

Decision Latency Theory:

It Is All About the Interval



Jim Johnson
Dreamer
The Standish Group

Copyright 2018 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.
60 State Street, Suite 700 Boston, MA 02109 508-760-3600 www.standishgroup.com

ISBN: 978-0-692-04830-6 Printed in the United States of America

Acknowledgment

Thanks to Michael Johnson for the fine portrait of the ship's mate and the other art work. Thanks to Jim Crear and Susan Bencuya, whose content contribution, advice, and patience in editing is commendable and highly valued. Thanks to Liz Bryan Muir for the cover and Suzanne Peterman for the inside book design. Thanks to all the Standish Group members who helped along the way. Thanks to the thousands of SURF members and CHAOS University attendees whose contributions made this work possible.

Standing six feet from the center, Tom Brady calls out, **“Rex Ryan hut!”**

The center snaps the ball to Tom Brady. As the defense charges forward, Brady looks around and sees Julian Edelman, 15 yards from scrimmage and crossing the middle of the field. Brady throws a strike right into Edelman's outstretched arms. The time from the snap to the throw was 1.9 seconds.

Edelman runs another 10 yards before he is brought down by the opposing safety. The interval from the catch to the tackle is 1.1 seconds—a total of 3 seconds for the entire play. Half a second after Brady passed the ball, Rob Gronkowski was open 40 yards downfield and Brady was wrapped in the arms of a 240-pound linebacker.

If Brady waited just a half a second longer, several things could have happened, and they often do: the pass could have gone incomplete or it could have been intercepted; Brady could have fumbled the ball, or been sacked, for a loss. On the other hand, an offensive lineman could have blocked the linebacker to give Brady that extra half second to throw to Rob Gronkowski to gain another 25 yards. The value is in the interval that Brady has from the snap to the throw. Brady, one of greatest quarterbacks of all time, has perfected the art of making timely decisions. The Standish Group has said for many years that time is the enemy of all projects—but the time is not the days or months of project work. The time is in the decision intervals.



any front or any electronic, mechanical, or
any form, repackaged in any way.

Standish Group International, Inc.
www.standishgroup.com

and the other art work. Thanks to Jim
in writing is commendable and
for the inside book design.
Thanks to the thousands of SURF
work possible.

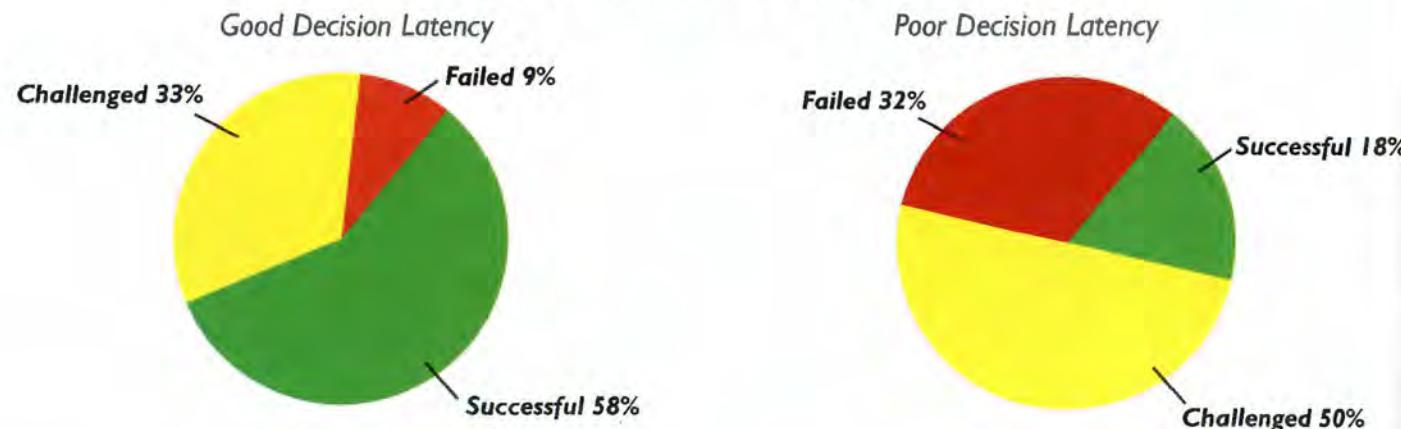
DECISION LATENCY THEORY

The Standish Group has determined that the root cause of software project failures and challenges is slow decision latency. Decision latency theory states: "The value of the interval is greater than the quality of the decision." Therefore, to improve project performance, organizations need to consider ways to speed-up their decisions.

The software industry has spent billions of dollars trying to find replicated ways to predict successful software developments and implementations. Most of the solutions center around process and tools. However, they do not address the underlying root cause of decision latency. In fact, many of these solutions make the problem of slow decision latency worse, while others, such as Scrum, can alleviate the underlying obstacle (if implemented correctly) by speeding up decision latency.

You can increase your project performance by simply recognizing this underlying disorder. You do not need any new tools. You do not need to overhaul your current process. What you need to do is eliminate things that have no value, but add time and cause delays. If your process is delaying decisions—get rid of it. If you are using a feature in a tool that causes decision delays, stop using that feature. If your meetings are causing decision delays, end those meetings. Make your meetings quicker, or use e-mail. Simply reducing decision latency can improve your project performance by 25%. Therefore, if your normal rate of success is 36%, directly addressing decision latency can increase that to 45%.

Effect of Decision Latency Skills



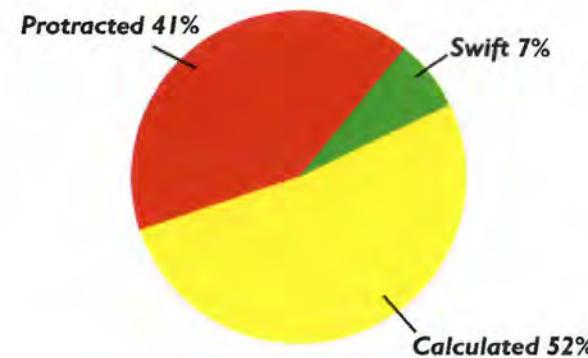
Compares projects with highly skilled decision latency teams against projects with poor decision latency teams. These projects are all drawn from our CHAOS database, and all of them took place from 2013 to 2017. Note that this assessment (and those that follow) uses "modern" (rather than traditional) assessment of all projects dating from FY2013 to 2017 within the CHAOS database. The "modern" assessment defines success as on time, on budget, and with satisfied customers. (The traditional assessment had been "on time, on budget, and on target.") However, those three metrics really involve project management success rather than project success, because they do not indicate customer satisfaction, or a return of value.) "Challenged" is defined as late or over budget, with less-than-satisfied customers. A failed project is defined as canceled before it is resolved, or resolved and not used.

DECISION LATENCY THEORY CONTINUED

The Standish Group has been leading the research on decision latency for over a decade. We discovered this issue in a work group. From there we issued a number of surveys through our DARTS service. We then ran a number of workshops focused on decision latency to discover two major items: the number of decisions to be made during a project; and the time required to make a decision. We then created a product, the Dezider.com, to be able to prove our numbers and average time to make a decision. We settled on this rule of thumb: a project will create one decision for every \$1,000 in project labor cost. In other words, a project that costs \$1 million in labor will require 1,000 decisions; a project that costs \$10 million in labor will generate 10,000 decision points.

The real key is the interval, and the cost of the interval. We interviewed one self-directed team of six people, who estimated that it takes them 10 minutes to make a firm decision—one hour, in terms of labor. In contrast, many of our work groups said they run meetings involving 20 people or more that go on for hours before a single decision is made. Even more disturbing: 40% of these decisions will be reversed by the next meeting. The general range of these intervals runs from less than 1 hour to more than 5 hours. A highly skilled work group can reduce their decision interval to 1 hour or less. A poorly skilled team will endure an interval of 5 hours or more.

Decision Speed



Question asked of 300 IT executives: "How would you rate your organization's speed in making critical decisions during the development and implementation of projects?" Source: SURF Database, 2017.

the project failures and challenges is slow. This is a greater than the quality of the project and how to consider ways to speed-up

associated ways to predict successful software development and tools. However, most of these solutions make the problem worse because they do not address the underlying obstacle

is a meeting disorder. You do not need to do anything but do is eliminate things causing disturbance—get rid of it. If you have meetings, if your meetings are causing problems simply reducing decision

and survival rate of success is 36%,



These projects are all drawn from the same source (follow) uses "modern" assessment methods. The "modern" assessments are based on time, on budget, and on scope, because they do not indicate the satisfaction of customers. A failed

DECISION LATENCY THEORY CONTINUED

Let us consider decision latency's effect on a project. The first line of the Table 1 is the estimated direct labor cost of a \$800,000 dollars and \$200,000 in decision cost for total of million dollars. (Over the years, our research has shown that a project this size has the best chance of success.) The second row shows our estimate of 1,000 decisions to be made during the project's life cycle. The third line is the decision latency skills of the organization, from highly skilled to poorly skilled. The fourth line describes the interval per hour by skill level. For example, a moderately skilled organization will have a decision latency interval of 3 hours. The fifth line is the total number of interval hours in decision-making for this project. This is calculated by interval per hour skill level (row 4) by the number of decisions (row 2). Example, 1,000 decisions by a moderately skilled team equals 3,000 hours of decision interval for this project.

Line 6 is the cost of decisions. Here, we settled on an average cost of \$200 per decision. We came to our estimate based on normalized labor costs and number of people involved in the average decision. (We think this estimate is very conservative.) We then multiplied the number of decisions by \$200 to get the cost of decisions. (Example: 1,000 decisions by \$200 equals \$200,000; 5,000 decisions equals \$1 million.) On line 7, we calculate the new cost of a project by adding the cost of decisions.

We then queried the CHAOS database by the new size and skill level to view the chances of success based on modern resolution.

Table 1: Cost of Decision Latency

| | | | | | |
|----|--------------------|-------------|-------------|-------------|-------------|
| 1 | Cost of project | \$800,000 | | | |
| 2 | # of decisions | 1,000 | | | |
| 3 | Latency skills | Highly | Skilled | Moderately | Poorly |
| 4 | Interval/hour | 1 | 2 | 3 | 5 |
| 5 | # of hours | 1,000 | 2,000 | 3,000 | 5,000 |
| 6 | Cost of decisions | \$200,000 | \$400,000 | \$600,000 | \$1,000,000 |
| 7 | Total project cost | \$1,000,000 | \$1,200,000 | \$1,400,000 | \$1,800,000 |
| 8 | Successful | 75% | 55% | 35% | 15% |
| 9 | Challenged | 23% | 39% | 51% | 62% |
| 10 | Failed | 2% | 6% | 14% | 23% |

DECISION LATENCY THEORY CONTINUED

The first thing you need to do to improve your decision latency is to be aware that it is a problem. You might start with some casual conversations about how long it takes to make decisions within your organization. In your next meeting, consider the cost of a decision by multiplying the average salary by the number of people in the meeting and time to make decisions. Then consider the value of the decision made against the cost to make the decision. Once you have done that and find you need to improve your skills at reducing decision latency, try benchmarking your decisions and set a goal for improvement. You can use The Standish Group Dezider.com to benchmark your decisions.

You might try to distribute decision-making power through the segmentation of decisions based on roles and responsibilities. Segmentation of decisions should be based on the impact of the decisions, from low to high.

A decision pipeline is a method to make decisions quickly and easily by involving the right people at the right time. This also engages more people in the decision process. The Standish Group Dezider.com is an example of a decision pipeline: a single-purpose application to help you make and be accountable for decisions.

Table 2: Resolution by Decision Latency

| Skill Level | Successful | Challenged | Failed |
|--------------------|------------|------------|--------|
| Highly Skilled | 58% | 33% | 9% |
| Skilled | 26% | 63% | 11% |
| Moderately Skilled | 20% | 51% | 29% |
| Poorly Skilled | 18% | 50% | 32% |

Table 2 shows the resolution of all software projects by size from FY2013 to 2017 within the CHAOS database.

THE WINNING HAND

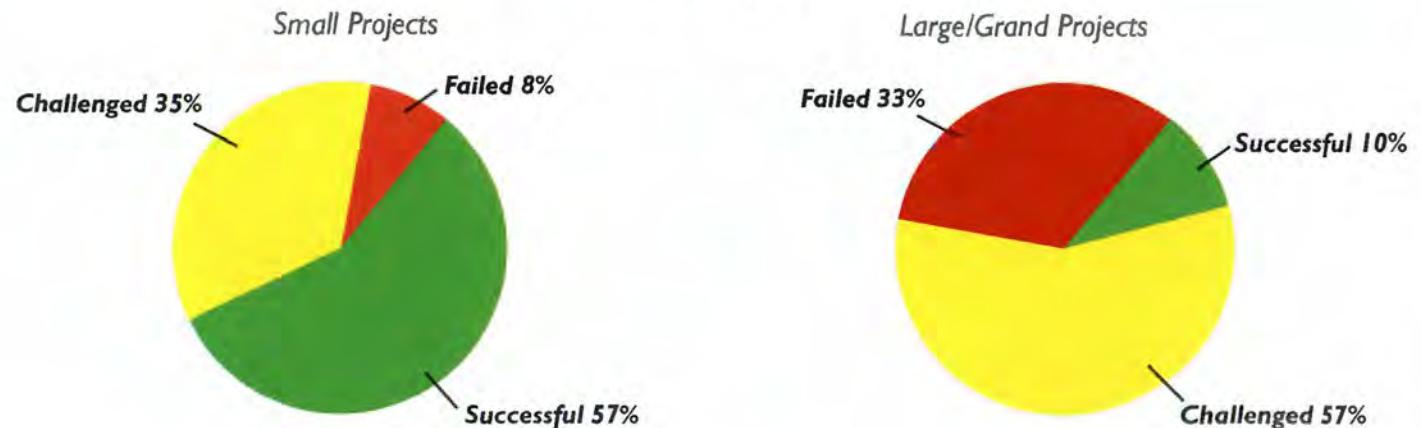
In our CHAOS Report 2016, we introduced a “winning hand” for project success. This hand comprises five cards.

- **First card:** The project needs to be small. This means six team members (maximum) with a time box of six months or less.
- **Second card:** The product owner or sponsor must be highly skilled.
- **Third card:** The process must be agile (using Scrum methodology, for example).
- **Fourth card:** The agile team must be highly skilled in both the agile process and the technology.
- **Fifth card:** The organization must be highly skilled at emotional maturity.

If we reduce decision delays, we can improve performance even further.

Project size has always been a major element in the CHAOS research. It was clear from the very beginning of this research that size is one of the most important factors in project outcome. On this page, we compare the resolution of small projects against large and grand projects.

Effect of Project Size



Compares small projects against large/grand projects, using the “modern” gauge of success (on time, on budget, with satisfied customers, for all projects from 2013 to 2017 within the CHAOS database. By “small,” we mean generally under \$1 million, or less than 10,000 hours, of productive labor. “Large” is \$6-10 million and 60,000 to 100,000 hours of productive labor. “Grand” is more than \$10 million or over 100,000 hours. These two pie charts clearly illustrate the impact of size on the results for “on time” and “on budget.” It is also clear that the larger the project, the less valuable the return rate. In many cases, larger projects never return value to an organization. The faster the projects go into production—like a Tom Brady pass—the quicker the payback starts to accumulate.

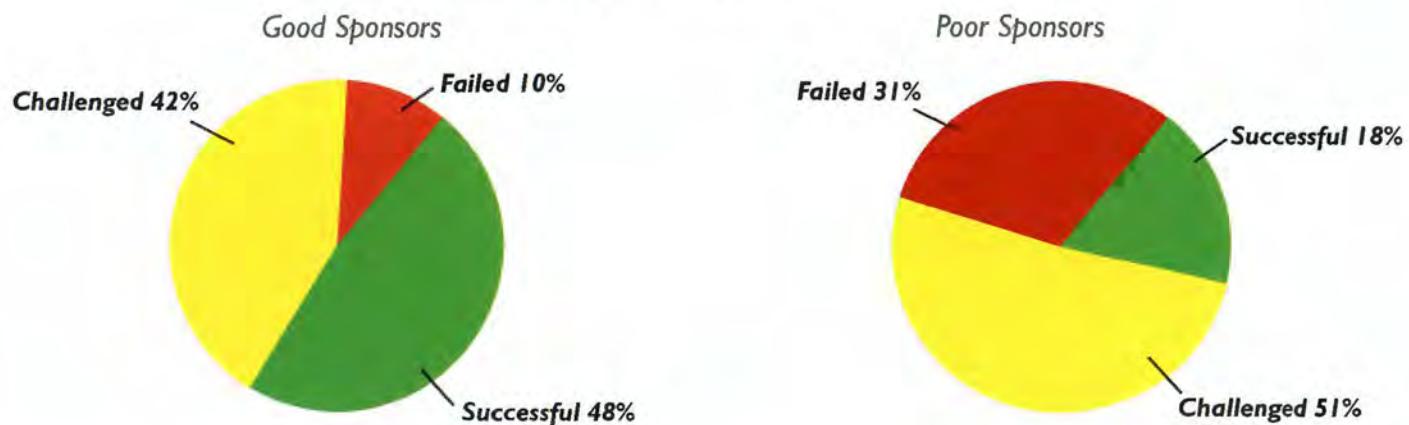
THE WINNING HAND: THE GOOD SPONSOR

The Standish Group has for many years held that the quality of the executive sponsor is paramount in the success of a project. It is the second card in our winning hand and, for a long time, was our number-one factor for success. (We have since replaced it with “reduced decision latency”—while noting that a good executive sponsor actually helps reduce decision latency, and thus the chance of project failure.)

