



Executive Brief
INTERTECH



The Pillars of Software Development Success

How to Deliver Software On-Time, On-Budget, On-Feature.

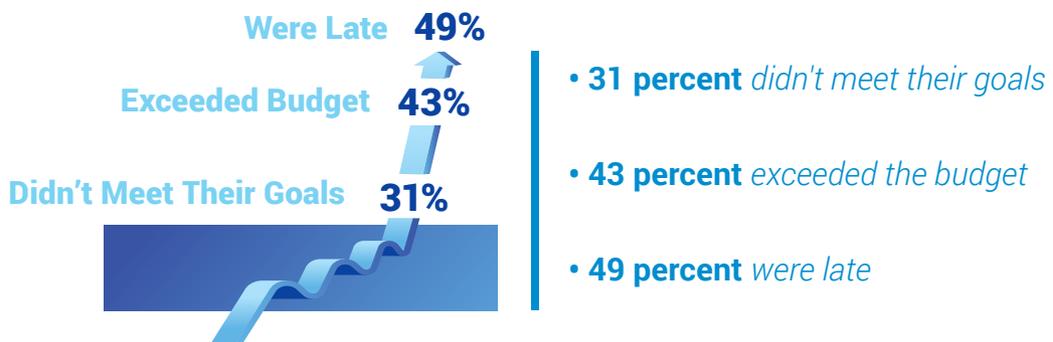
The Pillars of Software Development Success

How to Deliver Software
On-Time, On-Budget, On-Feature



Overview

A study by PricewaterhouseCoopers, which reviewed 10,640 projects from 200 companies in 30 countries, found that only 2.5 percent of the companies successfully completed 100 percent of their projects. Another more recent study was slightly more upbeat, but far from great news. We're referring to a 2017 report by the Project Management Institute (PMI), which found that *14 percent of IT projects fail and of the projects that didn't fail outright:*



The PMI report was particularly insightful because it

uncovered the nine top reasons projects fail, including:

1. Inaccurate requirements
2. Uninvolved project sponsors
3. Shifting project objectives
4. Inaccurate estimates
5. Unexpected risks
6. Dependency delays
7. Not enough resources (people)
8. Poor project management
9. Team member procrastination

Intertech works with hundreds of companies every year on a wide range of software development projects. We've learned a lot about how to avoid these predictable project roadblocks. This Intertech Executive Brief shares our best thinking, and the thinking of recognized software development experts, to help you avoid these issues – as well as “bonus success tips” for your next software development project.

Avoiding Inaccurate Requirements

“A problem well-stated is half solved.” That wise observation often is credited to inventor Charles Kettering, who once famously led the research function at General Motors. Succinctly stating the problem to be solved is equally critical in software development.

Preventable Failures

Unfortunately, many software projects fail at this crucial initial stage. ***In fact, poorly defined requirements cause 39 percent of software development project failures, according to the Project Management Institute.***¹ There are countless reasons for poor project requirements, and many are preventable. For example, development teams are sometimes shut out of the requirements development phase. Resist this. Without their input, the “why” of a project may not align with the attainable goals. Make a strong case for the development teams’ involvement from the very beginning.

Executive Insight

Requiring things like being cloud- platform or data- base-agnostic is expensive. ***You typically can get to about 80 percent of that goal for 20 percent of the costs.*** Likewise, blind commitment to a technology path, regardless of whether or not it is right for the project, is another mistake to watch out for during the requirements phase.



GOALS vs. COSTS

Recommendation ►

Recommendation

Don't make assumptions about what the end-user needs. Instead, talk to them (or an appropriate proxy) early on. Nothing is worse than finding out – too late – that what you have spent time and considerable effort developing is “not right.” Likewise, resist making project changes based on a particular team member's (or your own) personal preference or skill sets, versus what is best for the project or organization.

Uninvolved Project Sponsors

“Communicate project status—good or bad—early and often. Management and project sponsors need to know how things are going throughout the entire project.” When asked about the best way to make sure project sponsors are involved, our team responded that systematic communication is key. Ideally, that communication should happen directly with project sponsors on a frequent and regular basis. A mistake is involving someone with little project commitment or understanding of the business needs as a “go between.” This introduces delays and increases the possibility of misunderstandings.

A Single Product Owner

Multiple agendas, which means sponsors and teammates are not in sync, is a sure way to derail a project. Ideally, the project should have a single owner who makes final decisions about functionality questions, budget constraints, schedule issues, and anything else of consequence.

