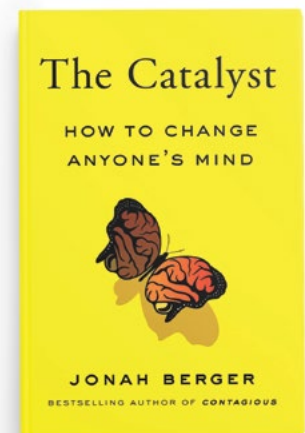


TOP 10 ORGANIZATIONAL CHANGE MANAGEMENT ANTI-PATTERNS FOR DIGITAL TRANSFORMATIONS

A WHITEPAPER BY GARY GRUVER

THE BIGGEST CHALLENGE in transforming how organizations develop and deliver software is getting people to invest in improvements and embrace new ways of working. In large organizations, it is hard to overcome the inertia of doing things the same old way. Most organizations have groups of forward-thinking people that understand what needs to change and why, but these change agents are often unsuccessful at influencing the larger organizations.

As someone committed to doing everything I can to help people succeed on their digital transformational journeys, I spend an increasing amount of time focused on these organizational change management issues. Recently, I came across the book *The Catalyst: How to Change Anyone's Mind*, by Jonah Berger, a professor at the Wharton School of Business, focused on how to influence change. What I realized after reading his recommendations is that most of what I see being attempted in software industry transformations is in direct opposition to what organizational change management experts, like Berger, recommend. Instead of executing on patterns that would set up organizations for success, the software industry is, instead, copying and executing common organizational change management “anti-patterns” — and therefore, are struggling to successfully master digital transformation initiatives.



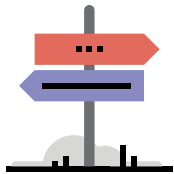
The goal of this white paper is to provide a brief review of the approaches recommended by Professor Berger, to highlight what I see as the major anti-patterns present in the software industry and propose some better options. My hope is to encourage more people to look outside the software industry for advice; to listen to experts like Berger and consider different approaches to influencing digital transformations that are more likely to succeed.

Berger points out that the biggest mistake change agents make is spending all of their time trying to convince people to change. Instead, they should first understand what is keeping them from embracing change and then remove those barriers. In *Catalyst* he covers five principles of resistance that hinder effective change: (1) Reactance, (2) Endowment, (3) Distance, (4) Uncertainty, and (5) Corroborating Evidence, and includes approaches for lowering the barriers to change for each principle. This whitepaper reviews each of these principles, and demonstrates how change agents can act as catalysts to reduce the barriers. I also will highlight the associated software anti-patterns and propose ideas for a new and different approach. All text in quotes I have paraphrased from the book to give readers a brief summary of his recommendations; I hope it encourages as many people as possible to read the entire book.

I. REACTANCE

“When pushed, people push back. Just like a missile defense system protects against incoming projectiles, people have an innate persuasion system. Radar that kicks in when they sense someone is trying to convince them.”

To avoid reactance and the “persuasion radar,” then, catalysts allow for agency. Stop trying to persuade people and instead get them to persuade themselves.



“One way to allow for agency is to **let people pick the path**. Let them choose how they get where you are hoping they will go.”



“Another route to self-persuasion is to **highlight a gap** – a disconnect between what they might recommend for others versus do themselves.”



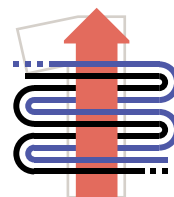
“Another way to allow for agency is to **ask questions rather than make statements**.” “Questions encourage listeners to commit to conclusions. To behave consistently with whatever answers they gave.”



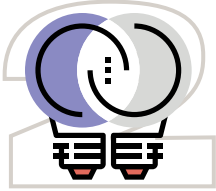
Start with understanding:
“Working to gain insight into where that person is coming from. Comprehending and appreciating the person’s situation, feelings, and motives, and showing them someone else understands.”

SOFTWARE ANTI-PATTERNS

Most of the problems related to reactance I experience in the software industry are just different approaches to forcing change on people based on what we think they should be doing.

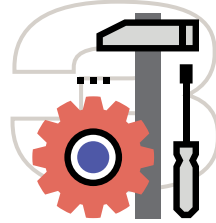


1. MATURITY MODELS: Maturity models assume there is one right path to improvements and it assumes everyone should follow that path. The maturity model then measures different groups against what you believe is the right path, and tries to force them to change so they can be seen as more “mature.”



