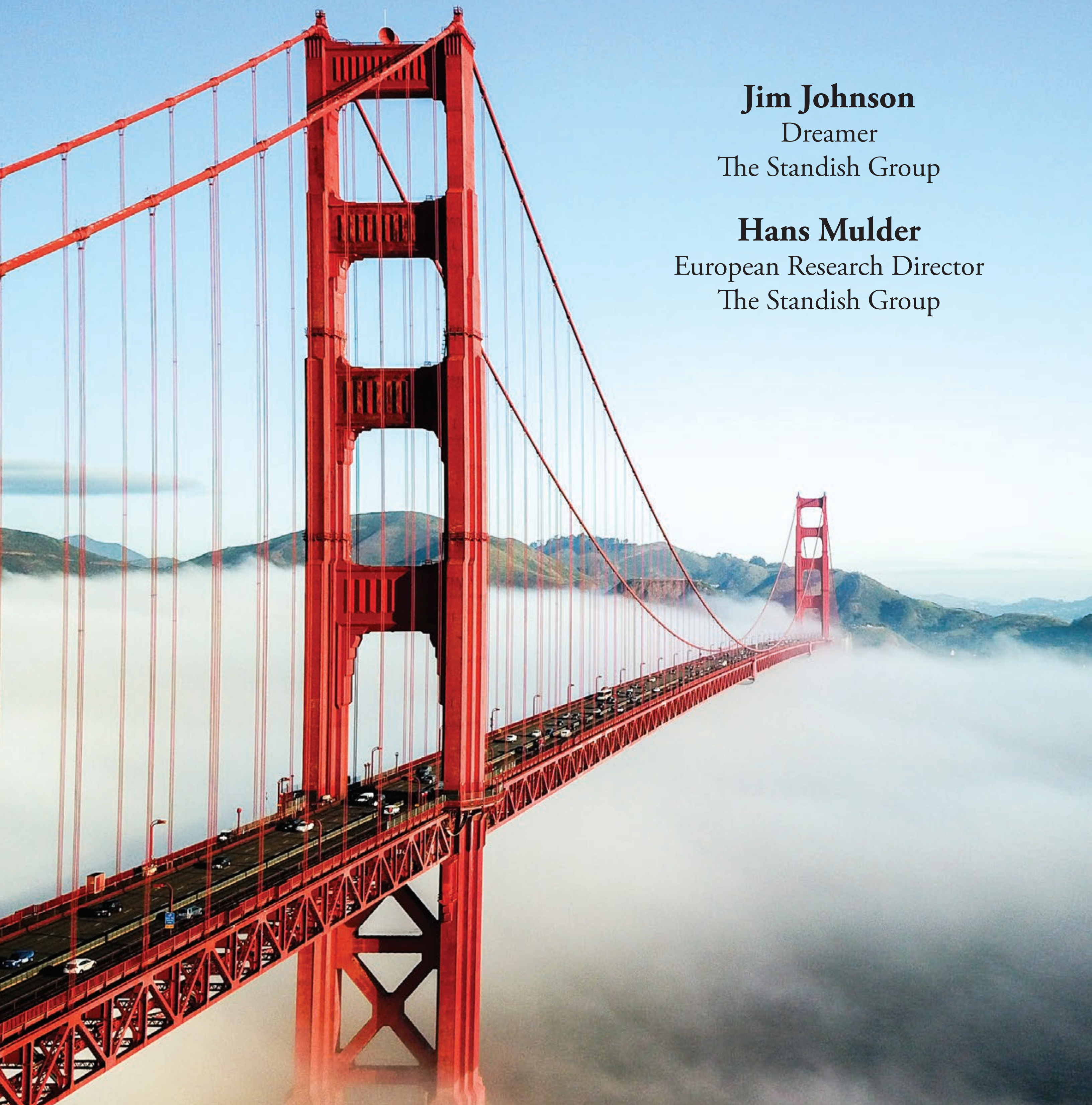
ENDLESS MODERNIZATION:

How Infinite Flow Keeps Software Fresh

Jim Johnson
Dreamer
The Standish Group

Hans Mulder
European Research Director
The Standish Group



INTRODUCTION

Keeping the Golden Gate Bridge modern is a permanent full-time job for a team of 30 painters and mechanics. Every working day, this team helps the bridge maintain its flexibility in the face of assaults from the strong winds off San Francisco Bay. The team also continually paints the bridge to preserve its majestic blood-orange glow. Keeping mission-critical software applications fresh and glowing requires the same dedication.