Executive Insight

Sometimes a customer representative or “product owner” can effectively bridge the gap between a busy project sponsor and the development team. Ideally, product owners are decisive and empowered to make

Executive Insights (continued)

decisions on behalf of sponsors. In a best case scenario, they can supply project strategy and direction, and product expertise. They play a vital role in ensuring the project goals are aligned with – and support – the overall business objectives.



Don't let micro-management doom your project

Recommendation

Uninvolved or overly engaged project sponsors can doom a project. Don't let it happen by proactively providing project updates so anxious sponsors are not tempted to micro-manage every technical detail. And by providing regular updates, sponsors will pay attention and communicate their enthusiastic project ownership to others throughout the organization.

Shifting Project Objectives

George Bernard Shaw once said, “Those who cannot change their minds cannot change anything.” Change is unsettling, but it also can be positive when better ideas hold sway. Project teams sometimes bristle when project objectives shift because past work might be wasted. But with agile there’s a middle way that allows change to happen without starting over from scratch. This provides a regular opportunity to weigh the long-term savings from change against the short-term benefit of sticking with the status quo.

Manage Changing Objectives

It’s typical for project owners or sponsors to weigh in with new ideas or wholesale changes. Budget changes also can trigger a shift in a project’s objectives. Fortunately, teams using agile have a proven method for managing these changes through product backlogs, daily meetings, frequent communication and sprint deadlines.

Executive Insight

No matter how often requirements or project objectives change, one rule should be iron-clad: the very end of a project is not the time to solicit feedback or to change project objectives. This is the time to nail the existing requirements and deliver the job.

Recommendation 



Recommendation

Encourage direct conversations between the IT project team and business representatives as much as possible. This helps build goodwill and makes surprise changes less likely by keeping everyone on the same page. But be prepared. The only constant in life – and in software development – is change. It's also wise to identify someone on your team with good communication skills who knows conflict management techniques. Empower that person to engage her skills when inevitable change requests arise.

Inaccurate Estimates

Inaccurate estimates – often due to ignoring infrastructure costs – are expensive. Research reveals that one in six of all software development projects overrun estimated costs by a whopping 200 percent.

For more information on **Software Estimation** and this statistic, [click here](#) 

Get Real

Avoid project overruns by starting with a realistic budget, which means including infrastructure expenses. The most commonly overlooked costs include:

- Setup/Configuration of Source Code Repository
- Setup/Configuration of CI server
- Artifact Management server
- Acquisition of Test servers
- VMs or physical server for each
- Permissions for each of these

Executive Insight

Seasoned team members capable of having unemotional discussions about features versus budget also are key to developing realistic estimates. These team members should engage in candid conversations with project sponsors and

Recommendation 

Executive Insights (continued)

owners during the estimation process. Using an agile development approach can help manage the problem too. **Agile experts Mark C. Layton and Steven J. Ostermiller have written:**

“A big benefit of (agile) projects is the capability to have a self-funding project. Scrum teams deliver working functionality at the end of each sprint and make that functionality available to the marketplace at the end of each release cycle. If your product is an income-producing product, you could use revenue from early releases to help fund the rest of your project.”²



Agile Development May Bring Added Benefits To Your Project

Recommendation

See the Intertech Executive Brief, ***A CIO's Guide to Avoiding Cost Estimation Errors in Software Development***, for a thorough overview of the best ways to keep your projects on feature and on budget.

Unexpected Risks ►

Unexpected Risks

*"In a 2015 study of 10,000 software projects by the Standish Group, small agile projects were found to be 30 percent more likely to succeed than traditional projects. Medium-sized projects were found to be four times (400 percent) more likely to succeed using agile over traditional approaches, and large, complex projects were found to be six times (600 percent) more likely to succeed with an agile approach."*³

Reduce Unexpected Risks

Risk, like change, is an inevitable part of software development. By using agile principles you can dramatically reduce the chance of unexpected risks derailing your project. Some may remember the days of huge projects failing after the investment of similarly huge resources. With agile, those days are over.

Executive Insight

Agile experts Mark C. Layton and Steven J. Ostermiller advise remembering the following basic agile principles to help avoid unexpected risks:

The highest priority is to satisfy the customer through early and continuous delivery of valuable software.