2. COPYING BEST-IN-CLASS:

Assuming the right approach is about convincing everyone to copy the best-in-class companies instead of truly understanding their actual situation. Different applications, different architectures and different products have different challenges, and to assume that the right path for improvement is to copy organizations that are developing websites with modern technology is naïve, at best.



3. TOOLS-FOCUSED

TRANSFORMATION: Many organizations assign process improvement initiatives to the same teams that are responsible for tools. These organizations focus on picking the right tools for the enterprise, creating pipelines, and then measuring teams based on the adoption of these tools without ever worrying if they are solving problems, or if teams are embracing new ways of working and really changing.

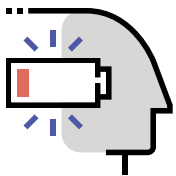
WHAT IF, INSTEAD...

We taught the people that are responsible for developing and delivering software how to improve their processes instead of having the tools teams trying to convince them to change? What if, instead, we gave each of the software teams a framework for highlighting opportunities for improvement specific to their applications and making those opportunities visible to everyone? What if, instead, we provide training that includes examples outside of our team's own respective organizations so that they can get used to making recommendations for different situations? This will help highlight the disconnect between what teams are recommending for others versus what they are doing in their own organizations. Teams can then be asked to make their processes visible and undertake an improvement project. Instead of being told what to do based on visibility of processes, take the time to really listen and understand what was chosen and why. Do we think letting people persuade themselves would help to avoid reactance and the instinctive activation of persuasion radar?

II. ENDOWMENT

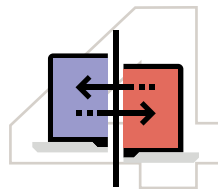
“As the old saying goes, ‘If it ain’t broke, don’t fix it.’ People are wedded to what they’re already doing. And unless what they’re doing is terrible, they don’t want to switch. To ease endowment, or people’s attachment to the status quo, catalysts highlight how inaction isn’t as costless as it seems.”

To reduce the endowment barrier, catalysts focus on the cost of inaction.



“Rather than focusing on how much better the new thing is than the old, or the potential gain of action, catalysts do the opposite. **They highlight how much people are losing by doing nothing.**”

SOFTWARE ANTI-PATTERNS



4. BENCHMARKING: Benchmarking against best-in-class organizations that often don’t have similar products to show what is possible.

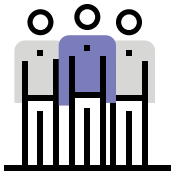
WHAT IF, INSTEAD...

We focused on an organization’s applications and processes, making visible the inefficiencies that are keeping them from achieving their objectives? Even better,, what if we teach them to use common sense approaches to highlight the costs of inaction themselves and require them to do an improvement project? This way, instead of using the gaps to convince them to change, they can use their own analysis to persuade themselves. Since we are not telling them what to improve, they can pick the improvements they think make sense from their analysis of the gaps. Do we think this would help lower the barriers of endowment?

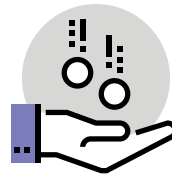
III. DISTANCE

“People have an innate anti-persuasion system, but even when we just provide information, sometimes it backfires. Why? Another barrier is distance. If new information is within people’s zone of acceptance, they’re willing to listen. But if it is too far away, in the region of rejection, everything flips. Communication is ignored or, even worse, increases opposition.”

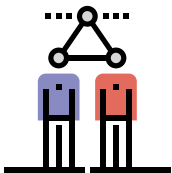
Catalysts avoid the region of rejection and encourage people to actually consider what they have to say by (1) finding the “movable middle,” (2) asking for less and (3) switching the field to find an “unsticking point.”



Focus on people that are more open to the ideas known as the **“movable middle”** — people who have a larger zone of acceptance or whose zone has greater overlap with the recommended change.

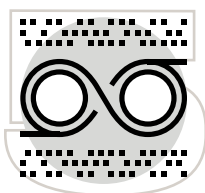


Start by **asking for less** — something that would fit within their zone of acceptance.