EXECUTIVE SUMMARY

This paper is the result of unique and extensive research for identifying the best methods for modernizing mission-critical application software. We have concluded that projects using the principles of what we call "Infinite Flow" (a continuous process, unlike projects that are designed as one-time events with a beginning and an end) achieve superior results. (For more information, see addendum, "About Infinite Flow.") As a result of these compelling findings, we are providing a method for transition from a project-based environment to non-project activities by adopting the principles of Infinite Flow to modernize legacy applications.

Since Infinite Flow is not a project, we cannot measure it. Therefore, we have reduced the observations from all modernization projects to those that generally meet the attributes of microprojects and some attributes of Infinite Flow (Flow), such as Scrum and DevOps. We call this "flow like modernization" in order to compare it more clearly with the other types of application modernization that we have observed. In essence, we have found that great progress can be made by ending project-based modernization in favor of the flow process.

Our findings show:

a. Flow like modernization projects that are a series of microservices or microprojects, rather than one large project, achieve much better outcomes.

b. Flow like microprojects reap greater customer satisfaction because of the built-in user/customer feedback loop.

c. Flow like modernization microprojects achieve a higher return of value.

d. Flow like microprojects offer a reduced risk of failure and monetary loss.

e. Flow like microprojects have a higher degree of sustainable innovation.

f. Flow achieves a longer lifespan of applications, avoiding premature retirements.

CHAOS RESEARCH AND DATA

For the last 25 years, The Standish Group has collected, adjudicated, and approved between 2,500 and 5,000 new project cases per year. In each of those 25 years, we have added and changed observations to gain a better understanding of why some projects are successful while others fail. We also research many other issues having a bearing on project success; these have led to books and workshops on how to be a good sponsor, a good team, and a good workplace. Most importantly, we have pinpointed the issue of decision latency as crucial to project success.

Our organization profiles, which inform the Standish database, study organizations from the viewpoint of 24 separate data points. Our project profiles depend on more than 80 data points through which projects are analyzed and assessed, and on a dozen worksheets developed for each project. The database is used to create reports like this one. It's also used for projects like "CHAOS 2020: Beyond Infinity," as well as general queries, single-project assessments, future portfolio predictions, and performance benchmarks. Our database is robust, so we can see many different views of data and measure attributes such as OnTime, OnBudget, OnTarget,

Copyright © 2020 by The Standish Group International, Inc.

OnGoal, Value, and Customer Satisfaction. We can also use any of these measurements for decision speed, team capability, size of projects, types of projects, methods of delivery, and a number of other common applications.

In Chart 1 we see success by the traditional metric of OnTime, OnBudget, and OnTarget over the last 25 years. After 25 years of research, the only sure way to prevent software projects from being either challenged or failed is to stop doing projects and adopt Infinite Flow.

CHART 1: LONGITUDINAL CHAOS RESEARCH

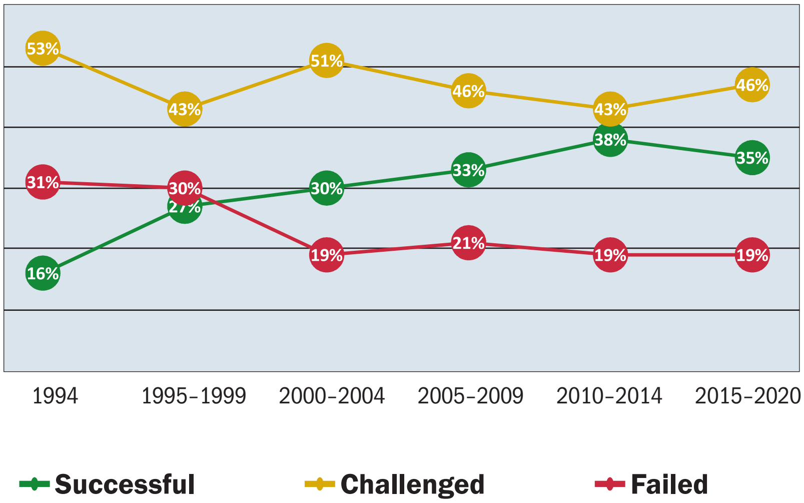


CHART 2: MODERN RESOLUTION BY PROJECT TYPE

PROJECT TYPE	SUCCESSFUL	CHALLENGED	FAILED
Developed from scratch	26%	54%	20%
Developed using components	37%	46%	17%
Purchased application (COTS)	44%	36%	20%
Flow Like Modernization	71%	28%	1%

The "modern" definition of success (OnBudget, OnTime, Customer Satisfied) also shows that Flow modernization projects do much better than other project types. This chart is based on the 50,000 projects in the CHAOS 2020 database.