Welcome changing requirements . . . Agile processes harness change for the customer's competitive advantage.

- **Deliver working software frequently**, from a couple of weeks to a couple of months, with a preference toward the shorter timescale.
- **Business people must work together** daily throughout the project.
- **Working software** is the primary measure of progress.



Have A Risk Plan In Place

Recommendation

Even when you're following agile principles to a "T," unexpected factors can arise that put your project at risk. Have a risk plan that defines what's likely to go wrong and what happens when it does. Here are two things to guard against in your plan: **(1)** delivering bad news long after it's too late to make a correction, and **(2)** being risk averse to the point of causing danger to the project or organization.

Dependency Delays

“If we don’t all row, the boat won’t go.”

Recognize Dependencies Early

Project dependencies cannot be underestimated. Integration with systems within an organization or an outside third-party organization – such as accessing a public API to collect weather forecasts or integrating with a credit card company to process payments – need to be recognized and built into the project planning (and budgeting) process.

Executive Insight

During the integration process it’s not uncommon to find defects or to learn that APIs or integration points don’t work as documented. While third parties or internal groups usually are willing to fix these problems, it can cause significant delays.



Recommendation

It is essential to factor in time and cost for internal and external system integration in order to avoid delays. Early on, create sandboxes to prove out integrations. For internal integration points, create communication channels between your team and the members of other projects where your system is integrating. If any of the systems your system is integrating with are in development, continuously monitor those projects’ development rates, bug counts, and other key trends. And finally, with the agile methodology, be sure to attend the demo of your project’s dependencies to understand how well all is working.

Not Enough Resources ►

Not Enough Resources

“There is a worldwide turnover rate of 10.9 percent in the IT industry, with the tech sector (software development only) showing the most volatility with 13.2 percent turnover.”⁴

Recognize Dependencies Early

“Resources” is a polite euphemism for “people.” With high turnover in the IT industry, and in the software sector in particular, finding and retaining great people is crucial to the success of any project. See our earlier Intertech Executive Brief, **“Finding and Retaining Stellar IT Employees,”** for a more comprehensive look at this important topic.

For more information on **Finding & Retaining Stellar Employees**, [click here](#) 

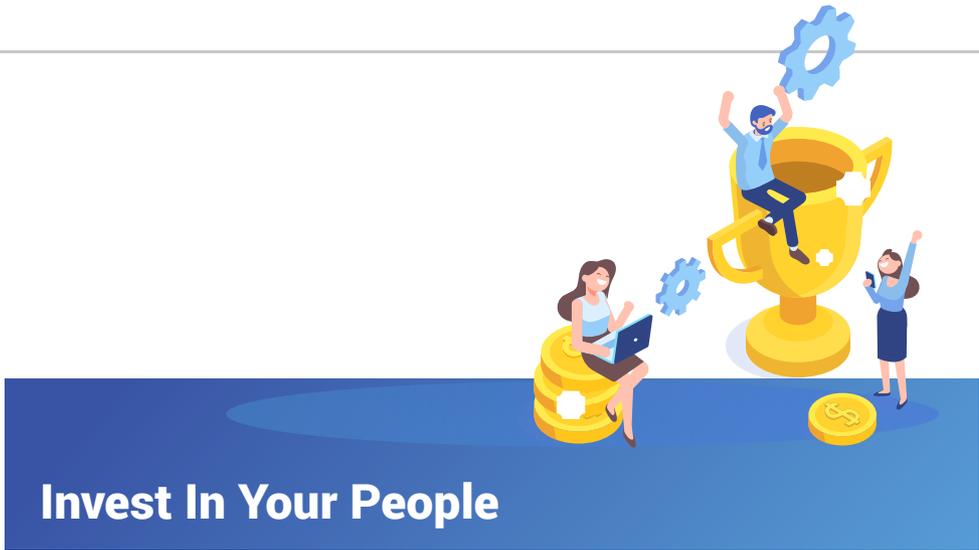
Executive Insight

Hiring consultants is an obvious way to fill gaps in your team, but make sure to go beyond basic staff supplementation. Good consultants are well versed in adapting to change and the challenges it implies. They understand the pros and cons of existing frameworks and the implications for capabilities and deliverables. In addition, many have deep agile and scrum experience, assuring that a well-constructed “augmentation” of your staff will further improve the results on your behalf.