One very critical red flag is when a committee acts as project sponsor, thus leaving the project without a single responsible owner. Committees themselves increase decision latency and cause many decisions to be reversed. A single, educated project sponsor has been shown to improve the likelihood of success by 15% to 20%. Many private and public software organizations have made having a single responsible owner a requirement for the approval of a project.

There is some consensus around the skills and temperament needed to be a good project sponsor. These skills are outlined in my book *The Good Sponsor*, and can be measured through the assessment that is featured in that book. Among other things, a good sponsor must be able to inspire people, work hard, dream, and make quick decisions. In addition, a good executive sponsor helps the team make decisions and creates an atmosphere in which rapid decisions can be made.

Effect of Good Sponsor



Compares projects with highly skilled project sponsors against projects with poorly skilled sponsors. “Success” is defined as “on time, on budget, good customer satisfaction.” These estimates cover all projects from 2013 to 2017 within the CHAOS database.

THE WINNING HAND: AN AGILE PROCESS

The third card in our winning hand is an agile process, such as the Scrum methodology. An agile process is all about teams. Many of these teams are set up to be self-managed or self-directed—which keeps the decision process within the team itself. Therefore, it is important that the team be moderately to highly skilled. We have seen many examples where a small team of six people outperforms large departments of software developers. We believe this is because they can make timely decisions. We also have seen that when decisions need to be made outside of the team, the process stalls. Therefore, decision latency must be considered inside and outside of the team if the agile process is to be effective.

A Scrum team is small number of people with complementary skills who are committed to a common purpose, performance goals, and approach for which they hold themselves mutually accountable. Scrum is the most popular agile method, with good reason. First, the product owner sets the goals for the sprint; second, there are daily stand-up meetings; third, there is an end-of-the-sprint retrospective; and fourth, there is a customer delivery process that requests feedback. This is all done in a time box of one to four weeks.

Our research has also shown that the longer teams work together, the better and more productive they become as a result of their ability to make faster decisions. As with Brady and Edelman, decisions begin to be made almost automatically.

Effect of the Agile Process with Skilled Teams



Compares projects using an agile process against projects that are not using an agile process. "Success" is defined as "on time, on budget, good customer satisfaction." These estimates cover all projects from 2013 to 2017 within the CHAOS database.

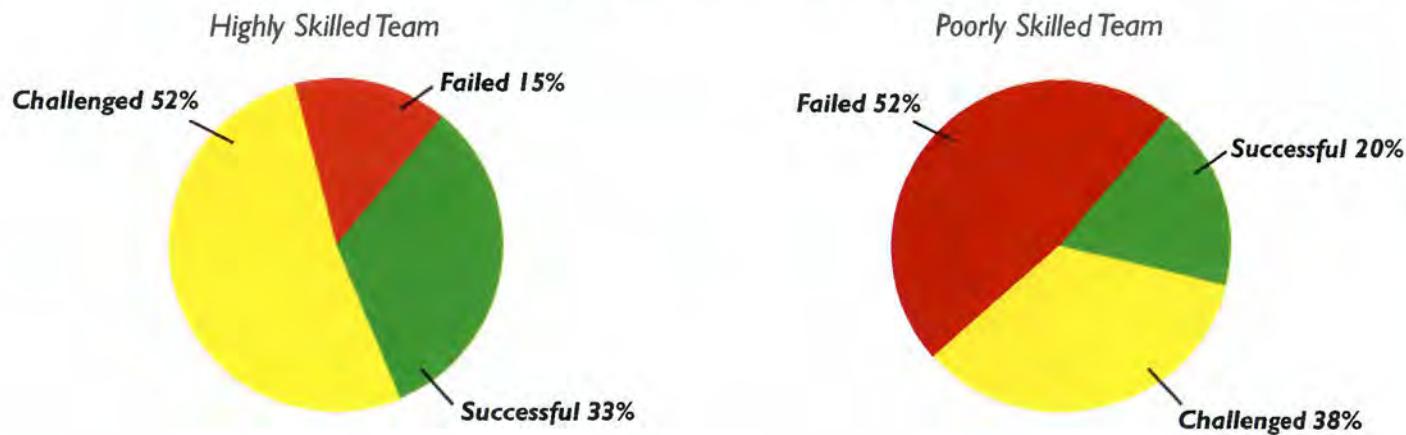
THE WINNING HAND: TEAM SKILLS

Antonio Garcia Martinez in his book, *Chaos Monkeys: Obscene Fortune and Random Failure in Silicon Valley*, stated, "With the right team, no man or organization can stand against you, and you will ultimately triumph. With the wrong team, you will produce internal problems even faster than the external world can, and your eventual death will effectively be a suicide." A key factor for a successful project is having the skills of the team match the project. Any team will struggle if its skills do not conform to the tasks at hand. In addition, the team will need to feature a variety of skills, since the project will require many different tasks to be accomplished.

For example, if you are building a database and the database you are using is Sybase, then a person familiar with Oracle will take longer to do these tasks and be less productive. It does not mean the person cannot learn the differences—it just will take a little longer. (A new receiver for Tom Brady would not know the plays, and the two would have to practice together.) These are the things you must consider in order to set your expectations for the project's timeline and productivity.

On the other hand, if you are also building user interfaces, then you need to have people that know how to build user interfaces. This doesn't mean everyone on the team needs to be gifted in that regard—but it would help.

Effect of Using the Skilled Teams



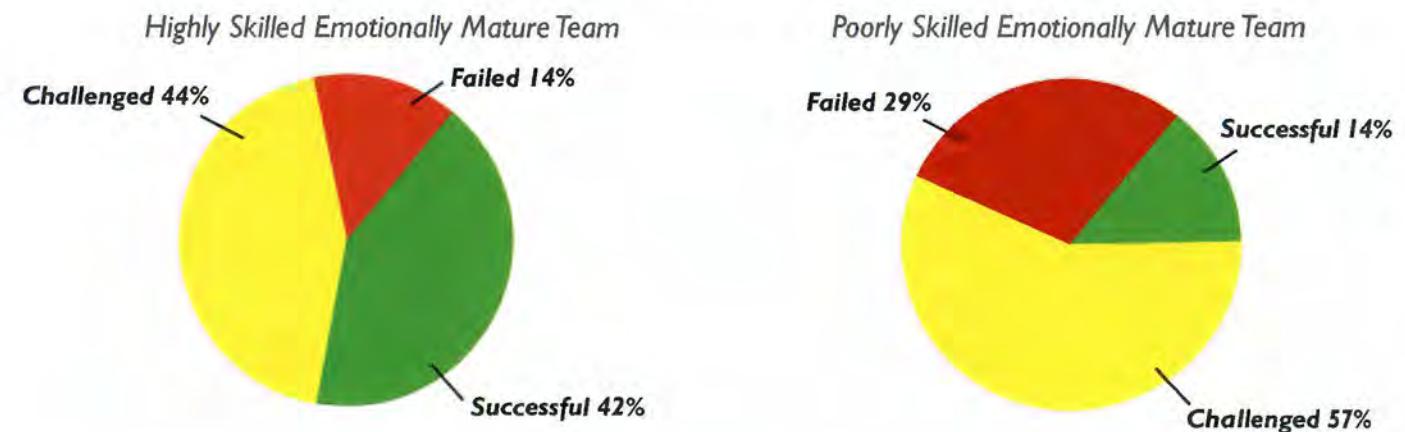
Compares projects using highly skilled team members against projects that are using poorly skilled members. "Success" is defined as "on time, on budget, good customer satisfaction." These estimates cover all projects from 2013 to 2017 within the CHAOS database.

THE WINNING HAND: EMOTIONAL MATURITY

The last card in the winning hand is the team must be emotionally mature. A team with emotional maturity has been shown to work together well and be highly productive. In 1994, we found that most project-failure issues revolved around human behavior; or what are generally called "soft skills." Soft skills include managing expectations and building consensus. The Standish Group has been doing research on these skills (under the category of "team emotional maturity") since 2003. In 2006, we published *The Public Execution of Miss Scarlet*, which outlined "five deadly sins of project management". Dealing with these five sins are part of the emotional maturity skill set. Later we published several research reports on those five deadly sins. In 2013, we published the Emotional Maturity Research Report and created an Emotional Maturity online assessment.

Currently we are working on a book titled *The Good Mate*, which provides self-help for team members who wish to improve their skills around emotional maturity. It's a combination of our research with research by some well-known marriage counselors. *The Good Mate* is due out in 2018. In it, we have identified 50 emotional maturity skills that are centered around 10 principles. The principles include being influential, practicing mindfulness, dealing with and overcoming the five deadly sins, and being a good problem-solver. Note that these are not stand-alone skills; rather, they have a combinatory effect. In order to be really skilled at emotional maturity, you need to improve those 50 skills and 10 principles.

Effects of Emotionally Mature Teams

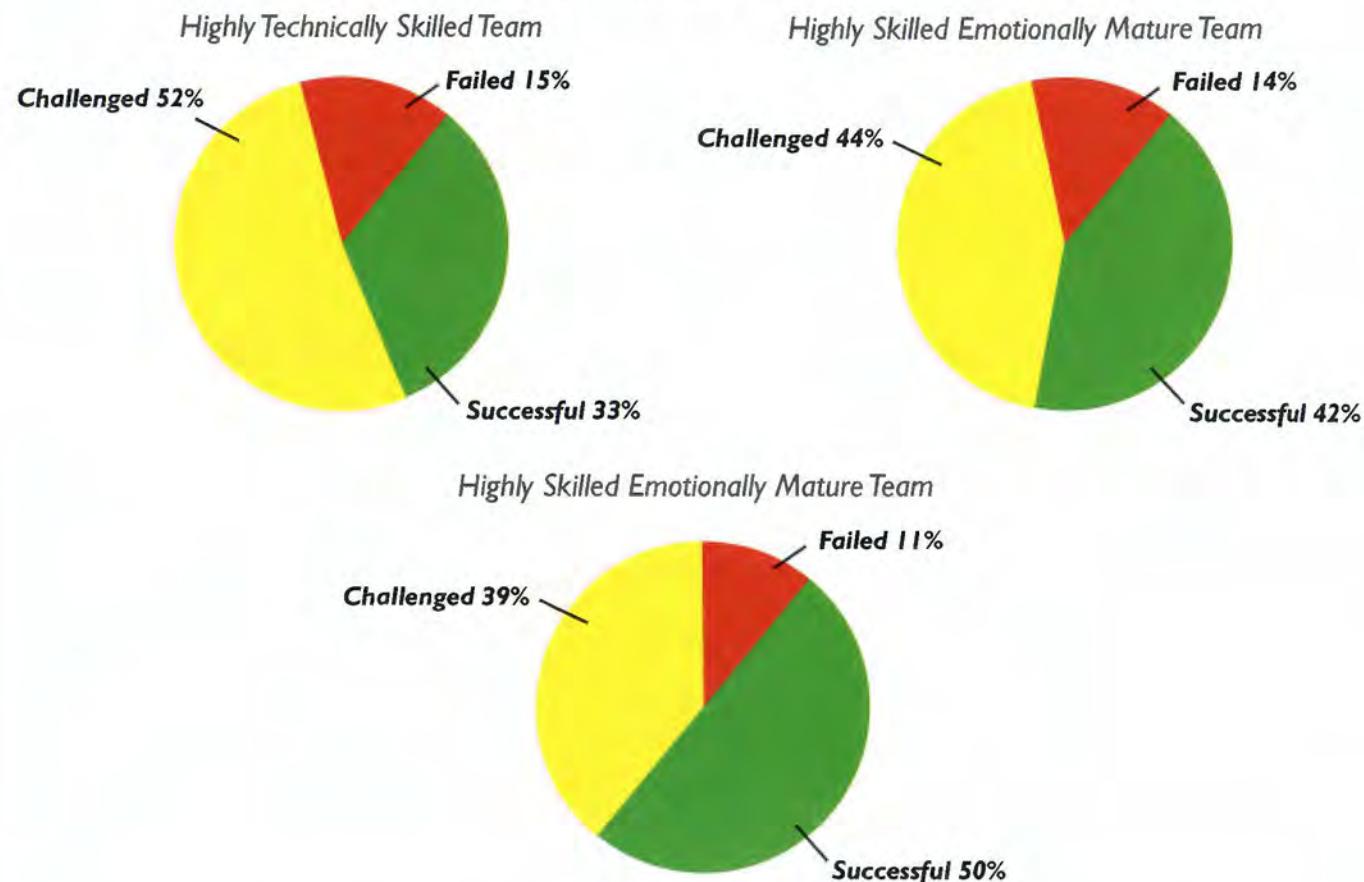


Compares projects using highly skilled emotionally mature team members against projects that involved teams that are poorly skilled in this regard. "Success" is defined as "on time, on budget, good customer satisfaction." These estimates cover all projects from 2013 to 2017 within the CHAOS database.

THE WINNING HAND: A COMBINATION

A single ace does not make a winning hand. A winning hand starts with two of a kind. The combinatory effect of each of the cards improves your chances of winning. For example, let's consider the last two cards, technical skills and emotional maturity skills. The combination of these two cards improves the success rate—from 33% for technical skills and 42% on emotional maturity to 50% when combined. Failure rates go down to 11%. The more aces you have, the better the likelihood that you will have a successful project that returns value.

Combination Effect of Technical and Emotional Skills



Compares projects using a highly technically skilled team against those using a highly skilled emotionally mature team and against projects using both. "Success" is defined as "on time, on budget, good customer satisfaction." These estimates cover all projects from 2013 to 2017 within the CHAOS database.

THE WINNING HAND: DECISION LATENCY

Let us assume you are highly skilled at decision latency. If you play your five cards, you have an 86% chance that a project will come in on time and on budget, with satisfied customers. You have only a 1% chance the project will fail, and only a 13% chance that it will be challenged in some way. More importantly, the project will have a 78% chance of returning very high to high value and only a 7% chance of returning no to low value. If you do not do these five things well, however, the chances of a failed, challenged, or low-value result increase. Even if you do a bunch of other things (outside of these five things in conjunction with swift decision making), you will most likely find that you've wasted your time and money and have made very little impact.

A losing hand coupled with poor decision latency predicts several things. Imagine a large project with hundreds of disbursed team members. A non-agile methodology with a mature process. A team that is moderately to poorly skilled in terms of process and technology. A product owner or sponsor that is also moderately to poorly skilled. An organization with moderate to poor emotional maturity skills. Add moderate to poor decision latency. This will give you only a 1% chance that your project will come in on time and on budget, with satisfied customers. You have a 72% chance the project will fail and a 27% chance that it will be challenged in one way or another. More importantly, your project will have a 77% chance of returning very low to no value and less than a 1% chance of returning high value.

Table 3: Winning Hand Versus Losing Hand Coupled with Decision Latency.

| | Successful | Challenged | Failed | High Value | Low Value |
|---------------------|------------|------------|--------|------------|-----------|
| Winning Hand | 86% | 13% | 1% | 78% | 7% |
| Losing Hand | 1% | 27% | 72% | 1% | 77% |

Table 3 shows the results of a winning hand versus a losing hand. A winning hand is defined as a small, agile project with a skilled team and sponsor and an emotionally mature environment. A losing hand is defined as a large, non-agile project with an unskilled team and sponsor, an emotionally immature environment, and poor decision latency. The results are a segment of the 25,000 projects in the CHAOS database from Fiscal Year 2013 to 2017. Success is defined as on time, on budget, and with satisfied customers. "Challenged" is defined as late or over budget, with less-than-satisfied customers. A "failed" project is canceled before it is resolved, or resolved and not used.

CLASSIC CHAOS: MEASURING SUCCESS

In the original CHAOS Report, we set out to examine three objectives: 1) the scope of software project failures; 2) the major factors that cause software projects to fail; and 3) the key ingredients that can reduce project failures. A couple years after the original report, we were challenged to find a root cause: this became objective number four. The last 20 years have seen some changes in the scope of project failures. In certain areas, there has been much improvement, while in other areas things have gotten worse. The result is that the statistical numbers look almost the same as they did 20 years ago! For example, in 1994, 31% of all projects were on time, on budget, and on target. In 2017, the number of projects that were on time, on budget, and on target had improved to 36%.

Each project in the CHAOS database features six different measurements of success: on time; on budget; on target; on goal; customer satisfaction; and return of value. Note that the traditional metrics of success are "on time, on budget, and on target." However, these three metrics really involve project management success rather than project success, because they do not indicate customer satisfaction, or a return of value. Basically, they measure how good project management can help with estimation of time, money, and tasks, as well as how good project management can control a project's time and money constraints to meet these goals.