This paper builds on our research, with an introduction to Infinite Flow and the discovery of decision latency as the root cause of poor project performance. That research includes our 2010 paper, "Modernization: Clearing a Pathway to Success," [1] which identified three common ways organizations go about replacing a current software application and compared the results of each method. Type 1 was essentially a new application development. It involved developing new user requirements, technical specifications, programming, tests, and user education, as well as a host of other project management and executive activities. Type 2 used a typical commercial off-the-shelf (COTS) package or implementation that required moderate modifications. Type 3 used a method we called "modernization": refurbishing an existing application rather than replacing it fully.

For the current paper, we have added a further category of application replacement, using a collection of software components. We have also replaced "modernization" with our discovery of "flow-like modernization."

In 2014, we published a second research paper, "Modernization in Place," [2] which presented a technique with which to modernize an application or a system while it is fully functional and being used by an organization and its customers. Infinite Flow again advances this concept by making changes a daily event.

Copyright © 2020 by The Standish Group International, Inc.

DECISION LATENCY THEORY

Decision Latency theory states: "The value of the interval is greater than the quality of the decision."[4] The Standish Group has determined that the root cause of software project failures and challenges is slow decision latency, and that to improve project performance, organizations need to find ways to make decisions faster. The problem is, "good project management" actually tends toward slower decisions out of fear of making a bad one. This seemingly wise practice causes *fragility* [FN2] in the pursuit of a better outcome. True, short decision intervals are likely to result in some minor errors—but being able to "tinker" with the project makes the whole process *antifragile*. Short decision latency also promotes agility and thus decreased future technical debt. Overall, learning to make decisions quickly results in faster time to market and additional savings.

For each of the three application-replacement techniques—development of a completely new application, implementation of a commercial off-the-shelf (COTS) package, or the use of a number of software components—The Standish Group has arrived at a formula for number of decisions to be made per thousand dollars invested. We developed an overall average of one decision per thousand dollars. Tellingly, however, Flow modernization projects came in at about half the decisions of the average—in other words, one decision per $2,000. Why? Consider that many of the decisions to be made during a project center around features and functions. However, modernization projects already have a feature and function role model, and this, of course, reduces the number of decisions to be made. These role models also improve the "dwell time" needed for many of those decisions. In addition, Flow modernization teams are usually self-directed and can make decisions on their own, further reducing the interval.

CHART 3 RESOLUTION BY DECISION LATENCY SKILLS

SKILL LEVEL	SUCCESSFUL	CHALLENGED	FAILED
Highly Skilled	63%	30%	7%
Skilled	28%	61%	11%
Moderately Skilled	20%	51%	29%
Poorly Skilled	18%	47%	35%

Teams that have better decision latency skills are also more successful. This chart uses the modern definition of success (OnBudget, OnTime, satisfied customer) and is based on the 50,000 projects in the CHAOS 2020 database.

Copyright © 2020 by The Standish Group International, Inc.

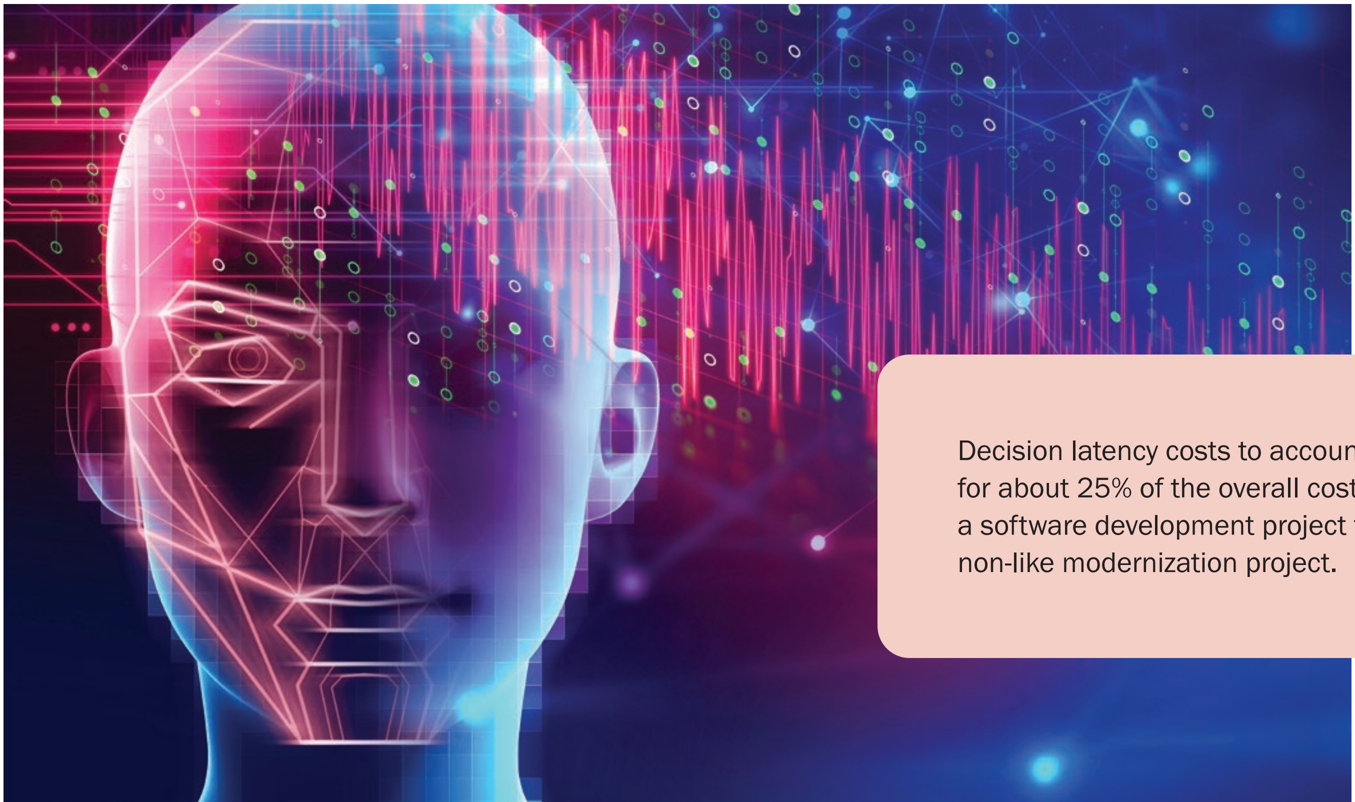
In general, our findings found decision latency costs to account for about 25% of the overall cost of a software development project. For example, a million-dollar modernization project will require about 500 decisions, and the average interval or dwell time is less than one hour, making the estimated cost around $50,000. In contrast, a typical million-dollar new application development project will require about 1,100 decisions at a dwell time of two hours. The cost of those decisions will come in at $220,000—more than four times the cost of a Flow Like modernization project.