*For more on hiring and working with consultants, check out the Intertech Executive Brief, **“What Every CIO Should Know: Hiring and Working with IT Consulting Firms.”***

For more information on **Hiring & Working With IT Consulting Firms**, [click here](#) 

Recommendation 



Recommendation

There are many things CIOs and other tech leaders can do to attract and retain great employees: from focusing on your people and their strengths, intentionally cultivating a values-based culture, and getting employees into alignment with your mission, to building team work, setting clear expectations and providing challenges and learning opportunities. Invest in your people and the investment will repay you with rewards many times over.

Poor Project Management

“Cost and time overruns have a profound effect on national economies. One estimate of IT failure rates is between 5 percent and 15 percent, which represents a loss of \$50 billion to \$150 billion per year in the United States.”⁵

Be Proactive

Not only do IT project failures cost our economy money, they cause your organization to lose money (and, sometimes, IT jobs are eliminated as a result). **A lot goes into proper project management, so consider this a 10,000-foot-view;** Our best advice from this lofty vantage point: take the time upfront to establish a software architecture, sound design principles, and good testing principles. Engage a senior, classic software architect to guide, participate in, and review the system design and implementation. And lastly, establish minimally viable product bounds for the level one release and push the non-mandatory to subsequent releases.

Executive Insight

Do not require perfection. Instead, carefully choose where you require the ideal. Look for opportunities that can be realized quickly and will start providing often overlooked return (*i.e., convincing uncertain managers, winning new business, securing existing customers*). It helps to use tools and infrastructure that enable fast turn-around and to keep the project

Executive Insights (continued)

in a known state, including IDE, build, CI, automated unit/integration/acceptance tests, and static analysis tools.



Establish An Agile Champion

Recommendation

If agile is new to an organization, do establish an agile champion at a high enough level to help remove older waterfall roadblocks. You can still succeed without this, but delivery will be slower and more expensive. Likewise, doing an agile/scrum project while requiring all of the older waterfall artifacts is like going on a diet and continuing to eat junk food in enormous quantities. Try to avoid the idea that your business is so “unique” that it needs its own version of scrum. Unless you have many years of experience to back up your decisions, it’s safer to follow an existing method firmly. Remember that these methodologies were refined through years of work and uncountable project volumes. And, as much as possible, help new people (especially those in authority) resist the temptation to make changes without appreciating and justifying the impact of those changes.

Take Time To Test ►

Take Time To Test

Testing is an essential aspect of good project management. Without a well thought out and executed testing plan, you are much more likely to fall short of end-user expectations. When this happens, you've missed the opportunity to build trust, gain market share and deliver as expected.



What Works:

Do require automated tests for large systems. There is a cost to unit testing, functional testing, and UI testing, but over time this will pay for itself.

Focus unit testing on the most complex areas of code or areas that typically have the most bugs.

What Doesn't Work

Requiring 100 percent code coverage by unit test is expensive and no guarantee of quality.

Testing software in the production environment is almost always a bad idea.

Team Member Procrastination ►

Team Member Procrastination

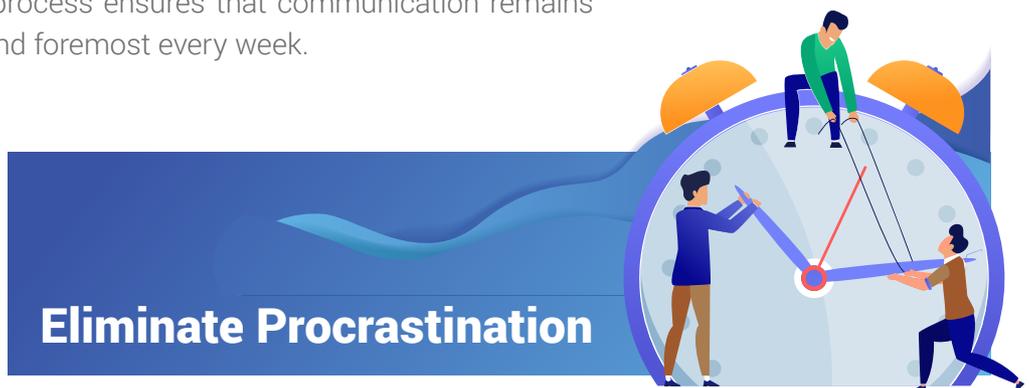
Pop psychology supplies many reasons for why people procrastinate, from fear of failure to paralyzing perfectionism. No matter the reason, procrastination is a project killer.