Table 4: Resolution by Traditional Measurement

| Year/Resolution | 2013 | 2014 | 2015 | 2016 | 2017 |
|-------------------|------|------|------|------|------|
| Successful | 41% | 36% | 36% | 36% | 36% |
| Challenged | 40% | 47% | 45% | 47% | 45% |
| Failed | 19% | 17% | 19% | 17% | 19% |

Table 4 show the traditional measures of all software projects from FY2013–2017 within the CHAOS database. Traditional metrics study how closely a project remains on time, on budget, and on target. All data, unless otherwise noted, represents results from FY2013 to 2017. The total number of software projects is 25,000-plus, with an average of 5,000 per yearly period.

CLASSIC CHAOS: CUSTOMER SATISFACTION

In the original CHAOS Report, we set out to examine three objectives: 1) the scope of software project failures; 2) the major factors that cause software projects to fail; and 3) the key ingredients that can reduce project failures. A couple years after the original report, we were challenged to find a root cause: this became objective number four. The last 20 years have seen some changes in the scope of project failures. In certain areas, there has been much improvement, while in other areas things have gotten worse. The result is that the statistical numbers look almost the same as they did 20 years ago! For example, in 1994, 31% of all projects were on time, on budget, and on target. In 2017, the number of projects that were on time, on budget, and on target had improved to 36%.

Each project in the CHAOS database features six different measurements of success: on time; on budget; on target; on goal; customer satisfaction; and return of value. Note that the traditional metrics of success are “on time, on budget, and on target.” However, these three metrics really involve project management success rather than project success, because they do not indicate customer satisfaction, or a return of value. Basically, they measure how good project management can help with estimation of time, money, and tasks, as well as how good project management can control a project’s time and money constraints to meet these goals.

Table 5: Resolution by Modern Measurement

| Year/Resolution | 2013 | 2014 | 2015 | 2016 | 2017 |
|-------------------|------|------|------|------|------|
| Successful | 31% | 28% | 29% | 29% | 33% |
| Challenged | 50% | 55% | 52% | 54% | 48% |
| Failed | 19% | 17% | 19% | 17% | 19% |

Table 5 shows “Modern” (as opposed to “traditional”) resolution of all software projects, 2013–2017, within the CHAOS database. “Modern” resolution measures are “on time” and “on budget,” with a satisfactory result. Note: Satisfaction measures are “very satisfied,” “satisfied,” and “somewhat satisfied.”

CLASSIC CHAOS: “PURE SUCCESS”

We have created a new definition of success that we call “pure success.” Pure success is the combination of high customer satisfaction with high return on value to the organization. The bar is high for this definition of success, since it only considers projects that have a high or very high return on value and high or very high customer satisfaction. The definition covers only pure project success and not, for example, success of project management. Nor is there consideration of on time, on budget, or the scope of the deliverable. We do not count “somewhat satisfied” as with the modern definition, or count “average return on value.” Consequently, “pure success” occurs at half the rate measured by traditional and modern metrics.

We found some interesting facts as we dug down into the “pure” numbers. For example, organizations that were highly skilled at decision latency enjoyed four times the project success rate of organizations with poor decision latency skills. Moreover, projects that featured a highly skilled executive sponsor had better than three times the success rate of projects without one. Small projects also enjoyed three times the success rate of larger projects as did organizations that were skilled at scope optimization.

Agile projects enjoyed twice the success rate of non-agile projects when using the “pure” metric. The chances of gaining “pure” success are almost 70% if the projects are executed by the attributes of the Winning Hand and the organization is highly skilled at decision latency.

Table 6: Resolution by Pure Measurement

| Year/Resolution | 2013 | 2014 | 2015 | 2016 | 2017 |
|-------------------|------|------|------|------|------|
| Successful | 16% | 15% | 17% | 17% | 14% |
| Challenged | 65% | 68% | 64% | 66% | 67% |
| Failed | 19% | 17% | 19% | 17% | 19% |

Table 6 shows “Pure” resolution of all software projects, 2013–2017, within the CHAOS database. Pure resolution measures only “high value” and “customer satisfaction.”

CLASSIC CHAOS: STRATEGIC GOAL

Another metric that could be considered is "strategic goal." We added this metric by customer request; many of our executive clients were in favor of it.

Here, project success is defined by the project's position in relation to the corporate goal. Table 7 shows that 11% of projects map precisely to the goal, and another 15% come close. Using these two numbers would make 26% of projects successful in terms of this metric. "Loose" to "distant" amounted to 56% of the projects, and 18% of the projects could be considered as failed.

The strategic-goal metric is not used in combination with other metrics. It stands alone, and the reason it stands alone is because it is either orthogonal or incompatible with the other five metrics. We tried to pair strategic goal with value and found a very interesting fact: only 20% of "precise" projects returned high value, while 34% of "distant" projects returned high value. At the same time, almost half of projects that received a "vague" measure returned high value. In other words, if we combined "strategic goal" with "value" as a measurement, the results would be strange.

We do have to consider what it means if the return of value scores higher as you move away from the strategic goal. Our opinion is that vague projects are more innovative, and are often disruptive with higher returns.

Table 7: Resolution by Strategic Goal Versus Value Measurement

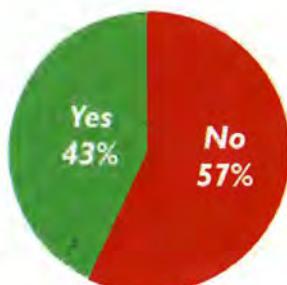
| Goal | % | Very High Value | High Value | Average Value | Low Value | Very Low Value |
|---------|-----|-----------------|------------|---------------|-----------|----------------|
| Precise | 11% | 7% | 13% | 53% | 21% | 6% |
| Close | 15% | 8% | 16% | 52% | 19% | 5% |
| Loose | 21% | 12% | 22% | 47% | 15% | 4% |
| Vague | 18% | 17% | 32% | 39% | 8% | 4% |
| Distant | 17% | 15% | 29% | 21% | 19% | 16% |
| Failed | 18% | | | | | |

Table 7 shows percent of projects in relationship to the strategic goal as measure of success (column 2). Columns 3 to 5 shows the return on value to the strategic goal.

CLASSIC CHAOS: ATTRIBUTES OF SUCCESS

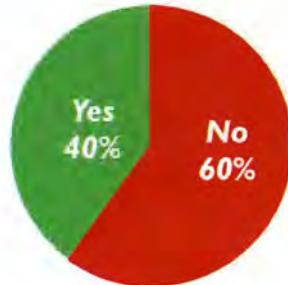
The CHAOS database includes six individual attributes of success: on budget, on time, on target, value, on goal, and satisfaction. Let's look at these attributes individually.

On Budget



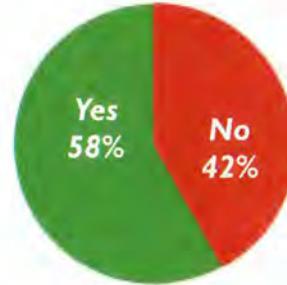
Percentage of projects that were on budget from 2013 to 2017 within the CHAOS database.

On Time



Percentage of projects that were on time from 2013 to 2017 within the CHAOS database.

On Target



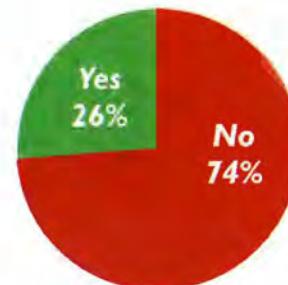
Percentage of projects that were on target from 2013 to 2017 within the CHAOS database.

Value



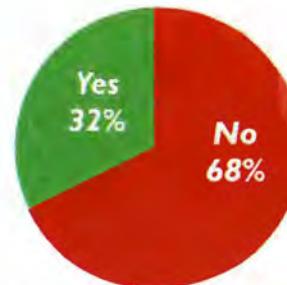
The percentage of projects considered valuable from 2013 to 2017 within the CHAOS database.
(Very valuable & valuable).

On Goal



Percentage of projects that were on goal (i.e., "precise" or "close") from 2013 to 2017 within the CHAOS database.

Satisfaction



Percentage of projects considered satisfactory (i.e., "very satisfied" or "satisfied") from 2013 to 2017 within the CHAOS database..

request,
Table 7 shows
that almost half (49%) of the
projects had the same number of

and the reason it
is often only used to pair
with a high
percentage of projects that
have been used with

the highest return
from the
highest value.

Project Management

| Low Value | Very Low Value |
|-----------|----------------|
| 0% | 6% |
| 10% | 5% |
| 20% | 4% |
| 30% | 3% |
| 40% | 1% |

CLASSIC CHAOS: PROJECT SIZE

Let's face facts: when it comes to project success, size does matter. The trend of minimal viable product (MVP) demonstrates this fact. Over almost two and a half decades, small projects have outperformed their larger counterparts by a wide margin. Do not let people tell you that size is dictated by the scope of requirements. Scope is what you make it, and you can make it small. Remember, "small" is the first card in our Winning Hand. Small makes for short decision latency.

One of the major services of our Value Portfolio Optimization and Management Service is to break up large software projects into multiple small projects, which means early delivery for success, quicker return on value, and greater customer and user satisfaction. We have found that most software projects only require a small team for a short duration in order to deliver value to the organization; only in very rare cases do projects need to be larger and longer.

Table 8: Resolution by Project Size

| Resolution/Size | Successful | Challenged | Failed |
|-----------------|------------|------------|--------|
| Grand | 4% | 53% | 43% |
| Large | 12% | 59% | 29% |
| Medium | 18% | 59% | 23% |
| Moderate | 25% | 62% | 13% |
| Small | 57% | 35% | 8% |

Table 8 shows "Modern" resolution of all software projects by size, from FY2013 to 2017, within the CHAOS database. The project size is based on productive labor. "Small" implies less than 10,000 hours of productive labor; "moderate" 10,000 to 30,000 hours; "medium" 30,000 to 60,000 hours; "large" 60,000 to 100,000 hours, and "grand" equals more than 100,000 hours.

CLASSIC CHAOS: PROJECT SPONSOR

All projects, from the smallest to the largest, show improvement with a skilled project sponsor. The project sponsor is the single most important person involved with a project and is ultimately responsible for its success or failure. According to the CHAOS database, on average a highly skilled project sponsor will garner a success rate of almost 50% and a failure rate of only 10%; a poorly skilled project sponsor, on the other hand, is likely to see fewer than 20% of associated projects coming in on time and on budget with satisfied customers. The larger and more complex the project, the greater the difference between success and failure due to the skills of the project sponsor.

The problem is, most executives do not know how to sponsor software projects. Often they cannot relate to the team, and they feel somewhat uncomfortable. My book *The Good Sponsor* helps the project sponsor understand their role, responsibilities, and the basic skills needed, as well as ways in which to test and improve those skills. It also offers an online assessment and benchmark for sponsor skills. You can purchase *The Good Sponsor* as well as the online appraisal on The Standish Group website (www.standishgroup.com/store).

Table 9: Resolution by Sponsor Skills

| Skill Level | Successful | Challenged | Failed |
|--------------------|------------|------------|--------|
| Highly Skilled | 48% | 42% | 10% |
| Skilled | 32% | 57% | 11% |
| Moderately Skilled | 21% | 52% | 27% |
| Poorly Skilled | 18% | 51% | 31% |

Table 9 shows "Modern" resolution of all software projects from 2013 to 2017 within the CHAOS database, ranked by skill level of the executive sponsor.

CLASSIC CHAOS: AGILE PROCESS

The third ace in the Winning Hand, and a major contributor to improved decision latency, is the agile process. Twenty years ago, when we started researching project performance, there were hundreds of methodologies we needed to study. While there are still many hundreds, it is now possible to divide them into two categories: agile and non-agile. Agile makes up about a quarter of the projects in the CHAOS Database; non-agile covers the rest. Of the non-agile projects, "waterfall" types account for about 30%, but that number is dropping like a rock. Many of the non-agile projects actually claim to be agile, and some of them do have some agile properties. However, we need to ensure that agile projects are really agile, and we have several tests to do just that.

The results for all projects show that agile projects enjoy a 60% greater chance of success than non-agile projects. Looking deeper, we find that "waterfall" projects are three times more likely to fail than agile projects, and when we break "agile versus non-agile" by size, we find some really interesting data. Large agile projects succeed at twice the rate of non-agile projects, and fail half as often. "Medium-agile" projects do not fare that much better (31% versus only 19% for non-agile projects). Only in the small category does non-agile come close to agile. Therefore, we conclude that size trumps agile methodology—but in combination, they can be deadly.

Table 10: Resolution by Method

| Size | Method | Successful | Challenged | Failed |
|----------------------|-----------|------------|------------|--------|
| All-Size Projects | Agile | 42% | 50% | 8% |
| | Non-agile | 26% | 53% | 21% |
| Large-Size Projects | Agile | 18% | 66% | 16% |
| | Non-agile | 9% | 56% | 35% |
| Medium-Size Projects | Agile | 31% | 59% | 10% |
| | Non-agile | 19% | 61% | 20% |
| Small-Size Projects | Agile | 59% | 37% | 4% |
| | Non-Agile | 56% | 34% | 10% |

Table 10 shows "Modern" resolution of all software projects by method—agile versus non-agile—within the CHAOS database, 2013–2017. (Note: The agile numbers in Winning Hand Section also include teams skilled at the agile methods.)

CLASSIC CHAOS: TEAM TECHNICAL SKILLS

Our fourth factor of success is a technically skilled team. Having skilled resources for a project means having the right people doing the right things at the right time. We have looked at this in two ways, one using the normal four-point range of "highly skilled" to "poorly skilled," and the other by using a five-point range, from "gifted" to "unskilled." Our research indicates that having "gifted" people increases the chances of success by about 10% over "highly skilled" teams. However, very few projects are able to deploy a gifted person, never mind a gifted team. On the other hand, your chances of success are cut in half by an unskilled team.

Table 11: Resolution by Technical Skill Level

| Skill Level | Successful | Challenged | Failed |
|--------------------|------------|------------|--------|
| Highly Skilled | 33% | 52% | 15% |
| Skilled | 34% | 52% | 14% |
| Moderately Skilled | 24% | 56% | 20% |
| Poorly Skilled | 21% | 37% | 42% |

| Skill Level | Successful | Challenged | Failed |
|-------------|------------|------------|--------|
| Gifted | 43% | 42% | 15% |
| Talented | 34% | 51% | 15% |
| Competent | 29% | 53% | 18% |
| Able | 22% | 54% | 24% |
| Unskilled | 19% | 58% | 23% |

Table 11 shows "Modern" resolution of all software projects, 2013-2017, within the CHAOS database, by team skill level.

around design latency, is the agile approach. There were hundreds of projects. It is now possible to divide them into three categories of the projects in the CHAOS database. They account for about 30%, but most likely there to be agile, and some of the remaining projects are really agile, and we

have a much better chance of success than non-agile projects. Agile projects are more likely to fail than non-agile projects. This is very interesting data. Large projects are often "Medium-agile" projects. Only in the small category do they use an agile methodology—but in

| Challenged | Failed |
|------------|--------|
| 8% | 8% |
| 21% | 21% |
| 16% | 16% |
| 35% | 35% |
| 10% | 10% |
| 20% | 20% |
| 4% | 4% |
| 10% | 10% |

CHAOS database, 2013-

CLASSIC CHAOS: EMOTIONAL MATURITY

The last (but not least) ace card is emotional maturity. Emotional maturity is also the third factor of success. Emotional maturity is at the very center of project performance. Up until recently we considered emotional maturity to be an organizational skill that needed improvement, but we found this to be too hard a task. We then turned attention to the emotional maturity for the team. Here we found gold. If the team and members of team are emotional maturity, project success increases. Our number that just skilled maturity team will improve project success by about 25%. Many agile teams are self-direct or self-manage teams. Our research on maturity has spawned three books: *The Public Execution of Miss Scarlet* (2006), "Emotional Maturity Research Report" (2013), and *The Good Mate* (2018). Check out our website store at www.standishgroup.com/store.

A good self-directed team starts with good teammates. Good teammates start with individual members. The Good Mate is our attempt to provide some basic guidance about how to behave and interact within the team and other teammates as an individual in a self-directed team. *The Good Mate* provides 50 basic skills a teammate can use to improve as a teammate to make a self-directed team a success. By understanding and improving on these basic skills a single teammate can be the beacon to help his or her teammates. Teammates like this work their magic not by direction or mandate, but by being a good example. They do it by being a good mate.

Table 12: Resolution by Emotional Maturity

| Skill Level | Successful | Challenged | Failed |
|--------------------|------------|------------|--------|
| Highly Skilled | 42% | 44% | 14% |
| Skilled | 41% | 49% | 10% |
| Moderately Skilled | 18% | 59% | 23% |
| Poorly Skilled | 14% | 57% | 29% |

Table 12 shows "Modern" resolution of all software projects by team skill level, 2013–2017, CHAOS database.

CLASSIC CHAOS: PROJECTS BY INDUSTRY

Looking at project resolution by industry provides another view of the CHAOS database. Table 13 shows the resolution of all software projects by industry from FY2013 to 2017 within the CHAOS database. The results show that retail projects had the highest success rate, at 34%, using the "modern" definition of success. Telecom projects had the highest failure rates at 23%, and financial projects the highest "challenged" rate, at 55%. Our focus groups and general interaction with different industries show that the retail industry benefits from very low decision latency, while government and regulated industries (telecom, for instance) tend to suffer from long decision latency. This was a major clue as we looked for root causes of project failure. Teams within governments and telecom projects that set out to reduce decision latency have seen improved project performance.