We believe improved decision latency is the main reason Flow like modernization projects enjoy a greater success rate and cost less than other types of projects. The Standish Group measures the skill level by information in the project profile.

CHART 4: DECISION LATENCY BY PROJECT TYPE

PROJECT TYPE	HIGHLY SKILLED	SKILLED	MODERATELY SKILLED	POORLY SKILLED
Developed from scratch	17%	38%	27%	18%
Developed using components	21%	39%	23%	17%
Purchased application (COTS)	22%	36%	26%	16%
Flow Like Modernization	27%	43%	18%	12%

Flow Like modernization project teams generally display better decision latency skills than teams on other types of projects. This chart is based on the 50,000 projects in the CHAOS 2020 database.



Decision latency costs to account for about 25% of the overall cost of a software development project for non-like modernization project.

CUSTOMER SATISFACTION

The real test of a successful software project is customer satisfaction—it's one of the major reasons we create and modernize software. There's a direct relationship between customer satisfaction and that customer's ability to absorb change. Large, "big bang" projects produce only a 6% rate of high satisfaction, and they produce a 60% rate of customer disappointment. Small, iterative modernization projects reap four times that rate of high satisfaction and produce only an 8% rate of customer disappointment. Retaining similarity in a transition to new features and functions avoids the necessity for retraining. Being able to make changes to the user experience that are intuitive increases user productivity. This is further spelled out in our Absorption Theory [FN3].

CHART 5: CUSTOMER SATISFACTION BY PROJECT TYPE

PROJECT TYPE	VERY SATISFIED	SATISFIED	SOMEWHAT SATISFIED	NOT SATISFIED	DISAPPOINTED
Developed from scratch	12%	18%	25%	22%	23%
Developed using components	14%	21%	25%	19%	21%
Purchased application (COTS)	13%	21%	26%	21%	19%
Flow Like Modernization	21%	34%	30%	9%	6%

Flow Like modernization projects attain a higher level of customer satisfaction than other project types. This chart is based on the 50,000 projects in the CHAOS 2020 database.

Large, "big bang" projects produce only a 6% rate of high satisfaction, and they produce a 60% rate of customer disappointment.

Copyright © 2020 by The Standish Group International, Inc.

VALUE OF FLOW MODERNIZATION

If you buy a car today but cannot drive it for two years, you obviously get no value for that period of time. In addition, the car will depreciate in value by about 24% before you do drive it. It's the same for the delivery of new or improved software features and functions. However, Flow modernization allows you to build a new software function today—and enjoy the return on value tomorrow. Value is the other measurement of Flow.

Our research shows that as much as 80% of new software features and functions are dead on arrival and rarely used. However, Flow modernization eliminates many of those features and functions before they are even created. Flow modernization also eliminates many unused features and functions through refactoring—which improves software maintenance costs, thus creating additional value. The proof is in the numbers: Flow like modernization returns twice the value on average than for other types of software development.

