," in *SPLC*, 2013.

[41] K. Lee, K. Kang, E. Koh, W. Chae, B. Kim, and B. Choi, "Domain-oriented engineering of elevator control software," in *SPLC*, 2000.

[42] D. Li and C. Chang, "Initiating and institutionalizing software product line engineering: From bottom-up approach to top-down practice," in *ICSAC*, 2009.

[43] D. Li and D. Weiss, "Adding value through software product line engineering: The evolution of the FISCAN software product lines," in *SPLC*, 2011.

[44] W. Lim, "Effects of reuse on quality, productivity, and economics," *IEEE Software*, vol. 11, no. 5, 1994.

[45] J. Martinez, X. Tërnava, and T. Ziadi, "Software product line extraction from variability-rich systems: The Robocode case study," in *SPLC*, 2018.

[46] M. Nagamine, T. Nakajima, and N. Kuno, "A case study of applying software product line engineering to the air conditioner domain," in *SPLC*, 2016.

[47] J. Otsuka, K. Kawarabata, T. Iwasaki, M. Uchiba, T. Nakanishi, and K. Hisazumi, "Small inexpensive core asset construction for large gainful product line development: Developing a communication system firmware product line," in *SPLC*, 2011.

[48] K. Pohl, G. Böckle, and F. van Der Linden, *Software Product Line Engineering: Foundations, Principles and Techniques*, 2005.

[49] G. Quilty and M. Ó. Cinnéide, "Experiences with software product line development in risk management software," in *SPLC*, 2011.

[50] D. Rine and R. Sonnemann, "Investments in reusable software. A study of software reuse investment success factors," *Journal of Systems and Software*, vol. 41, no. 1, 1998.

[51] D. Sharma, A. Aurum, and B. Paech, "Business value through product line engineering – a case study," in *SEAA*, 2008.

[52] O. Slyngstad, A. Gupta, R. Conradi, P. Mohagheghi, H. Rønneberg, and E. Landre, "An empirical study of developers views on software reuse in Statoil ASA," in *ISESE*, 2006.

[53] M. Staples and D. Hill, "Experiences adopting software product line development without a product line architecture," in *APSEC*, 2004.

[54] F. van der Linden, "Philips healthcare compositional diversity case," in *Systems and Software Variability Management*, 2013.

[55] F. van der Linden, K. Schmid, and E. Rommes, *Software Product Lines in Action - The Best Industrial Practice in Product Line Engineering*, 2007.

[56] R. Walker and R. Cottrell, "Pragmatic software reuse: A view from the trenches," University of Calgary, Tech. Rep. 2016-1088-07, 2016.

[57] G. Zhang, L. Shen, X. Peng, Z. Xing, and W. Zhao, "Incremental and iterative reengineering towards software product line: An industrial case study," in *ICSM*, 2011.