Many of our benchmark clients like to compare their results to the results of other organizations in the same industry, and we do this as matter of course. However, we found that "industry" is not the most accurate or important metric of comparison. The most accurate is to consider industry as a minor filter, with project type, size, skills, and methodology as primary filters. Other minor filters would include organizational size and geography. In our Resolution Benchmark we use this technique, filtering by project type, size, skills, and methodology to benchmark project portfolios.

Table 13 Resolution by Industry

| Industry | Successful | Challenged | Failed |
|---------------|------------|------------|--------|
| Banking | 32% | 54% | 14% |
| Financial | 30% | 55% | 15% |
| Government | 26% | 52% | 22% |
| Healthcare | 31% | 52% | 17% |
| Manufacturing | 31% | 51% | 18% |
| Retail | 34% | 51% | 15% |
| Services | 31% | 50% | 19% |
| Telecom | 24% | 53% | 23% |
| Other | 29% | 51% | 20% |

Table 13 shows Resolution of all software projects by industry, 2013–2017, within the CHAOS database.

CLASSIC CHAOS: WORLD REGIONS

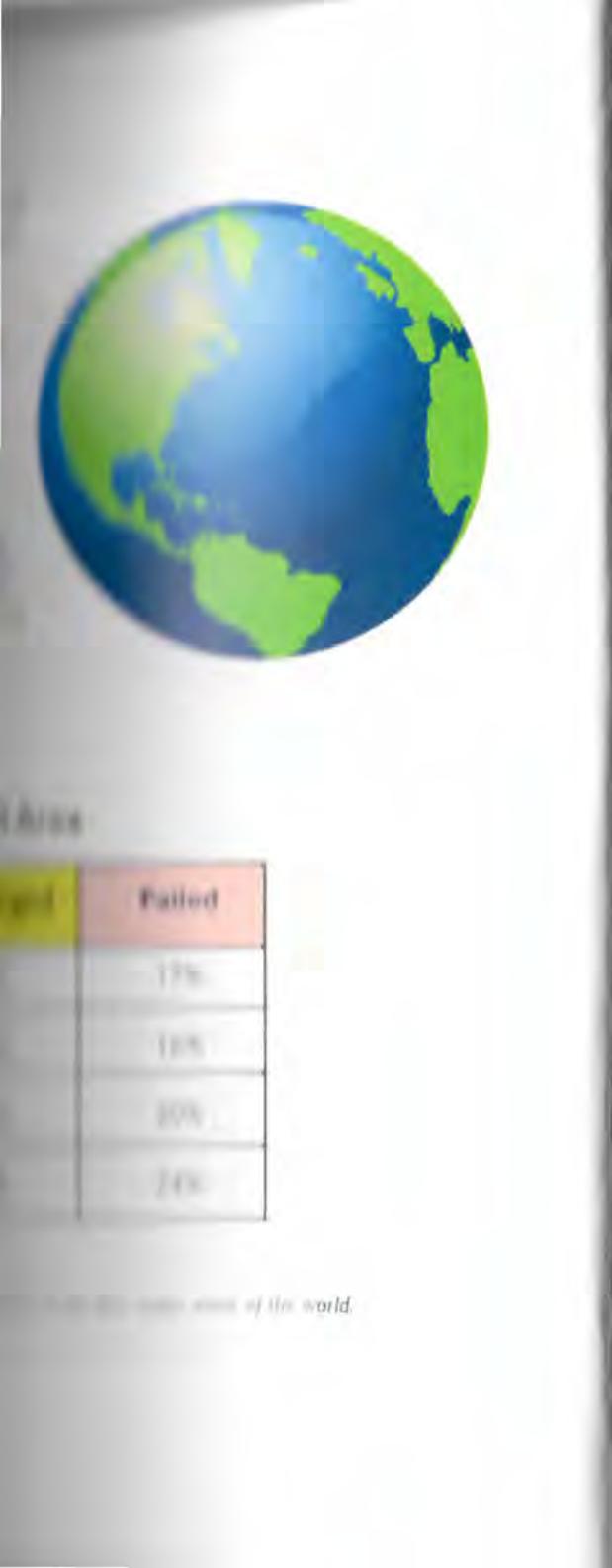
Our research shows that project resolution differs slightly by world regions. Using the “modern” definition of success (i.e., on time and on budget with a satisfactory result), we see that Europe and North America return much the same success rate, and the “rest of the world” has the highest fail rate, with 24% not making it. Interestingly, Asia exhibits the lowest success rate as well as the lowest emotional maturity scores, according to appraisals and benchmarks. We know from our research that Asian have a real and serious problem with individual decision-making and, probably as a consequence, suffer from long decision intervals. Individualism is more highly valued in Europe and North America and this would naturally decrease decision latency. Although other countries in the rest of the world also exhibit long decision latency, it is not as uniform a problem as in Asia.



Table 14: Resolution by World Area

| | Successful | Challenged | Failed |
|----------------------|------------|------------|--------|
| North America | 31% | 52% | 17% |
| Europe | 31% | 53% | 16% |
| Asia | 25% | 55% | 20% |
| Rest of World | 27% | 49% | 24% |

Table 14 shows Resolution by “modern” definition of all software projects, 2013–2017, in the four major areas of the world.



CLASSIC CHAOS: TYPE OF PROJECT

"Type of project" has a major effect on resolution. Table 15 shows the resolution of all software projects by project type, 2013–2017, within the CHAOS database, using the modern definition of success. Projects using a modernization-in-place technique had the highest success rate, at 57%. Projects that were developed from scratch using traditional languages and methods had the lowest success rate, at 23%.

The results also show that projects that were developed from scratch using traditional languages and methods had the highest "challenged" rate, at 61%. The lowest challenged rate of 28% went to projects of purchased application software with modifications. The highest failure rates, 23%, went to projects of purchased software with extensive modifications. Modernization projects had the lowest failure rate, at 8%.

Table 15: Resolution by Project Type

| Project Type | Successful | Challenged | Failed |
|--|------------|------------|--------|
| Developed from scratch using traditional languages and methods | 23% | 61% | 16% |
| Developed from scratch using modern methodologies | 26% | 52% | 22% |
| Developed some components & purchased others | 26% | 58% | 16% |
| Purchased components & assembled the application | 25% | 59% | 16% |
| Purchased application & extensively modified | 36% | 41% | 23% |
| Purchased application & modified | 55% | 28% | 17% |
| Purchased application & performed no modifications | 49% | 33% | 18% |
| Modernization | 57% | 35% | 8% |
| Other | 29% | 46% | 25% |

Table 15 shows "Modern" resolution of all software projects by project type, 2013–2017, within the CHAOS database.

FACTORS OF SUCCESS

Each year since 1995, when we wrote our "Unfinished Voyages" research paper, we create a list of 10 attributes that we call the Factors of Success. Table 16 is for 2018 and reflects our opinion of the importance of each attribute. Table 16 is also our recommendation of the amount of effort and investment

that should be considered to improve project success and value. It is our tradition to assign points to each factor to highlight its relevance. These points should also be considered as an investment guideline for organizational improvement. For example, if you are spending \$50 million on IT projects, then 2% of the money should be going toward improving the value of those projects. Based on this amount, our recommended breakdown of money to be allocated to each factor is calculated next to the point value on the chart. So, if you want your projects to be more successful, with higher value and greater customer satisfaction, you should carefully consider where you invest your project improvement money.

Table 16: Factors of Success

| Factors of Success | Points | Investment |
|---|------------|-------------|
| Decision Latency | 25 | 25% |
| Minimum Scope | 14 | 15% |
| Project Sponsors | 14 | 15% |
| Agile Process | 9 | 12% |
| Talented Staff | 9 | 12% |
| Team Maturity | 9 | 12% |
| User Involvement | 5 | 3% |
| SAME | 5 | 1% |
| Optimization | 5 | 1% |
| PM/Execution | 5 | 1% |
| Total Points & Yearly Investment | 100 | 100% |

search paper; we create a list of 18 and reflects our opinion of the amount of effort and investment that should be considered to improve project success and value. It is our tradition to assign points to each factor to highlight its relevance. These points should also be considered as an investment guideline for organizational improvement. For example, if you are spending \$50 million on IT projects, then 2% of the money should be going toward improving the value of those projects. Based on this amount, our recommended breakdown of money to be allocated to each factor is calculated next to the point value on the chart. So, if you want your projects to be more successful, with higher value and greater customer satisfaction, you should carefully consider where you invest your project improvement money.

FACTORS OF SUCCESS: DECISION LATENCY

The top six Factors of Success equal 80% of the points. We have assigned 25 points to decision latency. However, decision latency runs through all the other factors like veins. Slow decision latency is like plague that reduces progress flow. Exercising fast decision latency within the other factors keeps the veins clear and allows for smooth progress flow. Small projects with minimal scope help decision latency by reducing the number of decisions and getting fast feedback to promote quicker future decisions. A good executive project sponsor will promote faster decision in many ways. First, he or she will make fast decisions; second, he or she will create a pipeline of decisions to keep decisions flowing; and third, he or she will distribute the decisions to the team and to other knowledgeable people. A highly skilled technical team will benefit from the intuitive knowledge to make the right technical decisions. An emotionally mature team will work together to make rapid real-time decisions.

Looking at Table 17, we can see that highly skilled and skilled teams, paired only with good decision latency, will create high-value projects more than 70% of the time. However, combined with the effects of the other six skills, these teams should reap not only project success but also skyrocketing levels of customer satisfaction. There is much overlap between with the first six factors and the last four. However, the last four play a diminishing role in the success, value, cost, and customer satisfaction of software projects.

Now, let's take a look at these and see how they overlap.

Table 17: Value by Decision Latency Skills

| Value/Skill Level | Highly Skilled | Skilled | Moderately | Poorly Skilled |
|-------------------|----------------|---------|------------|----------------|
| Very High Value | 31% | 42% | 16% | 11% |
| High Value | 30% | 42% | 18% | 10% |
| Average Value | 18% | 43% | 25% | 14% |
| Low | 17% | 46% | 24% | 13% |
| Very Low | 13% | 30% | 34% | 23% |

Table 17 shows project value by the skill of the team.

FACTORS OF SUCCESS: USER INVOLVEMENT

"User involvement" was the number-one factor of success for many years. This year, we listed it as number seven. It would still be close to number one, if not for agile process and emotional maturity. Agile processes incorporate users naturally and also provide for fast feedback. Emotionally mature teams also have a good relationship with their users and tend to bring them into the process. Therefore, there is a reduced need to be proactive in bringing users into the project as a separate activity.

By the same reasoning, non-agile projects and emotionally immature teams will require investment in user involvement skills. A large percent of failed projects are projects that were completed, but then rejected by users and consequently not adopted. In addition, many projects were classified as "challenged" not because they were rejected by users but because users were dissatisfied with the results.

Another reason user involvement appears lower on the Factors of Success listing is because when we add this attribute to the top six factors, the success and value rates do not go up. User involvement does, however, improve the success rate if it is combined with other attributes in *non-agile projects*. Being highly skilled at user involvement can increase the success rate between 10% and 15% in non-agile projects, as shown in the Table 18. If you add highly skilled decision latency and user involvement, you can double your rate of success in non-agile projects. A team that is highly skilled in user involvement, exhibits good decision latency, and stays with small projects can come close to a winning hand.

Table 18: Resolution by User Involvement Skills

| Skill Level | Successful | Challenged | Failed |
|--------------------|------------|------------|--------|
| Highly Skilled | 38% | 51% | 11% |
| Skilled | 31% | 52% | 17% |
| Moderately Skilled | 26% | 53% | 21% |
| Poorly Skilled | 26% | 53% | 21% |

Table 18 shows "Modern" resolution of all software projects by user involvement skills, 2013–2017, within the CHAOS database.

FACTORS OF SUCCESS: OPTIMIZATION

Optimization should be one of the top Factors of Success. The problem is, optimization overlaps with many other factors. For example, optimization is a factor in small projects: one way you get small is by optimizing features and functions that are most important in the given microproject or service.

The second optimization takes place on the team. You want to optimize the size and talent of your team to the microproject at hand. Next, you'll optimize on delivery of value to your users and on customer satisfaction. You'll also want to optimize risk by doing the most important or least-risky things. (You may never do the riskier things; or perhaps you'll discover an easier way in the course of your work.)

The Standish Group developed OptiMix to optimize projects based on dependencies. OptiMix uses six different optimization scenarios: 1) maximum gain; 2) minimal risk; 3) maximum gain with calculated risk; 4) goal; 5) capability; and 6) true. *Maximum gain* provides you with the optimal set of projects that give the organization the greatest gain for the budget, regardless of risk. *Minimal risk* represents the optimal set of projects that give the organization the least risk for the budget, regardless of gain. *Maximum gain with calculated risk* represents the optimal set of projects that give the organization the least risk and greatest gain for the budget. *Goal* represents the optimal set of projects that are closest to the organization's goal with the greatest gain. *Capability* represents the optimal set of projects that are closest to the team's talent. And *true* is a combination of gain, risk, goal, and capability.

Table 19: Resolution by Optimization Skills

| Skill Level | Successful | Challenged | Failed |
|--------------------|------------|------------|--------|
| Highly Skilled | 42% | 47% | 11% |
| Skilled | 32% | 53% | 15% |
| Moderately Skilled | 28% | 53% | 19% |
| Poorly Skilled | 25% | 55% | 20% |

Table 19 shows "Modern" resolution of all software projects by optimization skills, 2013–2017, within the CHAOS database.

FACTORS OF SUCCESS: SAME

SAME (Standard Architectural Management Environment) is a consistent group of integrated practices, services, and products for developing, implementing, and operating software applications. SAME applies architecture principles to guide organizations through software technology transformations. It's very similar to enterprise architecture. In a recent study with the Antwerp Management School, working within the CHAOS Database, Eaglan Kurek found that projects using enterprise architecture demonstrated increased project success. We think this is an important finding, and it is another pointer to the importance of decision latency, since decisions about software architecture are premade. Decision latency for these issues is zero since there no decisions. If they are not premade through the SAME process, then there can be long and drawn out battles, making for major decision latency. This is not unusual—we have seen many technology battles fought over which tools and software to use and which systems should deploy the application.

Consider that organizations must adapt to new technologies; otherwise they could fall behind and become uncompetitive. However, it would not be prudent to have decisions on using new technologies made at the team level; at the same time, many projects cannot be started until such decisions are made. Recently, we saw a project fail because its chief architect had not made a new-technology decision; but meanwhile, four teams of developers had been sitting around waiting to find out in which direction they should proceed. An established enterprise architecture would have greatly helped this project and inspired the team.

Using SAME or an enterprise architecture decreases the learning curve for new tools and systems. This type of framework is helpful; on the other hand, note that the CHAOS charts do not show it as a major factor in achieving a "winning hand." It's a factor that calls for some awareness and perhaps a minor investment.

Table 20: Resolution by SAME Skills

| Skill Level | Successful | Challenged | Failed |
|--------------------|------------|------------|--------|
| Highly Skilled | 28% | 49% | 23% |
| Skilled | 28% | 51% | 21% |
| Moderately Skilled | 32% | 55% | 13% |
| Poorly Skilled | 32% | 57% | 11% |

"Modern" resolution of all software projects by SAME skills, 2013–2017, within the CHAOS database.
Page 31: [Section]: Factors of Success: PMI/Execution

group of integrated practices, software applications. SAME applies architecture transformations. It's very similar Management School, working within the architecture demonstrated increased pointer to the importance of decision latency for these issues is zero. If there is no process, then there can be long and unusual—we have seen many technology systems should deploy the application.

they could fall behind and become uninteresting new technologies made at the time such decisions are made. Recently, we made a technology decision; but meanwhile, four in which direction they should proceed. This project and inspired the team.

search for new tools and systems. This type of study do not show it as a major factor in success and perhaps a minor investment.

Skills

| Passed | Failed |
|--------|--------|
| 73% | 27% |
| 31% | 69% |
| 13% | 87% |
| 11% | 89% |

FACTORS OF SUCCESS: PM/EXECUTION

PM/Execution is a major overlap—as well as a conflict—with the agile process. Since Scrum is the major agile method, we will use it as our role model. The execution method is agile or Scrum. Therefore, non-agile requires an execution model. Waterfall is considered a major execution model, but—as mentioned before—its results are fairly unfavorable. Many other models use methods similar to agile. If you do not use Scrum or some other full-function agile process, then you need to develop your own lean process. A fat process, especially coupled with PM-heavy tools, will most likely produce poor results. You can mitigate the effects of such processes and tools by developing fast decision latency procedures.

The other major overlap and conflict is project manager versus no project manager. Interestingly, when we add “highly skilled project manager” to our winning hand, we find no change. We believe this is because Scrum splits most of the project-manager duties between the Scrum master and the Scrum owner. Therefore, project managers have less of a role in agile projects. Many current project managers are moving to become either Scrum master or Scrum owner.

Non-agile projects will need at minimum a moderately skilled project manager. Our results show very little difference between those who are highly skilled and only moderately skilled. However, poorly skilled project managers have a negative effect on a project. At the very least, a project manager should have the basic mechanical skills of planning, tracking, and controlling. Also, any project manager should be aware of the negative effects of decision latency.