CHART 6: VALUE BY PROJECT TYPE

PROJECT TYPE	VERY HIGH VALUE	HIGH VALUE	AVERAGE VALUE	LOW VALUE	VERY LOW VALUE
Developed from scratch	8%	19%	32%	15%	26%
Developed using components	11%	22%	32%	13%	22%
Purchased application (COTS)	10%	21%	31%	12%	26%
Flow Like Modernization	22%	32%	30%	9%	7%

Flow Like modernization projects offer a much higher return on investment/value than for other project types. This chart is based on the 50,000 projects in the CHAOS 2020 database.

Flow Like modernization returns twice the value on average than for other types of software development.

PROJECT ENVIRONMENT

The Standish Group has reduced the "Factors of Success" to these three practices: the good place, the good team, and the good sponsor. The place is the project environment. The environment is the people, places, and things, such as tools, through which the project is executed—basically, everything but the team and the sponsor. An environment that values skill in its employees and is attentive to its other attributes improves the chances of a successful project; a poorly skilled environment presents challenges. The Standish Group has identified 10 principles and 50 skills that go into making a "good place." The Good Place skills are listed in the current CHAOS2020: Beyond Infinity Report [10]. The Standish Group also has an online appraisal to measure the skills of The Good Place. Setting up the right culture to support Flow is essential, that is why we have created a set of principles, practices, and skills as a recommendation for a Flow Modernization environment.

CHART 7: PROJECT ENVIRONMENTAL SKILLS BY PROJECT TYPE

PROJECT TYPE	HIGHLY SKILLED	SKILLED	MODERATELY SKILLED	POORLY SKILLED
Developed from scratch	7%	49%	39%	5%
Developed using components	8%	50%	37%	5%
Purchased application (COTS)	8%	49%	39%	4%
Flow Like Modernization	12%	56%	29%	3%

Flow Like modernization projects tend to have project environments that are slightly more skilled places than do other project types. This chart is based on the 50,000 projects in the CHAOS 2020 database.

Copyright © 2020 by The Standish Group International, Inc.

SKILLED STAFF

Our continuous research on project success shows successful projects need smart, trained people. Not surprisingly, one of the key project success factors that Standish has identified since the beginning of its CHAOS research is a competent staff [5]. However, as with any team, Flow modernization teams improve with time, experience, feedback, and collaboration. Fortunately, the Flow modernization process promotes rapid feedback, experimentation, small failures, and rapid customer engagements with actual use. Also fortunately, Flow modernization teams tend to be smaller, at around four to six team members, and this improves agility and communication. Our book *The Good Mate* offers advice, insights, and practical approaches to improve teamwork and teams [6]. The nice thing about Flow is that you can work on a task while improving skills for future tasks without losing valuable production time. In addition, special tasks can be easily augmented by temporary gig members.

CHART 8: TEAM SKILLS BY PROJECT TYPE

PROJECT TYPE	GIFTED	TALENTED	COMPETENT	ABLE	UNSKILLED
Developed from scratch	10%	31%	40%	13%	5%
Developed using components	12%	30%	39%	14%	6%
Purchased application (COTS)	11%	32%	42%	12%	4%
Flow Like Modernization	14%	26%	45%	12%	3%

Flow Like modernization projects tend to have slightly more skilled teams than do other project types. This chart is based on the 50,000 projects in the CHAOS 2020 database.

PROJECT SPONSOR

Key to success for any project is a skilled sponsor, and our book *The Good Sponsor* [7] identifies 10 principles and 50 skills that characterize good sponsors. We believe that unlike teams, which improve with time, sponsors tend to lose interest over time, and hence their skills degrade. Flow modernization projects come in at about average in terms of project sponsor skills. However, maintaining rapid and constant Flow modernization deliveries helps maintain sponsor interest, and changing sponsors as a project's focus changes also helps. Of course, the most effective thing you can do to improve sponsor skills is to encourage them to take a course, like our Good Sponsor Crash Course.

CHART 9: PROJECT SPONSOR SKILLS BY PROJECT TYPE

PROJECT TYPE	HIGHLY SKILLED	SKILLED	MODERATELY SKILLED	POORLY SKILLED
Developed from scratch	16%	39%	28%	17%
Developed using components	21%	38%	25%	16%
Purchased application (COTS)	22%	35%	27%	16%
Flow Like Modernization	25%	43%	20%	12%

Skill levels for sponsors of Flow Like modernization projects come in slightly higher than for other project types. This chart is based on the 50,000 projects in the CHAOS 2020 database.