Shared Baseline Knowledge

Don't let it happen on your team. Begin by making sure everyone has shared baseline knowledge of the technology project path and – most importantly – are committed to the project's stated objectives.

Executive Insight

An aggressive timeline can be a great motivator, especially when combined with a good process and regular, clear, and open communication. Using an agile process ensures that communication remains first and foremost every week.



Eliminate Procrastination

Recommendation

Don't let conflicts fester. Unresolved conflict can cause people to start shutting down and deliver poor quality work (or not delivering at all). The best way we know to keep projects moving forward is through positive teamwork!

Teams That Work ►

Teams That Work

Foster effective teamwork and team management.



What Works

- | **Encouraging developers to prioritize company wins over team wins, and team wins over personal wins.**
- | **“Quickly praising jobs well done** and willingness to take responsibility for personal mistakes. Likewise, developers should be courageous enough to try things outside of their “comfort zone.”
- | **Honesty and willingness to quickly research something unknown.**
- | **Eating meals together,** which is a significant indicator of team cohesion.

What Doesn't Work

- | **Taking offense or judging others,** particularly when all the facts behind the actions are not understood.”
- | **Complaining or being unwilling to consider constructive feedback.**
- | **Building things in isolation.** And, the flip side of this coin: refusing to help each other or ‘chip in’ when necessary.



Effective Team Management

What Works

Team leaders understanding the current competencies of all team members and only requiring modest changes by gradually broadening or redirecting them, if necessary, over the course of the project.

Documenting processes so future team members can adapt quickly and easily.

Making sure the development team understands the business value of the software project. Engineers love to build technically beautiful things, but without proper understanding of the business needs, that beautiful software can completely miss the mark.

Ensuring that your team has at least one team member with leadership and coaching capabilities. A good leader is not always a good coach, but if you have someone who has both of these capabilities, consider elevating their responsibility and visibility within the overall project. Otherwise, identify staff members who are good coaches and consider pairing them with less experienced staff to elevate the team's overall capabilities.

What Doesn't Work

Project leaders that insist on team "shock (change) treatments" usually send their team reeling without confidence or a rudder. This makes it extremely hard for them to honor the agile principle of regular usable software deliverables.

Closing Thoughts



Closing Thoughts



Remember, there is no silver bullet. Success is the result of a series of things consistently done well.

Another parting thought: don't get trapped into thinking that software only can be delivered when it's 'done.' We've worked on many successful projects where a small piece of valuable functionality was delivered early, which helped build critical enthusiasm about the project and its future.

Six Factors That Lead To Success

For a project to be successful, six factors must be met:⁶

1. It's delivered on time.
2. Cost doesn't exceed budget.
3. It works as designed.
4. People use it.
5. The people who funded the project are happy with it.
6. It meets the goals that drove the project.

If you can check off all these markers, don't forget to celebrate and

Closing Thoughts (continued)

recognize team members when the project is finished. Whether it's a formal event or a beer out with the team, it matters. ■

About Intertech Executive Brief

Intertech publishes original articles, reports, and periodicals that provide insights for business leaders. Our goal is to draw upon research and experience from throughout our professional services organization, to include consulting and training, as well as coauthors in academia and business throughout the world, to advance the understanding of important principles of interest to executives.

About The Author



Tom Salonek is the founder and CEO of Intertech, a technology consulting and training firm. Intertech has won more than 50 awards for growth and innovation, including being named one of the Top 30 Places to Work in Tech by Fortune magazine. He has an undergraduate degree in Quantitative Methods from the University of St. Thomas, where he was also an instructor at the Graduate School of Business Management Center, and has completed executive education at the Harvard School of Business and MIT.

To find out more about how **Intertech Can Help You**
With Your Next Project, [click here](#)



Copyright © 2020 Intertech, Inc. All rights reserved.



Sources

- (1) PMI – Success Rates Rise – Transforming the high cost of low performance
- (2) Agile Project Management for dummies, 2nd edition
- (3) Agile for dummies, 2nd edition
- (4) LinkedIn 2017 member data
- (5) Gallup Business Journal – The Cost of Bad Project Management
- (6) PMI – Success Rates Rise – Transforming the high cost of low performance



Executive Brief
INTERTECH

651. 288. 7000

www.intertech.com