Table 21: Resolution by PM/Execution Skills

| Skill Level | Successful | Challenged | Failed |
|--------------------|------------|------------|--------|
| Highly Skilled | 34% | 50% | 16% |
| Skilled | 30% | 51% | 19% |
| Moderately Skilled | 32% | 55% | 13% |
| Poorly Skilled | 23% | 58% | 19% |

Table 21 shows “Modern” resolution of all software projects by PM/Execution skills, 2013–2017, within the CHAOS database.

FACTORS OF SUCCESS: CLEAR BUSINESS OBJECTIVES

We dropped "clear business objectives: from the Factors of Success, even though for many years it was number three in importance. In addition, it was a measurement of success. We would ask, "What was your objective? Did you meet your objective?" If the answer was yes, that was how the project would be categorized. Over the years, however, this metric has lost much of its luster and has given way to more innovative ideas. One component of clear business objectives was what we called "being on the same page" (in terms of features and functions. We now realize this is not viable. What is viable is having a big picture—and only the team needs to be on the same page. The overlap here involves the team's emotional maturity skills. The vagaries are worked out within the flexible or agile team.

Looking at Table 22, we see organizations that are being poorly skilled have more successful projects. Remember, in our "modern" definition, success includes customer satisfaction. The success numbers do go up when using the traditional measurement of "on target," which would equate to clear business objectives. When clear business objectives skills were applied to the "winning hand," the success numbers decreased or went down. We wrestled with dropping this factor from the Factors of Success, but in the end, it was all about the numbers.

Table 22: Resolution by Clear Business Objective Skills

| Skill Level | Successful | Challenged | Failed |
|--------------------|------------|------------|--------|
| Highly Skilled | 27% | 56% | 17% |
| Skilled | 26% | 55% | 19% |
| Moderately Skilled | 34% | 47% | 19% |
| Poorly Skilled | 44% | 41% | 15% |

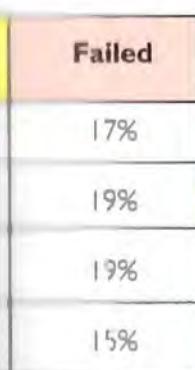
"Modern" resolution of all software projects by clear business objective skills, 2013–2017, within the CHAOS database.

OBJECTIVES

Even though for many years it was part of success. We would ask, "What was the problem that was how the project would be run?" and has given way to more what we called "being on the same page" What is viable is having a big picture— which involves the team's emotional maturity

more successful projects. The success numbers do go up to clear business objectives. In the end, the success numbers decreased from the factors of success, but in the end, it was all

Objective Skills



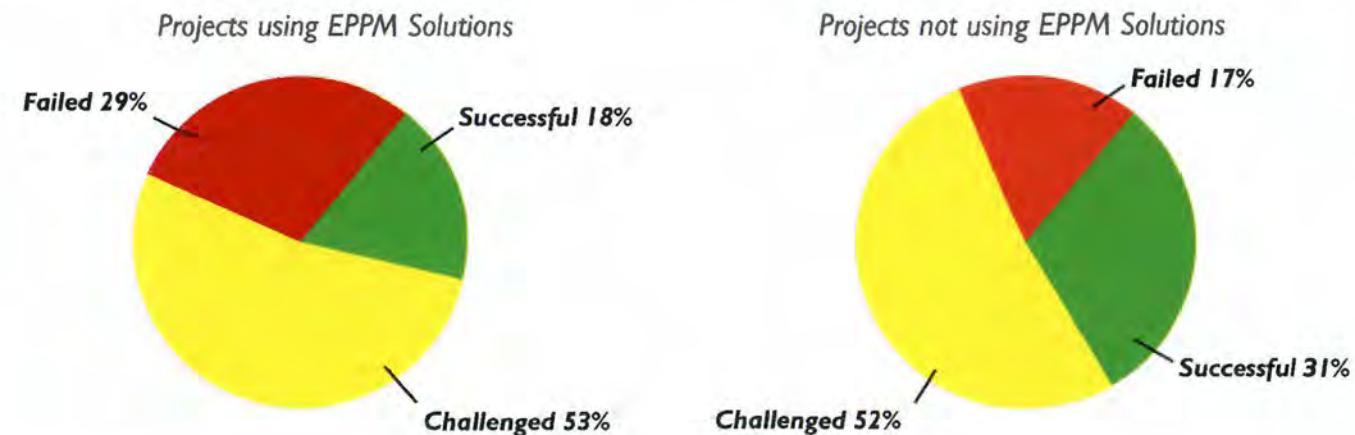
Projects using EPPM database.

FACTORS OF SUCCESS: PROJECT MANAGEMENT TOOLS

Also on the Factors of Success list for many years was "project management tools." We dropped this metric a few years ago and replaced it with Standard Architectural Management Environment (SAME). We think basic and lightweight tools do help, but their features and functions have grown to the point of negative value. Many organizations use them as a crutch rather than an aid. We loosely call these tools *enterprise project portfolio management (EPPM)* solutions. In some ways, investing in them is easy. Changing human behavior is much harder. Investing in helping the project sponsor looks difficult and expensive. Investing in decreasing decision latency also looks difficult and expensive. However, the cost to move the investment from an EPPM solution to the "winning hand" will greatly increase software project health.

EPPM solutions should increase the value of your project investments. Our research shows they do not. EPPM solutions should increase the customer satisfaction derived from your project investments. Our research shows they do not. EPPM solutions should increase your project success rates. Our research shows they do not. EPPM solutions should cut your project costs. Our research shows they do the opposite. Finally, EPPM solutions should provide a clearer picture of your project progress. Our research shows that in many cases, they do not. We suggest that you either stop or reduce using them.

Effect of Using Enterprise Project portfolio management (EPPM) solutions



Projects using EPPM solutions compared against projects that do not use them. Data is based on "modern" resolution of 2013-2017 software projects from the CHAOS database.

FACTORS OF SUCCESS: COMPLEXITY

Complexity is one of the main reasons for project failure. Table 23 shows the resolution of all software projects by complexity from 2013 to 2017 within the CHAOS database using the modern definition of success. The results show that 39% of very easy projects were successful. Very complex projects, on the other hand, scored high as challenged (58%) and failed (27%).

Inside every complex problem are simple solutions. Complexity is often caused by size, conflicting goals, large budgets, and the egos of executive sponsors. Complexity creates costs and confusion. True, if you are trying to do something complex and innovative, that project may well take longer and cost more. It took us five years to invent and develop the software for our OptiMlx project, because it was not only complex but had never been done before. Breaking new ground is one of the attributes we use to categorize complexity.

Determining complexity is... complex. We use about 25 attributes from the project profile, such as number of stakeholders, diversity of user profiles, and number of innovations. Our adjudicators assign complexity of project using a range based on these 25 attributes. The adjudicator uses a generic name and types to help assign the range. Complexity ranges from very complex to very simple. As you can see throughout this paper, we use ranges for almost all attributes. In the profile we have detailed numbers, but we convert these to a standard range prior to accepting the project into the CHAOS Database.

Table 23: Resolution by Complexity

| | Successful | Challenged | Failed |
|--------------|-------------------|-------------------|---------------|
| Very Complex | 15% | 58% | 27% |
| Complex | 20% | 56% | 24% |
| Average | 29% | 53% | 18% |
| Easy | 38% | 48% | 14% |
| Very Easy | 39% | 48% | 13% |

Table 23 shows the resolution of all software projects by complexity from 2013 to 2017 within the CHAOS database..

the resolution of all software projects using the modern definition of Very complex projects, on the

is often caused by size, conflicting goals, large teams and confusion. True, if you are trying to take longer and cost more. It took us five years because it was not only complex but had never been used to categorize complexity.

numbers from the project profile, such as number of users. Our adjudicators assign complexity of user cases a generic name and types to help identify simple. As you can see throughout this report we have detailed numbers, but we convert these into the HAOS Database.

complexity

| Challenged | Failed |
|------------|--------|
| 100% | 27% |
| 100% | 24% |
| 100% | 18% |
| 100% | 14% |
| 100% | 13% |

© 2014 ProjectSmart, Inc. HAOS database.

SKILLS OF THE FACTORS OF SUCCESS: OVERVIEW

The search for the root cause of why software projects fail has been a 25-year journey. This journey covered over 120,000 project profiles, countless work groups, and interviews with individuals ranging from developers to CIOs. We have written, read and digested thousands of books, research papers and articles. We executed over 200 various and distinct types of surveys with a range of 30 to 100 questions in each. Our conclusion to the research is this: The root cause of poor software project performance is long decision latency. And the converse is also true: The root cause of good software project performance is short decision latency.

A root cause is an underlying or fundamental reason for failure. Identifying the root cause is essential if one is to address and fix a problem or prevent it from happening in the future. Not knowing the root cause often focuses us on issues with existing systems, rather than addressing an underlying problem. In the case of software development projects, that focus on existing systems actually brought us a continuous stream of new methods and a host of project management tools that only increased the complexity of the process. Not knowing the root cause led to sophisticated governance, oversight, and compliance schemes. But the heart of the problem was poor decision latency.

Many of the so-called cures actually made decision latency worse, and the result is that software project performance has not improved. The slower the decision latency, the more likely a project is to experience stress as well as failure. Based on this one factor, we can predict a project's outcome with a certainty of about 90%.

Decision latency is not only the root cause of failure; it's the leading factor for success. Reducing decision latency turns out to be a thread that runs through all of the 50 skills that we've determined contribute most to project success. However, these skills will only work if the organization and its teams recognize long decision latency as a problem and take action to prevent it.

The root cause of good software project performance is short decision latency

The Standish Group has boiled down thousands of reasons why software projects are successful into 300 common reasons. We have also reduced the reasons why software projects are challenged or have failed into 300 common reasons. A few years ago, we added decision latency to the mix, and it quickly became clear that it was particularly significant and influential. We also added decision latency to a number of assessment and profile instruments carried out over the last 10 years, and again, it was very clearly a major contributor to success or failure.

The faster the average decision latency, the greater the chances of success and higher value.

We define decision latency as the time between the surfacing of an issue and its resolution. Over the last few years, in future as well as post-mortem project assessments and in our workshops, we have studied decision latency as a factor for success, and we have concluded that it is not just one thing—it is everything. Decision latency is part of being a good sponsor. Decision latency is part of being a good teammate. Decision latency is part of the agile process. Long decision latency plays a big part in destroying a project. From our research, we have distilled the top 10 factors involved in project success, as well as the personal skills that best support these factors. This is our final list, and decision latency is at the top.

Consider each skill that we name and describe in what follows, and ask yourself: How good is my organization at doing this? Mark them as either “very skilled,” “somewhat skilled,” or “poorly skilled.” These are the questions that we ask you to answer for our Project Environmental Benchmark. The benchmark weights them by their importance in reducing decision latency and improving project performance. When you’ve finished our assessment, you’ll know the top three skills that you should work on to improve your decision-latency skills. The result will also give you a benchmark score so you’ll know how your organization stacks up against other organizations. You can obtain access to the benchmark on our website.

SKILLS OF THE FACTORS OF SUCCESS: DECISION LATENCY SKILLS

If your organization wants to reduce decision latency, you must first be aware of it as an issue; only then can you be proactive in reducing decision time. You might try having your teams benchmark the time it takes for them to make decisions, and set a goal for improvement. Teams need to make faster decisions, but management and stakeholders are also responsible for slowing progress by delaying their own decisions.

There are five skills that aid in reducing decision latency.

1. Reducing decision latency reduces the interval between decisions.

How do you do this? You set in place a decision process that keeps pace with the time frames already set forth as part of the project's goals. If this is a problem for your organization or team, The Standish Group suggests that your organization set standards, roles, and responsibilities, and goes on to benchmark the decision process. In any project, participants are constantly faced with making prompt, effective decisions. A project sponsor, as well as team members who can guide discussions and encourage creative thinking, can help a team to fully consider all the alternatives, while still reducing decision latency.

2. Make quick decisions. Strive to make decisions in the smallest possible time frame. Make it automatic and a natural habit to make quick decisions, and you will be helping your organization in its drive to reduce decision latency.

3. Distribute decisions. Distribution of the power to make decisions is done by segmenting those decisions based on roles and responsibilities. Segmentation of decisions should be based on the impact of the decisions, from low to high. The biggest barrier to rapid decision latency is caused when one or two people become bottlenecks in the decision process. We have seen this in many assessments and benchmarks. Distributing decision-making power relieves these bottlenecks. In order to improve your decision distribution skills, start with empowering your project team to be more self-directed.

If your organization wants to reduce decision latency, you must first be aware of this issue; only then can you be proactive in reducing decision time.

4. **Rapid consensus.** A consensus is a general agreement or an accord, indicating harmony regarding the project's direction and future decisions to be made.

A decision pipeline is a method that identifies the right people to make decisions quickly and easily at the right time.

How skilled is your organization at rapidly gaining and maintaining consensus? The Standish Group suggests that project teams practice gaining consensus to improve their skills and reduce decision latency. The team can benchmark the time it takes to reach a consensus and set goals to shorten the lapsed time from the beginning to the end, when a consensus is reached. Consensus should help to reduce decision reversals, which are disastrous for decision latency.

Eric Newcomer, Chief Architect for Citi's Treasury and Trade Services, notes, "You build a decision pipeline by starting with small decisions and building trust. You definitely want to be able to delegate decisions. Usually I would give somebody who's working for me a trial run on a decision that maybe wasn't too consequential or too long-term and see how it worked out. And in cases where somebody would bring a proposal to me, and was very sure of it, I'd give them the benefit of the doubt, let them make the decision, and see how it worked out."

5. **Decision pipeline.** A decision pipeline is a method that identifies the right people to make decisions quickly and easily at the right time. It is also a method to engage more people in the decision process. As a former member of the U.S. Naval Reserve and the commander of a series of PT boats in the Pacific theater of World War II, U.S. President John F. Kennedy was trained to make quick decisions. He also empowered his cabinet—in an early example of a decision pipeline—to make quick decisions. How skilled is your organization at making fast decisions through a decision pipeline? After all, you do need to have a method that helps you identify how and when decisions should be made throughout the life of a project. (Dezider.com will help you create a decision pipeline.)

SKILLS OF THE FACTORS OF SUCCESS: MINIMUM SCOPE SKILLS

Another important set of skills helps keep a project's scope to the absolute minimum set of necessary features and functions. ("Scope" refers to the total or sum of the deliverables and services to be provided through an IT project or program.) A "minimum viable product (MVP)" is a product that offers just enough features to satisfy early customers and to provide feedback for future improvements. We have long recommended "microprojects" as a way of doing just that. Using our OptiMix Service, a project team can clearly see which set of requirements offer the highest value and best minimize scope.

The Standish Group has identified five skills that promote minimum scope.

1. Small projects. A small project requires between six and eight team members, and lasts no longer than six months. Small projects reduce decision latency by simply being small. They also earn a higher-than-average value rating because their size constraint limits functions and features to those that are the most important and valuable. Larger projects often end up with medium- or low-value functions and features because of increased costs and longer time to deliver. Moreover, smaller teams are generally nimbler and require less time to make decisions.

2. Managing scope. Reducing scope also reduces decision latency. A statement of project scope includes a list of deliverables and project objectives and a description of project success criteria, such as cost, quality, and schedules. All time and cost estimates are based on a project's scope. How skilled is your organization at containing the scope of a project? The Standish Group suggests that your project teams determine how much can be accomplished realistically in the time frame available.

3. Managing trade-offs. A trade-off is the act of exchanging one or more items for other items as a compromise. For example, a stakeholder or sponsor might drop or delay features and functions if that will facilitate an earlier delivery. Stakeholders may collaborate on exchanging support for one function by dropping support for another. Managing trade-offs well decreases decision latency. To improve your organization's skills at managing trade-offs, practice and collaborate on trade-off decisions.

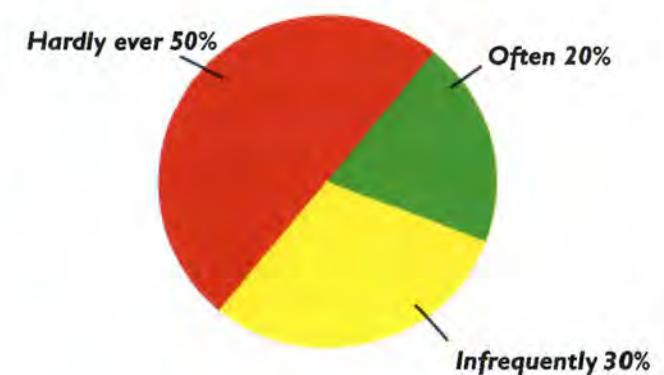
**Reducing scope
also reduces
decision latency.**

4. The Rule of Three. This rule suggests that you should limit the number of objectives to three or fewer. Again, this will reduce decision latency. Limiting the number of objectives gives you a much better chance of meeting those objectives. Having more objectives will unnecessarily complicate the project by making it larger and involving more people. Breaking up large projects into microprojects helps limit the objectives.

5. Simple vision. A simple vision is easy to understand and is not too artificially elaborate. A simple vision promotes rapid decisions and reduces decision latency. "Vision" refers to the act or power of anticipating that which will or may come to be. A simple vision is an easy-to-understand description or direction of a product or service to be delivered in the future. It should be clear about the benefits the project or product is expected to deliver. How skilled is your organization at conveying a simple vision?