Flow Like modernization projects that have a highly skilled project sponsor have an 81% success rate.

SUCCESS LADDER BENCHMARK

The Success Ladder benchmark offers a quick and easy way to get a rough order of magnitude for a single project's chances of success. The benchmark is part of the premium member service. By completing 7 project profile questions the system will return an estimate of success The benchmark returns a result based on answers to questions ranked against answers given for the 50,000 project profiles in the CHAOS database. By changing the answers, you can see the effect of the change. For example, you answer that the skills of the project sponsor are poor. You then look at how the chances of success change by selecting highly skilled. Then you can decide if you want to either change the sponsor to a highly skilled sponsor or provide education to improve the project sponsor.

Currently, the Success Ladder has six rungs. Rung 1 is the actuary or project type (i.e., developed from scratch or through modernization) and the project method (i.e., agile or waterfall). Rung 2 considers project size; the results change based on our labor cost range. Rung 3 measures complexity. Rung 4 measures the environment in which the project takes place. Rung 5 considers the emotional maturity of the team, and Rung 6 measures the skills of the sponsor. Creating a Success Ladder benchmark is a simple task and only takes a few minutes.

CHART 10: SUCCESS LADDER BENCHMARK FOR FLOW LIKE MODERNIZATION

SKILLS	LEVEL	SUCCESSFUL	CHALLENGED	FAILED
Good Sponsor	Highly	87%	12%	1%
Good Team	Highly	82%	15%	3%
Good Place	Highly	79%	20%	1%
Complexity	Very Easy	66%	31%	3%
Size	Small	62%	35%	3%
Base	Actuary	54%	37%	9%

Charts 10 and 11 present results for the same project, using two methods: Flow modernization, and the development of an entirely new application. Chart 10 attacks the project using small, iterative, agile modernization methods and features high levels of skills. The project is broken into 10 individual projects at about $1 million each.

Chart 11 presents the complete opposite—a large, very complex, waterfall software development project featuring moderate skill levels and running at $10 million in total.

The results seen in Chart 10 show an extremely positive outcome (OnTime, OnBudget, with a satisfied customer). In contrast, predictions for the project shown in Chart 11 come in at a mere 1% chance of success.

Copyright © 2020 by The Standish Group International, Inc.

CHART 11: SUCCESS LADDER BENCHMARK FOR NON-FLOW LIKE MODERNIZATION

SKILLS	LEVEL	SUCCESSFUL	CHALLENGED	FAILED
Good Sponsor	Moderately	1%	56%	43%
Good Team	Moderately	1%	56%	43%
Good Place	Moderately	2%	51%	46%
Complexity	Very Complex	3%	54%	42%
Size	Grand	5%	57%	37%
Base	Actuary	12%	61%	27%

EXAMPLE

Flow Like modernization is not just a theory or a wild idea. It works for developing successful software projects. Take, for example, a large European financial organization that used the method to replace a 40-year-old system. The organization had found that the increasing complexity of the interdependencies between infrastructure software and application software components was preventing agility and leading to longer lead times. These are major risks because of the number of people involved, the dependencies between projects, and issues of lifecycle management and user training and acceptance. The institution had already tried a couple of times to do a full replacement—once with a new development and once with a package. In both cases, the projects failed, at a cost of many millions of dollars. Using small projects instead, with some Flow-like techniques, kept many dependencies in place while gradually eliminating them. The effort has taken several years, but at reduced cost with no down time. This organization is also saving about 80% on its annual software maintenance by reducing most of the technical debt.

This large European organization is saving about 80% on its annual software maintenance by reducing most of the technical debt.

Copyright © 2020 by The Standish Group International, Inc.

THE FUTURE IS FLOW

The world is getting faster every day. Attention spans are measured in the space of a "tweet." Executives are impatient and want everything NOW. Timeframes for projects have gone from years to months, to weeks, and sometimes days. The next step we foresee is no projects at all—and tasks will be measured in hours for the majority of software development and implementation activity. This is a major culture change that fits into today's business environment. The demand is for more value, higher customer satisfaction, and lower costs. It is our opinion—based on our extensive research and observation of role models—that the move to Infinite Flow satisfies all three of these demands.

Infinite Flow greatly reduces overhead. A typical waterfall-type modernization project will incur about 80% in overhead and offer a net value of less than 20%. The average net value of such a project is derived from the mere 16% of features and function that are frequently used. Most of the overhead that results comes from command-and-control efforts such as gates, steering committees, and meetings that increase decision latency.

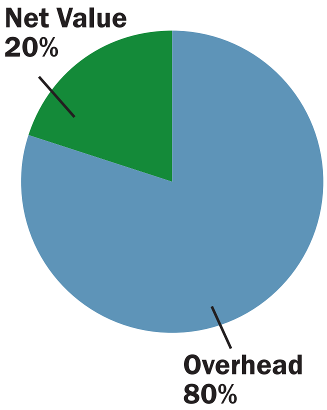
The typical agile-type modernization project will cut that overhead in half and triple the net value. This increase in value is the direct result of self-managed teams, which allow faster decisions to be made. The Infinite Flow method results in an overhead cost of only 20%, and a net value of 80%.

Infinite Flow greatly increases customer satisfaction. A typical waterfall-type modernization project will result in a high level of customer satisfaction in perhaps 20% of customers, at best. Much of this unhappiness is caused by the low level of feature function absorption, which reflects customer frustration with a difficult change management process. However, the use of agile doubles the satisfaction level in customers. Much of this increase is the result of faster delivery of features and functions that embrace change and allow for greater user absorption. Infinite Flow daily delivery and value increases customer satisfaction to a level of about 80%.

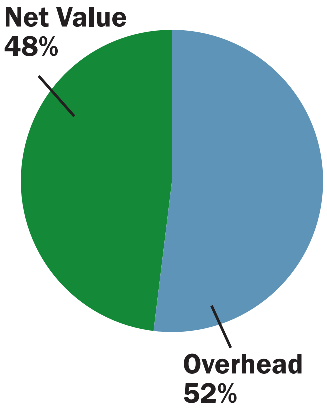
Infinite Flow also greatly decreases costs. A typical waterfall-type modernization project might cost $12 million in direct labor. For the same features and functions delivered, an agile approach would cut that cost in half. The same features and functions using Infinite Flow practices would cut the costs in half again—in other words, a cost of about $3 million.

The great thing about Infinite Flow is that it involves no estimates or budget items, no project plans, no steering committees, and no deadlines. There are no conflicts on what should be included and what should not be included. There are no project failures or "challenged" projects: There are no projects. Budgets are based on teams, and tasks are based on obvious needs.

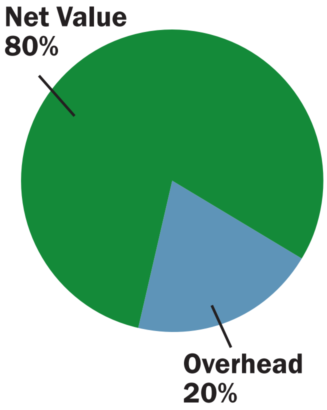
WATERFALL PROJECTS



AGILE PROJECTS



INFINITE FLOW



Copyright © 2020 by The Standish Group International, Inc.

IN SUMMARY

Starting a Flow modernization group is not a big investment or decision. You can begin with a small team of four to six members; this is the best size to begin a test project. Make sure the team has the right technical skills for the Flow-like project. Train each team member in our Good Mate program. Choose a Scrum master or a DevOps professional to guide the team. Encourage the sponsor to take our Good Sponsor class and find a certified Scrum product owner to help that sponsor. Choose a project team that has executed projects within four to six months. Choose a small application that is highly visible and offers both good value and visibility. Every day your permanent team will deliver something, and every day your team—as well as the software application—will improve. You will break the cycle of technical debt.

Flow involves five principles: (1) Only deliverables count; (2) Have a good sponsor; (3) Have a good team; (4) Have a good place to work; (5) Always promote antifragility. We further break down these principles into practices and skills.

One way to think of Flow is as a slim version of Scrum/DevOps—without the project distinction and just doing daily tasks.

The team on the Golden Gate bridge works on that bridge—every day—but that work doesn't stop any of the 12,000 vehicles from using it—every day.

ADDENDUM – ABOUT INFINITE FLOW

The Flow process continuously manages software modernization, implementation, and maintenance . The Flow process is a service-oriented method that incorporates many of the features and functions of agile/ Scrum and DevOps [3] to reduce the friction and delays associated with traditional software project development methods. Flow also reduces technical debt [FN1] by a continuous refactoring in order to keep software fresh and usable, thus reducing the high cost of maintenance. Flow pipelines are a single budget item and eliminate the high overhead costs associated with traditional project and change management.

The Flow structure is broken down into a Flow sponsor and a team. A single sponsor supports teams of producers and actors. The sponsor provides inspiration, direction, and imagination. The sponsor is ultimately responsible for the work product of the teams.

Producers define the work, manage the backlog, and present activities to actors. Producers are a combination of former project managers, business analysts, Scrum product owners, and subject matter experts. Producers generally work in teams of two to four people. Their main job is to break down the work into daily microservices that actors can complete in one day. They are developers, installers, security experts, and people who work in quality assurance and computer operations. They perform and implement the work the producers give them. Actors work in teams, of which three or four are full-time workers and one or two are available on demand. Both producers and actors work in self-directed teams.