The Standish Group has utilized many different types of research vehicles, from TCO studies to project optimization, modernization, single-project assessment, project postmortems, and other research cases, to look at minimizing scope. We also have data from workshops and focus groups that from time to time over the last 20 years have looked at this topic. Based on these casual observations, we have reached this conclusion: Of the features available to most mission-critical applications, 20% are used "often," 30% "infrequently," and a full 50% are used "almost never."

Features Used

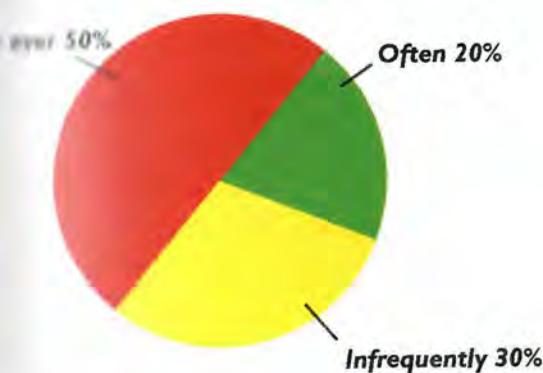


Features commonly seen in custom applications: How often are they actually used? Remember, the more objectives are attached to a project, the more people become involved, and the more complex it becomes.

limit the number of objectives to three or four. A number of objectives gives you a much larger scope, which will unnecessarily complicate the project. Breaking projects into microprojects helps limit

the project to a more manageable size. A simple "Vision" refers to the act or power of creating an easy-to-understand description or plan. It should be clear about the benefits the organization will receive from the project at conveying a simple vision?

Features Used



and in custom applications: How often are they used? Moreover, the more objectives are attached to a project, the more involved, and the more complex it becomes.

SKILLS OF THE FACTORS OF SUCCESS: PROJECT SPONSOR SKILLS

The project sponsor is responsible for the success of the project and for returning value to the organization. The project sponsor provides the leadership, the vision, the motivation, and the resources. We have identified 10 traits and 50 skills needed to be a good project sponsor; and they are presented in our book *The Good Sponsor*. We also offer a self-assessment instrument as well as a class on how to be a good sponsor.

For this book, however, The Standish Group has identified the five most important principles and skills that promote good project sponsorship.

1. Inspiration. Inspiration is the process of emotionally stimulating teams and organizations toward a creative outcome by moving a project toward its ultimate goals. The project sponsor must study how best to inspire the project team. That inspiration will be infectious and will keep the team going. How skilled are your project sponsors at providing inspiration? In order to more fully answer this question, you should have your sponsor take the Good Sponsor Assessment.

2. Perspiration. It takes hard work and a commitment to the project or program to be a good sponsor. A sponsor also needs to strive to contain the scope of a project and to ward off overzealous stakeholders. We suggest that the best way for sponsors to contain and reduce scope is to understand the value of each major feature and function. The path to optimization is through feature relationships.

3. Imagination. Imagination is the ability to form new ideas and concepts of objects and services not yet realized. It is the creative power to innovate, invent, and originate. It is the ability to envision how things can be versus how they are today. It is the ability to see the future. It is the ability to daydream. And imagination is the project sponsor's responsibility; a good sponsor knows how to ask the right questions and how to imagine the possible answers. We suggest you imagine how you might structure the project to stimulate your team, your leadership, and your stakeholders.

**Sponsor inspiration
will be infectious
and will keep the
team going.**

4. Decisiveness. Project sponsors will be called upon to make many decisions over the life of a project—and those decisions must be made promptly and firmly. Standish Group research shows that decision latency is a major contributor to project delays and failures. In addition, projects in which decisions are rapid have a greater success rate than those in which decisions are delayed or extended. Time is the killer of projects. Sponsors need to consider the advantages and benefits of a decision pipeline. They also need to promote decision acceptance and moving on from unpopular decisions.



5. Resourcefulness. Resourcefulness is the ability to assemble the right resources for the right work at the right time. Project resources are anything that is assigned to an activity or needed to complete an activity: equipment, people, buildings, and the like. Connections, in terms of people, are a resource as well; gaining and keeping connections is a basic skill a project sponsor needs to have. That's one of the reasons a sponsor requires good communication skills in speaking and in writing.

Bill Heil, former president and COO of WebEx, notes, "We had a tremendous executive sponsor for our support CRM system. We were using Siebel for support and it was working fine, but it wasn't the same platform as for sales. The sales department was using Salesforce.com, so we decided to move support over as well in order to have a single enterprise view of the customer. Support had three distinct roles, and the executive sponsor understood these roles really well. He also had a pulse on the support leadership. He focused on the minimum set of success criteria that would make the teams happy. He was absolutely in charge of scope control. One of the key things an executive sponsor can do is control scope and make sure it aligns with the success criteria."

SKILLS OF THE FACTORS OF SUCCESS: AGILE PROCESS SKILLS

Agile software development is a group of software development methodologies based on iterative development. Requirements and solutions evolve through collaboration among self-organizing cross-functional teams. Agile processes promote rapid feedback, testing and delivery. We first described the basic agile process in our 1998 CHAOS Report. Later, in 2001, a group of 14 developers created the Agile Manifesto, which defined this philosophy.

The agile process promotes people over process and liberates people to be productive and creative. Scrum has emerged as the most popular form of the agile methods and enjoys wide adoption.

The Standish Group has identified five important skills that promote good agile methods.

1. An iterative process. In software development, "iterative" basically means development by trial and error. It provides for a constant cycle of design, development, test, user acceptance, and then back to design or redesign. The idea is that stakeholders get to see the system in small steppingstones before it is fully built, thereby allowing changes to occur without having a major impact on the project's cost and schedule. The iterative development process allows for gradual course correction and reduces decision latency.

2. Time boxing. "Time boxing" means setting a firm date for deliverables or steppingstones. These intervals could be weekly or monthly (The Standish Group recommends two-week intervals). Time boxing provides for regular delivery or demonstration of working software. Done right, time boxing allows for the delivery of the highest-priority or most valuable features and functions, as defined by the stakeholders. Time boxing reduces decision latency, because decisions need to be made "inside the box." How skilled is your organization at developing and maintaining time boxes?

Jeff Sutherland, co-founder of Scrum, suggests there are three things a good Scrum product owner needs to have: knowledge, availability, and the ability to make decisions. They need to know the market. They need to know how to position a product to enter the market. If it is a legacy product, they need to know how to improve it to meet the competition.

Time boxing reduces decision latency because decisions need to be made "inside the box."

3. Refactoring. "Refactoring" means removing useless, dead, and inefficient code to make what remains easier to comprehend, as well as more maintainable, more efficient, and more amenable to change. Pruning improves the understandability of the code and changes its internal structure and design. The Standish Group suggests that your organization make refactoring a common and frequent task. The focus for many software development projects is to ensure completeness, when in fact it should be the opposite. Deficiencies will show up quickly and can be dealt with at the time of their exposures. Focusing on high-value items reduces decision latency.

4. Retrospective meetings. A retrospective is a meeting held by a project team at the end of a project, process, or iteration to discuss what was successful about the project or time period covered by that retrospective, what could be improved, and how to incorporate successes and improvements in future iterations or projects. The Standish Group suggests that after each steppingstone and/or microproject, your project teams should take a little time to look back on what went right, what went wrong, and what changes need to be made for the next iteration or microproject. These meetings should be used to figure out how to reduce decision latency.

5. Staged deployment. Staged deployment provides for the early delivery of clear benefits. Staged deployment optimizes scope and has a number of other benefits: for instance, rolling out features and functions in stages takes the focus off the complete project to gain value sooner. Because it implements and delivers functionality in stages, users can learn and experience these features easier and faster. Staged deployment reduces decision latency because the team only focuses on the current delivery. How skilled is your organization at delivering functionality in stages?

Jeff Sutherland, co-founder of Scrum, suggests a good Scrum product owner needs to know how to maximize the business value of the product or project. He or she needs to explain the requirements clearly. In turn, the team will help the project owner develop the work backlog

**Refactoring means
removing useless,
dead, and inefficient
code.**

nd inefficient code to make what remains efficient, and more amenable to change. Its internal structure and design. The a common and frequent task. The focus when in fact it should be the opposite. of their exposures. Focusing on

Refactoring means removing useless, dead, and inefficient code.

the early delivery of clear benefits. Staged delivery for instance, rolling out features and value sooner. Because it implements and features easier and faster. Staged on the current delivery. How skilled

product owner needs to know how to needs to explain the requirements clearly. Planning

SKILLS OF THE FACTORS OF SUCCESS: TALENTED TEAM SKILLS

Successful projects result from smart, trained people. Not surprisingly, one of the key project success factors identified in The Standish Group's CHAOS research is a talented staff.

There are five key fundamentals that ensure staff competency.

1. Competency.

A competent team has the level of skills necessary to complete project activities within agreed-upon times and schedules. Team members who demonstrate these skills will instill confidence in their colleagues. The Standish Group suggests that a project team should include highly talented people who understand both the business and the technology involved. Doing this reduces decision latency, because carefully chosen people are more likely to have the knowledge necessary to make an informed decision. Team skills should, of course, correspond with the skills needed for the project, but those skills may need to change as new innovations become apparent. How skilled is your organization at maintaining highly talented and competent people to work on your project teams?

2. Balanced teams. The Standish Group suggests that your organization build balanced and diverse teams. This will avoid "superstar syndrome," in which particular members want to dissect every decision and no issue is too small for a debate: the result is a team that doesn't work together well or suffers from destructive chemistry. Building and maintaining equilibrium among team members is a balancing act in and of itself. Additionally, consideration must be given as to whether the skills needed for a particular project are available in-house or can be developed, taught, or bought, and whether they will be available when needed. How skilled is your organization at balancing its teams?

3. Motivation. Effective motivation starts with incentives that are meaningful and earned. Everyone on the team should have an equal opportunity for recognition. It's important to understand what motivates your team. Realize that incentives can be negative as well as positive. Interestingly, poor decision latency causes people to be less motivated and lose interest; rapid decision-making tends to improve motivation, because results come more quickly.

Teamwork makes the dream work.

4. Team chemistry. The Standish Group suggests that your organization appraise the chemistry of its project teams and make changes if required. We have all seen enough TV programs and movies to recognize good and bad chemistry. Some actors just work better together. The same is true with project team members. Building and maintaining team chemistry is an ongoing process—and by the way, good decision latency improves team chemistry.

5. Self-directed teams.

Self-directed individuals actually work better in a chaotic situation. As the work moves from one area to the next, the group moves with that competency. An optimized self-directed team contains between 5 and 10 members, 6 being the optimum. Self-directed teams rely more on individual talent than on any hierarchical or seniority structure. That's why they require some relaxation of the usual management structures and oversight. Self-directed teams that work well together also improve decision latency, because they are able to make quick decisions.

Great Britain's King Charles I (1600-1649) and his archenemy Oliver Cromwell, a leader of the Parliamentarians, had been locked in a civil war for more than two years. The summer and fall battles of 1644 resulted in a stalemate. Cromwell, unhappy with the performance of his army in those battles, reinvented the organization and gave it a new name: the New Model Army. The New Model Army went through months of special training, combat exercises, and battle simulations. Officers were schooled in battle strategies and tactics.

The Battle of Naseby took place on June 14, 1645, in Northamptonshire, England. Under Cromwell, the New Model Army, made up of 3,000 men, was deployed along a parallel ridge to battle Charles' Royalist troops, which numbered 10,000. The Royalists attacked along Cromwell's front line. Then Cromwell set a trap. Charles had driven back Cromwell's left wing and followed them in pursuit as they fled. This allowed Cromwell's cavalry to attack and overwhelm the Royalists' position. When the battle was over, the New Model Army had lost only 200 men in the encounter, but the Royalists had nearly 1,000 dead and 4,500 taken prisoner. Perhaps this proves, once again, that a well-trained and competent team makes a big difference.

SKILLS OF THE FACTORS OF SUCCESS: TEAM MATURITY SKILLS

In his book *Chaos Monkeys*, Antonio Garcia Martinez wrote, "With the right team, no man or organization can stand against you and you will ultimately triumph. With the wrong team, you will produce internal problems even faster than the external world can and your eventual death will effectively be a suicide." We have identified 10 principal traits and 50 skills that make for a good project teammate; They are presented in our book *The Good Mate*. We also have an assessment instrument available, and we run classes on how to be a good mate.

For this book, however, The Standish Group has identified the five most important principles and skills that promote good project teams.

1. Mutual influence. Having influence means each team has the power to shape how team members feel and act as they work on a project. They have influence over each other, but they do it without managing, leading or directing their colleagues: This is the essence of being a good team member in a peer-level organization. A good team member can also accept influence from his or her colleagues. It's the true measure of good team skills.

2. Mindfulness. Mindfulness is the product of a disciplined mind. It cultivates awareness, alertness, consciousness, and being in the present. It teaches you to observe and accept things as they happen. There are five basic mindfulness skills: meditation; managing emotional flooding; emotional regulation; discernment; and diffusing stress. Meditation is the act or process of spending time in quiet thought to help you focus. Emotional flooding is when cortisol and adrenaline flood the bloodstream and activate the "fight or flight" response system. Good teammates learn to recognize when another teammate is stressed.

3. The Five Deadly Sins. The Five Deadly Sins of project management are overambition, arrogance, ignorance, abstinence, and fraudulence. You are likely to encounter one or another of them just about everywhere. They are part of all project and team ecosystems, healthy and unhealthy, and everyone is guilty of one or more of them from time to time.

Influence is the true measure of good team skills.

4. Problem-solving. There will be many problems to solve during a project undertaken by a self-directed team. To be successful at problem-solving, the team must first have a clear, concise description of the issue at hand. They can start by looking at what has been done in the past with other teams and use that knowledge to solve their current problems. They can also look at similar projects and use them as role models. They can brainstorm and talk with internal and external users. One of the best tools for dealing with these issues as they come up is empathy for teammates. In addition, a good team member must have the skills to recognize the Five Deadly Sins and head off problems.

5. General team maturity. The hallmarks of a mature team are communication, acceptance, respect, civility, being driven, and being a good "confrontationist"—by which we mean being able to argue a point yourself or mediate the arguments of others without rancor or insult. Communication is about how you communicate inside the team. (For example, communications inside the team are made in real-time—before and while the events are unfolding.) Acceptance is the ability to have mutual influence and accept compromise to overcome gridlock. Acceptance nurtures common admiration. Respectfulness requires that you consider the needs of your teammates and honor them in all your interactions, being considerate of their investment of time, emotions, and resources in the project and its goals.

In any team, there will be many times that people just do not agree with each other. In these times, the teammates need to confront the issues head-on; letting them fester will only make matters worse. Good teammates should be good confrontationists; they need to know how to deal with situations in which people oppose or challenge each other, and they're not afraid to manage disagreement. Being civil to one another makes a happy team, and a happy team is more productive. A good team is driven to be successful; it focuses on the deliverables, and knows the difference between activity and progress.

To be successful at problem-solving, the team must first have a clear, concise description of the issue at hand.

during a project undertaken by a self-managed team must first have a clear, concise description of the requirements in the past with other teams and use them in similar projects and use them as role models. One of the best tools for dealing with requirements is a good team member must have the

To be successful at problem-solving, the team must first have a clear, concise description of the issue at hand.

work with each other. In these times, the team will only make matters worse. Good communication is key to deal with situations in which there is disagreement. Being civil to one another is important. A good team is driven to be successful; it is focused on activity and progress.

SKILLS OF THE FACTORS OF SUCCESS: USER INVOLVEMENT SKILLS

The Standish Group highly recommends that software projects build in substantial user involvement during the information-gathering process and as an aid in decision-making. User feedback, requirements review, basic research, prototyping, and other consensus-building tools are a must, and project teams should include end users as well as IT personnel. The CHAOS research clearly shows that projects that lack user involvement perform poorly, and besides, including the user in the decision-making process reduces decision latency.

The Standish Group has identified five important skills that promote good user involvement.

1. Subject matter expertise. The Standish Group suggests that finding and engaging the right users to aid with a project is extremely important. This means users who already have the business knowledge, want the project, wish to provide the effort, and have the time to participate. The right users will have an understanding of how other types of users will use the new system and what their needs and wants might be.

2. Motive, means, and desire. Nothing is more frustrating to the team than a user who has the subject matter expertise, but who lacks the motive, means, and desire to participate. Users with motive, means, and desire will greatly reduce decision latency. The project team should ensure that the users and their management team are aware of the time they will need to spend on the project and away from their normal activities. The project team should include both time and cost estimates for this involvement in the project, as well as an estimated return on investment.

3. Attentive listening. Pay close attention to what your users say, and you'll reap benefits. Attentive listening provides encouragement and makes people feel important. In addition, tone of voice and body language provide hints as to how a person feels about getting involved in a project. The project team can use this information in their interactions and to move the project forward. In addition, attentively listening to users' needs reduces decision latency because it helps the team focus.

Having the right user involvement at the right times reduces decision latency.

4. Real user needs. Managing a project's scope—the sum of its deliverables and the services it is expected to provide—is a never-ending task. Business analysts and other software professionals must work closely with users and stakeholders to document a business process, map system flows, and develop user models—in other words, to produce a product that meets user needs. Focusing on those real user needs—the “must-haves” rather than a “kitchen sink” of features — requires the attentive listening referred to above, as well as a good grounding in the basic skill of empathy.

In their book, *Tuned In: Uncover the Extraordinary Opportunities That Lead to Business Breakthroughs*, Craig Stull, Phil Myers, and David Meerman Scott describe a “buyer persona,” which you could translate into a “user persona.” This would help the team understand the user needs of different stakeholders in order to address their requirements more effectively. User involvement is a major reason for project success or failure. Teams that are tuned in to the users’ true needs will have greater success than teams that just go through the process, no matter how good that process is.

5. Absorption. If you want your organization to improve its user absorption and feature adoption rates, you must think in terms of continuous change—of adding new functionality or changing functionality that is current. Continuous change helps to keep software modern and useful, reducing the need for large, mass absorptions. Second, you should always strive to reduce system complexity. Third, you must look for ways to reinforce conservation of familiarity—maintaining the user experience while making changes that are intuitive for the user.

For its Virtual Fence Project, the U.S. federal government spent a billion dollars to prevent or reduce the number of immigrants crossing the 2,000-mile border between Mexico and the United States. The project failed because the users—border patrol agents—could not absorb or adopt the features and functions of the new system. The teams developing the project had failed to interview its end users or to get them involved in the planning process.

deliverables and the services it is and other software professionals must map system flows, and develop needs. Focusing on those real user requires the attentive listening referred

to Business Breakthroughs, persona," which you could translate needs of different stakeholders in

For its Virtual Fence Project, the U.S. federal government spent a billion dollars to prevent or reduce the number of immigrants crossing the 2,000 mile border between Mexico and the United States. The project failed because the users—border patrol agents—could not absorb or adopt the features and functions of the new system. The teams developing the project had failed to interview its end users or to get them involved in the planning process.

SKILLS OF THE FACTORS OF SUCCESS: OPTIMIZATION SKILLS

Optimization is the process of making a project environment as efficient as possible. Optimization is a skill in itself. Consider the art of optimizing and managing scope. (Scope is the total of the deliverables and services to be provided from an IT project or program.) You optimize a project based on the business values that offer the greatest benefits. You then move on to optimizing the team and to benchmarking and optimizing decision latency.

Improving your optimization skills takes practice. One way to do this is by using your project retrospectives as an optimizing vehicle

The Standish Group has identified five important skills that promote optimization.

1. Decision optimization. Always try to make decisions firmly and without delay. Even minor decisions with minor delays cause frustration for team members. If they perceive that the organization does not feel the project is important enough to make timely decisions, then they will feel they are similarly entitled to slack off and not put forth maximum effort. Slow and irresolute decision-making causes unnecessary delays and increased costs. How skilled is your organization at making timely decisions?

2. Optimal team. The Standish Group suggests that the ideal team size is six people—not including the owner or the project sponsor. A team is a cooperative unit—a collection of people, assigned to perform a well-defined function to advance a project. Team members may be part-time participants and have other primary responsibilities. Their skills should be varied, but they should, of course, match the project requirements. The diversity of a team adds considerable strength to its efforts. How skilled is your organization at creating and maintaining an optimal team that is the right size and includes the right skills and amount of diversity?

3. Complexity optimization. Intricacy—many parts that interact with each other in multiple ways—contributes to confusion and increased costs. The best way to solve a complex problem is to find a simple solution. In fact, there may be many simple solutions hiding inside a complex problem! Complexity, as you might expect, also increases decision latency.

The diversity of a team adds considerable strength to its efforts.

Complexity also increases decision latency. Try to keep it simple.

4. Delivering bad news fast. Team members need to know about unpleasant or detrimental issues as soon as they are discovered or even predicted. Bad news should be delivered with action plans and generally without excuses or blaming others. In other words, deliver bad news early and bravely, with solutions.

The longer bad news is allowed to fester, the more severely it will be rejected when it is delivered, and the deliverer will not go unpunished either. One of the biggest reasons bad news is not accepted well is because it often goes against what a person believes to be true—and it makes that person feel foolish for believing it. Such events can, obviously, increase decision latency.

5. Optimizing business value. This is a structured means of improving business effectiveness and optimizing scope; it includes management techniques such as value engineering and value analysis. It will also decrease decision latency. The Standish Group suggests that your organization consider how the project will change the business—perhaps by increasing sales, reducing cost, or improving process efficiency. Optimize the project based on what business values have the greatest benefits.

It is rare that we can present projects that are truly identical, but the U.S. federal government has provided 50 of them in its Statewide Automated Child Welfare Information Systems (SACWIS) project. The state of Florida, for example, took eight years to finish up its work on this project. Their original cost estimate was \$32 million—but the final cost came to more than \$230 million. The state had a 109-person project team.

In contrast, the state of Minnesota worked on their SACWIS project for one year with a team of eight people. They concentrated on creating a standard infrastructure for collecting data and relied heavily on user involvement to minimize requirements and focus on the essential ones. Phase One was completed in seven months and Phase Two was completed in five months. The state spent \$1.1 million.

In short, the State of Florida created a project with a much larger scope and spent 230 percent more than the State of Minnesota, which optimized their project on its business value.

Complexity also increases decision latency. Try to keep it simple.

that person feel foolish for improving business effectiveness and value engineering and value analysis. What your organization consider how the during cost, or improving process efficiency. benefits.

but the U.S. federal automated Child Welfare Florida, for example, took eight estimate was \$32 million. The state had a

A WIS project for one year creating a standard infrastructure went to minimize requirements completed in seven months and spent \$1.1 million.

much larger scope and spent optimized their project on its

SKILLS OF THE FACTORS OF SUCCESS: SAME SKILLS

SAME (Standard Architectural Management Environment) is a consistent group of integrated practices, services, and products for developing, implementing, and operating software applications. SAME applies architecture principles to guide organizations through software technology transformations. It includes the communication platform, the vocabulary, the social media involved, and the necessity of keeping everyone on the same page. All these things reduce decision latency, because all the decisions on what, how, when, where, who, and why are premade.

We have identified five skills that go along with proficiency at SAME.

1. Enterprise architecture. Enterprise architecture covers the business, the organization, the information involved, and the technology that distributes that information. IT architecture is a coherent and consistent set of principles and standards that provide normative guidance for the design of an information system. An efficient organization has not only a functional enterprise architecture but also an individual or a team of people who can educate others about its use and function. Enterprise Architecture covers who, what, where, when, why and how relationships between them, and operating principles for a complete organization, a line of business, or a solution from several viewpoints. A well-thought-out enterprise architecture can reduce decision latency, because it reduces the number of decisions that need to be made during an active project.

2. The communication platform. This is a standard and well-known method of communicating with stakeholders and the project team, allowing tracking of issues and decisions. Standard methods include email, text messages, bulletin boards, and burndown charts. Social media works very well as a communication platform. The Standish Group suggests that your project teams create a common and easy-to-use communication platform and verify that effective communication is in place. Make sure the process is a closed-loop system to improve visibility—nothing turns people off more than not getting messages returned or answered. Use a mix of media and methods that highlight urgency when it is needed; do not overuse the message of urgency. Set goals and benchmark against them to track and improve responses to questions and decisions.

A well-thought-out enterprise architecture can reduce decision latency.

3. Vocabulary. Having a common vocabulary of words and phrases within an organization promotes good communications and breaks down the barrier of "IT speak." Good communication encourages low decision latency. Having a common vocabulary also builds transparency and accountability into the process and the program. A common vocabulary provides objectivity, making for less ambiguity in the results. It also speaks to goals held in common and is mindful of the reasons the organization is doing a particular project or program.

4. Social media. It is important that an organization learn to use social media products and concepts to manage, collaborate, and communicate project information and activities. Social media platforms include Facebook, Twitter, and LinkedIn. Microblogs, podcasts, YouTube videos, and wikis are useful for managing, collaborating, and communicating project information and activities. Adopting social media techniques and tools can provide for more effective communication. How savvy is your organization at using social media to communicate?

David Saul, Senior Vice President and Chief Scientist at State Street Services, suggests project sponsors be visible to the project community by creating a social network. "I think it's very important for the executive to be visible on social networking communities. By having the right collaboration technology, you can be a lot more effective and get to a lot more people more quickly. With the use of collaboration technology, I have already seen that people and different groups can share best practices as well as ask questions."

5. Being on the same page. Project stakeholders and team members should be looking at the same project goals, requirements, time lines, costs, gains, risks, and commitments in the same way—with a common understanding of the central business objectives of the project. "Being on the same page" also implies agreement among the stakeholders about the project's purpose and direction, and reduces decision latency. How skilled is your organization at making sure that everyone has the same understanding of a project's business objectives?

A common vocabulary provides objectivity, making for less ambiguity in the results.

SKILLS OF THE FACTORS OF SUCCESS: PM/EXECUTION SKILLS

PM/Execution skills are the skills involved in managing software projects—skills that help you remove things that get in the way of making rapid decisions.

We have identified five skills that go along with PM/Execution Skills.

1. Count only what counts. Select just three to five measurements that matter, and then track them over time. So much of what is measured and tracked within the project management process has marginal value and causes increases in cost and time. These kinds of distractions reduce the project's overall value and increase decision latency. In fact, many organizations track hundreds of measurements and produce countless charts and reports, which no one really reads or understands. Counting only what counts is a good argument for breaking larger projects down into "micro-projects."

2. Do only what matters. This is the corollary to "Count only what counts." General project/portfolio tools and processes are a collection of integrated applications that aim to provide visibility into planning and progress in order to make timely and informed decisions. However, many of these tools and processes actually just get in the way of making timely decisions and only increase decision latency. Rank items by value in reducing decision and drop things that do not or offer marginal value.

3. Negotiation. Negotiation is the act of mutual discussion and arrangement of the size, scope, resource allocation, and commitment to the direction in which a project is heading—a basic understanding of what will happen when, by whom, why, and how. Negotiations are carried out with sponsors, users, and technical staff, and good negotiations lead to shorter decision latency. Remember that every project must have a business objective, whether that is increasing revenue, improving customer satisfaction, cutting costs, or improving control. The scope—the functions affected as well as the constraints, such as deadlines, budget ceilings, and other assumptions—must therefore be defined in business terms. Successful projects have stakeholders who know how to negotiate and know how to compromise.

How good is your organization at teaching and encouraging the art of negotiation?

A common vocabulary provides objectivity, making for less ambiguity in the results.

Be visible on social networking sites a lot more effective and get to a lot more I have already seen that people and

members should be looking at the same commitments in the same way—with a project. Being on the same page" also improves and direction, and reduces decision fatigue for the same understanding of a

4. Dealing with a project saboteur. A project saboteur is a person who does not want the project to succeed and takes action or refrains from action to sabotage the project. In some cases a project saboteur will be outspoken, but more often he or she will use passive-aggressive techniques to disrupt the project. A project saboteur will clearly increase decision latency. You can improve your skills at discovering and dealing with a project saboteur by reading *The Project Saboteur* (Claret Press, May 2016) by Dion Kotteman and Jeroen Gietema.

5. Celebrate accomplishments. Celebrating accomplishments is a means of positive reinforcement. Celebrating accomplishments rewards the accomplishment and reinforces proper behavior that will advance the project's progress; it also encourages the team to focus on execution, thus reducing decision latency. In short: Never take your team or stakeholders for granted.

One the easiest and most effective methods of celebrating is through recognition of an accomplishment. The project sponsor or leadership that says thanks for a job well done or for an extra effort will find a great payoff. Recognition from the company, such as naming a person of the month or designating a special parking place for a week or a month, is also a small token of appreciation. An organization that understands and recognizes true accomplishments and concrete results will earn the respect and admiration of the team.

Rewards can come in many forms: a special lunch, an added bonus, an extra day off, or pizza and beer on Friday afternoons, to name a few. Rewards can be individual or go to the whole group. After a major event or at the end of the project, the executive sponsor should hold an event, like a party or dinner out. This could also include a Broadway show or a boat ride that includes spouses; a picnic in the park; or a baseball game. This is a great equalizer, where all the team members get to celebrate success and no one is ignored. An event provides a platform for encouraging greater performance and team harmony.

**A project saboteur
will clearly increase
decision latency.**

a partner who does not want the project to succeed. In some cases a project manager will use negative techniques to disrupt the project. Even your skills at discovering problems can be used against you. (Project Management Press, May 2016) by Dion Hinchliffe

A project saboteur will clearly increase decision latency.

Recognition from the company, taking place for a week or a month, is also a true accomplishment and

an extra day off, or pizza and beer on to the whole group. After a major event like a party or dinner out. This spares a picnic in the park; or a baseball game to celebrate success and no one is ignored. and team harmony.

DECISION LATENCY OVER THE LIFE OF A PROJECT

In his book *Thinking Fast and Slow*, Daniel Kahneman points out that there are two systems working within our minds when we make decisions. System one makes decisions very quickly based on intuition, emotions, and experiences. System two makes decisions slowly, after weighing all the evidence and possible outcomes. Martin Gladwell, in his book *Blink*, suggests that quick decisions based on intuition can be more accurate than well-researched and careful analysis. On the other hand, James Surowiecki suggests in his book *The Wisdom of Crowds* that a group of people often makes better decisions than single "experts." If we could intersect these three decision methods, perhaps we could improve the quality of our decisions. However, what we're really looking for is the shortest possible interval in which a good decision can be made: in other words, short decision latency.

Perhaps there's another way to think about this issue.

In recent years, stages or gates have become popular in developing software. Some of these systems have five gates, while others have six. However, we prefer to think of them as three periods: pre-project, active project, and post-project. In the pre-project period we gather evidence, specifications, business justifications, resource allocation, and information on user needs. During this period, we can make slow decisions. We can do some crowd-sourcing. We can think big, small or wild. We can make sure the teams are ready to do battle by making sure they have the right training and platforms. We can try to optimize projects into microprojects or micro-services.

In the active-project period, though, we want fast thinkers. We want our product owners and sponsors to set priorities, give the team feedback, and inspire the team to greatness by reducing decision latency. We want our teams to make decisions rapidly and decisively. In this period, we don't want change management to slow us down; we want to embrace change. We want the team to have the freedom to be successful.

In the post-project period, we need even faster thinkers. Why? Because the product is now in the hands of users and stakeholders, so critical changes must be implemented quickly.

SKILLS OF THE FACTORS OF SUCCESS: SUMMING UP

The 50 skills we've outlined share one attribute—besides improving decision latency—and that is that they can all be implemented at very little cost. Many of these skills can be improved through individual and small-group workshops and exercises. The 50 skills do not require new tools; in fact, they call for reducing the use of tools and reducing costs in that area. They focus less on control and more on delivering the product. Our research tells us that organizations that heavily control their projects with the intense use of tools and processes suffer from long decision latency.

The 50 skills of the Factors of Success not only improve latency but can also be implemented at very little cost.

Let's sum up the findings we've picked out in this book. First, we know that the agile process helps reduce decision latency, while non-agile and waterfall projects hurt decision latency by increasing the interval between problem and resolution. Second: a skilled project sponsor will decrease the interval by distributing the decision process among a few trusted individuals, but a poorly skilled sponsor will only increase that interval. Third: small projects involve fewer decisions and fewer people to involve in those decisions. Large projects increase decision-making intervals by requiring more decisions and longer paths to them. Fourth: talented and emotionally mature teams make rapid and firm decisions. Fifth: industries such as retail make rapid and firm decisions; more bureaucratic organizations (think governments) make slow and shaky decisions.

Now that you've ranked your organization on its proficiency at each of the skills we've outlined above, you're ready to fill out our Project Environmental Benchmark, which is available on our website. The assessment weights each of the skills by their importance in reducing decision latency and improving project performance. When you've finished our assessment, you'll know the top three skills that your organization should work on to improve your organization's decision-latency skills. The result will also give you a benchmark score so you'll know how your organization stacks up against other organizations. You can obtain access to the assessment and the benchmark on our website.

SUMMING UP

reduced latency—and that is that they represent through individual and small teams in fact, they call for reducing the use and focus on delivering the product. Our study on the intense use of tools and

in the book. First, we know that the agile and waterfall projects hurt decision problem and resolution. Second: a model by distributing the decision process global sponsor will only increase that decision and fewer people to involve in decision-making intervals by requiring more talented and emotionally mature teams such as retail make rapid and firm decisions (governments) make slow and shaky

on its proficiency at each of the skills Digital Benchmark, which is available on our site in reducing decision latency and improving you'll know the top three skills that your latency skills. The result will also give you up against other organizations. You can

FACTORS OF SUCCESS: THE SIZE-COMPLEXITY MATRIX

A few years ago, we created the Size-Complexity Matrix as a way to determine the estimated likelihood of success based on both a rating system and a color code. This matrix is based on more than 120,000 projects collected over 20 years. Green means the project has a good chance of success; yellow means the project will most likely be challenged; and red means the project has a very good chance of failure. The Size-Complexity Matrix provides guidelines for categorizing a project in order to assess risk and effort. It uses a 5-point scale for both size and complexity. The lowest-point project is simple and small and would score 100 points. The largest and most complex project would score at 1,000 points. Green means low risk and effort; yellow means medium risk and effort; and red implies high risk and effort.