Copyright © 2020 by The Standish Group International, Inc.

FOOTNOTES

FN1: *Technical debt* is the cost to maintain existing applications, features, functions and even hardware that have lost their value and no longer serve a useful purpose for the organization. Technical debt is depicted in rising IT costs, but it's often hard to recognize within the general IT inventory, and even harder to eliminate. The tradeoff is often to just pay the debt and then invest in new application development. However, this creates a cycle of ever-increasing technical debt, which leads to bigger IT budgets and increased staff.

FN2: *Fragility* is a hallmark of systems that fail to thrive as a result of stressors, shocks, volatility, noise, mistakes, faults, attacks, or failures. In software applications, the goal has always been to make systems resilient as well as robust. However, this concept has actually made software harder to modernize or keep current because of the ever-growing base of code; this increases technical debt and fragility. Killing off unused features and functions and adding new functions allows applications to "wander" more naturally and makes them less fragile. It's a concept that is fundamentally different from what we think of as "resiliency" [8].

FN3: *Absorption Theory* comprises three broad concepts—continuous change, decreasing complexity, and conservation of familiarity—that will increase the adoption of features and functions in software projects [9]. Absorption Theory describes the ability of an organization to successfully grasp business and technical changes without disruption.

Copyright © 2020 by The Standish Group International, Inc.

REFERENCES

[1] The Standish Group, "Modernization: Clearing a Pathway to Success" (Standish Group, 2010). Can be accessed by Standish general members in the research report section of the Standish website.

[2] The Standish Group, "Modernization in Place" (Standish Group, 2014). Can be accessed by Standish general members in the research report section of the Standish website.

[3] Jim Johnson and Hans Mulder, "Go with the Flow: Envisioning a Successful Pipeline of Software Projects" (Standish Group, 2019). Can be accessed by Standish general members in the research report section of the Standish website.

[4] Jim Johnson, "Decision Latency Theory: CHAOS Report 2018" (Standish Group, 2018). Can be accessed by Standish premium members in the premium report section of the Standish website.

[5] The Standish Group, "CHAOS Report 2016: The Winning Hand" (Standish Group, 2016). Can be accessed by Standish premium members in the premium report section of the Standish website.

[6] Jim Johnson and Evan Sorensen, *The Good Mate* (Standish Group, 2018). Available in the Standish store.

[7] Jim Johnson, *The Good Sponsor* (Standish Group, 2016). Available in the Standish store.

[8] Nassim Nicholas Taleb, *Antifragile: Things That Gain from Disorder* (Random House, 2014).

[9] The Standish Group, "CHAOS Manifesto 2015, The Law of Diminishing Returns" (Standish Group, 2015). Can be accessed by Standish general members in the CHAOS report section of the Standish website.

[10] Jim Johnson, "CHAOS2020 Beyond Infinity" (Standish Group, 2020). Can be accessed by Standish premium members in the premium report section of the Standish website.

ABOUT THE STANDISH GROUP

The Standish Group is a primary research advisory organization that focuses on software development performance. Using our extensive primary research, you can improve your investments in software. We are a group of highly dedicated professionals with years of practical experience helping organizations improve.

ABOUT THE CHAOS RESEARCH

The Standish Group collects, adjudicates, and approves between 2,500 and 5,000 new project cases per year. Each organization profile has 24 data points, and each project profile has more than 80 data points. In addition, there are 12 worksheets for each project. The CHAOS database is used to create reports such as the CHAOS 2020 Report as well as general queries, single-project assessments, future portfolio predictions, and performance benchmarks.

ABOUT THE AUTHORS

Jim Johnson is the founder and chairman of The Standish Group. He has a combination of technical, marketing, and research achievements focused on mission-critical applications and technology. He is best known for his research on project performance and early recognizing technology trends. Jim is a pioneer of modern research techniques and continues to advance in the research industry through case-based analytical technology.

Hans Mulder Prof.dr.ing MSc BA is Standish European research director and professor at the Antwerp Management School. As the founder of his own company, Venture Informatisering Adviesgroep, he is on the management and executive boards of various IT companies. He is regularly engaged as an IT expert when conflicts between companies need to be resolved in or out of court.

Copyright 2020 by The Standish Group International, Incorporated

All rights reserved. No part of this book may be reproduced in any form or by any electronic, mechanical, or manual means, including information storage and retrieval systems, without explicit permission in writing from the publisher. In no circumstances can this work be retransmitted in any form, repackaged in any way, or resold through any media.

PLEASE RESPECT INTELLECTUAL RIGHTS! Published by: The Standish Group International, Inc.
66 Tern Lane, Centerville, MA 02632 508-760-3600 www.standishgroup.com