The Size-Complexity Matrix provides a rough estimate of risk. A Single Project Appraisal is also an easy way to predict and assess the critical elements of an individual project to improve its chances of success. This appraisal lets you compare your project with 50,000 other projects, as well as the results from 1,000 other organizations that carried out a similar project.

A Single Project Appraisal includes these estimates:

- Chances of success
- Time and cost overrun percentages
- Success factors and best practices
- Suggestions to overcome stress factors
- Value optimization

Table 24: Size-Complexity Matrix

| | | C O M P L E X I T Y | | | | |
|------------------|----------|---------------------|--------|--------------------|---------|--------------|
| | | Very Simple | Simple | Average Complexity | Complex | Very Complex |
| S I Z E | Small | 100 | 250 | 400 | 550 | 625 |
| | Moderate | 175 | 325 | 475 | 625 | 775 |
| | Medium | 250 | 400 | 550 | 700 | 850 |
| | Large | 325 | 475 | 625 | 775 | 925 |
| | Grand | 400 | 550 | 700 | 850 | 1000 |

FACTORS OF SUCCESS: THE SIZE-COMPLEXITY MATRIX

It is easy to create your own Size-Complexity Matrix estimate using the following tables and guidelines. The size is selected by using Size Guidelines Table. (Standish uses labor effort as a major ingredient to measure size; therefore, when selecting the project size in Table 25, use normal US labor rates as your guide.) Remember these are guidelines, not rules.

To use the Complexity Guidelines Table 26, assign points and add them up based on the attributes of the project. The higher the points, the more complex the project. We use two dimensions to assess complexity: environment and scope. If none of the table's attributes apply (i.e., diverse user base, multiple team locations, and the like), then your project is clearly very simple. If you score fewer than 6 points, your project is simple. If you score 7 to 10 points, you have an average project; over 10 points indicate a complex project, and anything above 12 points indicates a very complex project.

There are a couple of ways you should use this matrix. First, determine the project forecast in terms of size and complexity. Then, think of it in terms of your project experience as a role model. We had the benefit of 100,000 detailed projects to draw on as our role models.

Table 25: Size Guidelines

| Size Description | Size |
|--|----------|
| Under \$1 million labor or less than 10,000 hours of productive labor | Small |
| \$1 million to \$3 million or 10,000 to 30,000 hours | Moderate |
| \$3 million to \$6 million or 30,000 to 60,000 hours | Medium |
| \$6 million to \$10 million or 60,000 to 100,000 hours of productive labor | Large |
| Over \$10 Million or over 100,000 hours | Grand |

Table 26: Complexity Guidelines

| Environment/Scope | Points |
|-------------------------------------|--------|
| Diverse User Base | 1 |
| Multiple Team Locations | 1 |
| Multiple Stakeholder Locations | 1 |
| Uncooperative Peers | 2 |
| Uncooperative Stakeholders | 3 |
| Many Requirements - Large scope | 1 |
| Ambiguous Basic scope | 1 |
| Fuzzy Undefined Requirements | 1 |
| Diverse and Multifaceted Objectives | 2 |
| Breaking New Ground | 3 |

COMPLEXITY MATRIX

the following tables and guidelines.
labor effort as a major ingredient to
(use normal US labor rates as your guide.)

them up based on the attributes of the
two dimensions to assess complexity:
diverse user base, multiple team locations,
than 6 points, your project is simple.
indicate a complex project, and

the project forecast in terms of size
as a role model. We had the benefit of

Table 26: Complexity Guidelines

| Environment/Scope | Points |
|-------------------------------------|--------|
| Diverse User Base | 1 |
| Multiple Team Locations | 1 |
| Multiple Stakeholder Locations | 1 |
| Uncooperative Peers | 2 |
| Uncooperative Stakeholders | 3 |
| Many Requirements - Large scope | 1 |
| Ambiguous Basic scope | 1 |
| Fuzzy Undefined Requirements | 1 |
| Diverse and Multifaceted Objectives | 2 |
| Breaking New Ground | 3 |

FACTORS OF SUCCESS: MEANING OF CHAOS

CHAOS—the Comprehensive Human Appraisal for Originating Software—is all about the human factor. For almost a quarter of a century we have been looking at why some software projects succeed and other projects fail. We collected more than 300 common reasons. We catalogued them and then broke them down into 100 points and 3 elements each, and then spread them over the 10 Factors of Success. These 300 common reasons point to decision latency as the root cause of why project fail or succeed. They explain why some projects are valuable and others have no value. They point to why some projects delight their customers, while others have the opposite effect.

It is very human to put off making decisions. Many people just do not like to make them. Software is built by humans to be used by or for humans. This is why CHAOS's study of human behavior in the development and execution of software projects is so important. Decision Latency Theory states, "The value of the interval is greater than the quality of the decision." Therefore, we have to continually remind humans that decisions need to be made. That's why you should decide to get your boss and co-workers to buy a copy of this book to help them decide to do the same.



BENCHMARK YOUR SOFTWARE PROJECT PERFORMANCE

Benchmarking involves measuring the performance of your projects and project skills against our CHAOS Database of 1,000 organizations and 50,000 project profiles. Comparing your projects and project environment to others is a valuable way of improving your understanding of your project performance and potential improvement. Benchmarking your project environment allows you to discover what areas need improvements and helps you to develop a plan toward achieving those improvements.

Benchmarking is used as a tool to help your business evaluate opportunities for improvement, such as:

- Identifying and prioritizing specific areas of improvement
- Understanding your project needs better
- Comparing your organization to others
- Identifying your strengths and weaknesses
- Setting goals and performance expectations
- Monitoring your performance and effectively managing change
- Understanding your project environment

The Standish Group offers its customers several benchmark services. One of our most popular benchmark services helps you look into a future project, assessing its likelihood of success and how to improve the chances the project will be successful. For some recent projects, we have improved the chances of success from 20% to better than 80% through finding the right combination of success factors. Another popular benchmark is our Value Portfolio Optimization and Management Service. This is a forward-thinking and predictive visualization of the value of your software investments. By focusing your project portfolio on value, our service frees your organization to create value.

These two benchmark services are custom-made for the specific organization. They are premium services that require onsite Standish staff as well as a commitment of staff from the organization to be benchmarked.

PERFORMANCE

and project skills against our
Comparing your projects and
understanding of your project
environment allows you to discover
achieve those improvements.
opportunities for improvement, such as:

One of our most popular benchmark
and how to improve the
improved the chances of success
factors. Another popular
This is a forward-thinking and
your project portfolio on
organisation. They are premium services
the organization to be benchmarked.

The Standish Group also offers self-service benchmarks. These have five basic attributes: 1) they can be done online; 2) questions are closed-ended and multiple-choice; 3) they can be done quickly; 4) they can be done by an individual, with no need for consultants; and 5) they are inexpensive. On completion of each benchmark, you will receive a formal report. All reports contain your score against our standard, your comparisons to other organizations/individuals, and three suggestions on how to improve your score and benchmark.

The Standish Group offers four self-service benchmarks:

- The Good Mate Benchmark
- The Good Sponsor Benchmark
- The Project Environmental Benchmark
- The Project Resolution Benchmark

The Good Mate Benchmark—a companion to our book *The Good Mate*—is for an individual who wants to improve his or her team relationship skills. Both the book and the benchmark cover 10 principles and 50 skills we have identified to improve teammate relationships. Upon completion of your Good Mate benchmark, you will receive a formal report. This report contains with three sections: your score against our standard; you in comparison to other teammates; and three suggestions on how to improve your score and benchmark.

The Good Sponsor Benchmark is for an individual who wants to be a successful project sponsor. The benchmark is a companion to our book *The Good Sponsor*. Both the book and the benchmark cover 10 principles and 50 skills we have identified to improve a sponsor's chances of having a successful project. On completion of your Good Sponsor benchmark, you'll receive a formal report with three sections: your score against our standard; your comparisons to other sponsors; and three suggestions on how to improve your score and benchmark.

Our Project Environmental Benchmark is aimed at organizations that want to improve their project environment to allow for greater project value. This benchmark is a companion to our "CHAOS Report: Decision Latency Theory: It's All About the Interval." Both the book and the benchmark cover 10 principles and 50 skills we have identified to improve project value. Upon completion of your Project Environmental benchmark, you will receive a formal report. The report contains three sections: your score against our standard; your comparisons to other organizations; and three suggestions on how to improve your score and benchmark.

Progress

Iterative Process

Iterative is basically trial-and-error software development. It provides for a constant cycle of design, development, test, user acceptance, and then back to design or redesign. The idea is that stakeholders get to see the system in small steppingstones before it is fully built, thereby allowing changes to occur without having a major impact on the project cost and schedule.

How skilled is your organization at executing an iterative process?

- VERY SKILLED
- SKILLED
- MODERATELY SKILLED
- POORLY SKILLED

[Save & Continue](#)

[Save & Exit](#)

Progress

Retrospective

A retrospective is a meeting held by a project team at the end of a project, process, or iteration to discuss what was successful about the project or time period covered by that retrospective, what could be improved, and how to incorporate the successes and improvements in future iterations or projects.

How skilled is your organization at executing retrospectives?

- VERY SKILLED
- SKILLED
- MODERATELY SKILLED
- POORLY SKILLED

[Save & Continue](#)

[Save & Exit](#)

Upon the purchase of a Good Mate, Good Sponsor, or Project Environmental Benchmark, you will be sent a link. This link will bring you to the 50 questions to be answered in order to complete the benchmark. You should be able to complete the benchmark in about 20 minutes. Once you complete the questions, you'll have access to your report, which you can download and save. You'll have taken the first step on your journey toward improvement.

I want to improve their project
comparison to our "CHAOS Report:
and the benchmark cover 10 principles
of your Project Environmental
your score against our
on how to improve your score

at the end of a
successful about the
what could be
and improvements in

a retrospective?

Save & Exit

Benchmark, you will be sent
to complete the benchmark.
Once you complete the questions,
you'll have taken the first step on

The Project Resolution Benchmark is for an organization that wants to improve its project delivery to allow for greater project output. The benchmark compares your projects against the 50,000 projects and 1,000 organizations in our CHAOS Database. The report contains four sections: your project results against similar projects in the CHAOS Database; your comparisons to other organizations; your score against our standard; and three suggestions on how to improve your project delivery. There are three steps in completing the resolution benchmark: the organizational profile; a skills appraisal; and the project profile.

The organizational profile involves a total of 12 questions, starting with the company name and ending with user feedback. This step should take about 10 minutes to complete.

Step 1 - Create Company

Fields with * are required.

Company *

Name

Industry *

Company Size *

Country *

Region *

PMO Oversight

EPPM Tool Use

Project Management Tool Use

Requirements Management Tool Use

Change Management Tool Use

EA Compliance

Decision Speed

The skills appraisal covers the same factors for success and skill questions found in the Project Environmental Benchmark. These 50 questions are also in our "CHAOS Report: Decision Latency Theory: It's All About the Interval" book. However, here the questions are used to provide a higher level of recommendations for improvement, and draw on our reasons for project success or failure to arrive at customized recommendations.

Progress

Decision Pipeline

A decision pipeline is a method to make decisions quickly and easily by the right people at the right time. It is also a method to engage more people in the decision process. The Standish Group Dezider.com is an example of a decision pipeline.

How skilled is your organization at making fast decisions through a decision pipeline?

- VERY SKILLED
- SKILLED
- MODERATELY SKILLED
- POORLY SKILLED

Save & Continue

Save & Exit

Progress

Imagination

Imagination is the ability to form new ideas and concepts of objects and services not yet realized. It is the creative power to innovate, invent, and originate. It is the ability to envision how things can be versus how they are today. It is the ability to see the future. It is the ability to daydream.

How skilled are your sponsors at imagination and daydreaming?

- VERY SKILLED
- SKILLED
- MODERATELY SKILLED
- POORLY SKILLED

Save & Continue

Save & Exit

found in the Project
AMCI's Report Decision Latency
are used to provide a higher
for project success or failure

of objects and
to innovate, invent,
things can be
versus how
Perhaps it is the ability to

and learning?

Start Test

The project profile is based on 14 questions that are answered using drop-down boxes and are closed-end. In order to make a statistical relevantly benchmark, you'll need to describe a minimum of 25 projects and a maximum of 35 projects. It will take you about 10 minutes to complete each profile, or a total of 4 to 6 hours. Including the first two steps, you should be able to complete our benchmark in one day.

It's best to fill out these benchmarks with a team of 3 to 4 people. That means your investment of time is less than one "person" week.

The screenshot shows a web-based form for creating a project profile. The form is divided into two main sections: 'Project Number' and 'Goal'.

Project Number:

- Name:
- Generic Name:
- Size:
- Type:
- Methodology:
- Value:

Goal:

- Goal:
- Capability:
- Complexity:
- Satisfaction:
- Did you complete the project and is it being used?
Completed:
- On Time:
- On Budget:
- On Target:

Buttons:

- Save Project #1

Note: We can customize your Project Resolution Benchmark with more projects and with multiple departments or divisions, and we offer telephone and onsite support. For more information on our benchmarks, go to our website: www.standishgroup.com

JIM JOHNSON is a professor at the Antwerp Management Schools and Chief Dreamer of The Standish Group. He has been professionally involved in the computer industry for over 50 years and has a long list of published books, papers, articles, and speeches. 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 recognition and predictions of emerging technology trends. Jim is a pioneer of modern research techniques and continues to advance study methodologies in the research industry through case-based analytical technology.



Other books by Jim Johnson include:

The Good Mate

The Good Sponsor

The Dead Presidents' Guide to Project Management

My Life is Failure

The Public Execution of Miss Scarlet

Executive Sponsor Research Report

Emotional Maturity Research Report

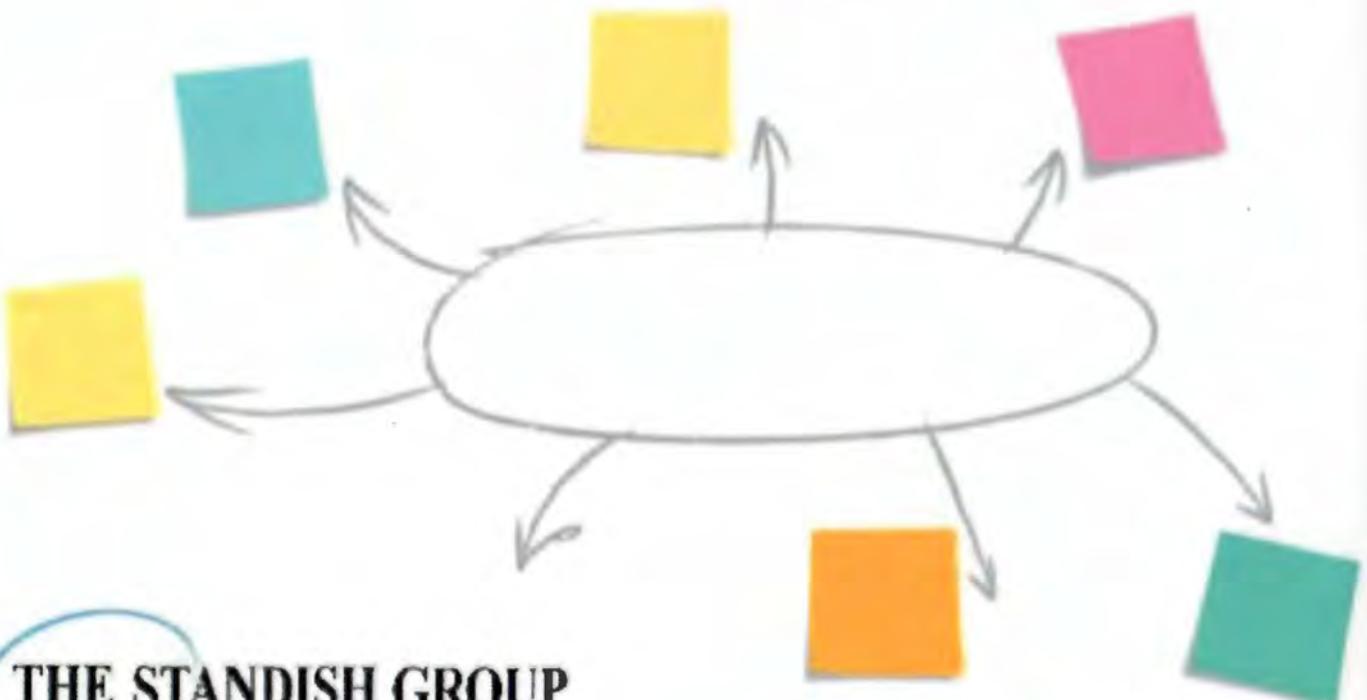
20 years of CHAOS Reports

Children's Books

Dragon2Dragon

Saving Jimmy: Adventures of Jackie the Squirrel

Jackie and the Three Bears



THE STANDISH GROUP

The Standish Group is a primary research advisory organization that focuses on software project performance. Using our extensive primary research, you can improve your investments in software projects. We are a group of highly dedicated professionals with years of practical experience helping organizations improve: innovation, value, and predictability. The Standish Group was formed in 1985.

with a vision of innovating group refection using case-based reasoning techniques. We do this in order to profile your projects and environments against thousands of cases to deliver more precise advice based on collective wisdom. For over 30 years The Standish Group has been researching and providing advice on how to increase the value of software investments.

ISBN 978-0-692-04830-6
9 0000 >



9 780692 048306