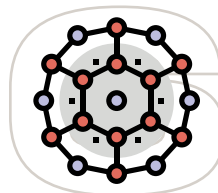


Find a **common point of agreement** that can be used as an “unsticking point” to overcome resistance

SOFTWARE ANTI-PATTERNS



5. MICRO SERVICES: Recommending microservices and loosely coupled architectures **as the only way** to do DevOps.



6. COMPLEX PRESCRIPTIVE FRAMEWORKS: Recommending large, complex, prescriptive frameworks **that were solutions to someone else’s problem** and expecting it to address your issues.

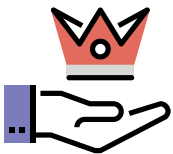
WHAT IF, INSTEAD...

We helped people analyze their own specific products and applications so that they can prioritize the improvements that would help them the most? For example, if their product is a website that owns deployment and has modern technology, then micro-services and empowering the team probably makes sense, and you'll be able to find some people in the movable middle to start influencing that change. If, on the other hand, you're approaching an organization with embedded products or large, tightly coupled systems (like a bank, for instance) and recommending microservices with developers directly pushing code into production, it is probably going to land in the zone of rejection and create more resistance. What if, for these more challenged organizations, we asked for less, instead helping them learn how to analyze those types of applications using a common sense approach to make their challenges visible, instead of forcing them to implement large, complex frameworks designed for someone else? This approach could let them pick solutions that make sense for them and are in their comfort zone, as a first step in the continuous improvement journey. As they have success with smaller steps, their comfort zone expands and they are ready for the next improvements. Do we think this would help reduce the barrier for distance?

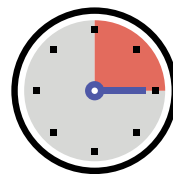
IV. UNCERTAINTY

“Change often involves uncertainty. Will a new product, service or idea be as good as the old one? It’s hard to know for sure, and this uncertainty makes people hit the pause button, halting action.”

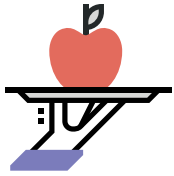
“Catalysts reduce uncertainty by lowering the barrier to trial. Four key ways to do that are to (1) harness freemium, (2) reduce upfront costs, (3) drive discovery and (4) make it reversible.”



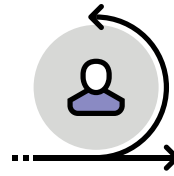
Freemium is used by lots of product companies to remove uncertainty by letting people try a base product for free. Later, it's suggested that customers upgrade to the premium version once they've gotten familiar with the product.



“Catalysts **shrink the amount** of time, money or effort required at the outset to experience something.”

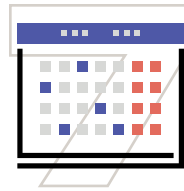


“If people don’t know something exists, or don’t think they will like it, they’re unlikely to go looking to try it.” In these situations, catalysts work to **provide people the experience without them making an effort** to intentionally try it.

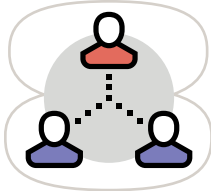


“Making the change easily reversible lowers uncertainty and reduces inertia, encouraging people to change their minds from no to yes.”

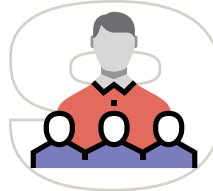
SOFTWARE ANTI-PATTERNS



7. BIG BANG TRANSFORMATIONS: Major transformational **multi-year programs with locked-in, long-range plans** and big consulting engagements.



8. COMPLEX REORGANIZATIONS: Complete re-organizations being required for Agile or DevOps transformations.



9. MANAGING CAPACITY TO 100%: Project Management Offices (PMO) controlling 100% of capacity so that any idea for improvement must go through rigorous formal approval processes.

WHAT IF, INSTEAD...

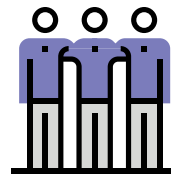
We focused on getting people started on a continuous improvement journey? What if we teach them how to analyze their processes; identify opportunities for improvement and encourage them to implement lots of small improvement projects instead of large, big-bang transformations? These improvement projects can be implemented in monthly, reversible iterations, allowing the teams to modify their plans for next month based on what was learned this month — small iterations where we can learn and adjust without major consequences. What if we allow people to fail in a safe environment? Not everything is going to work, so it’s preferable to expect them to fail fast, learn from the experience and adjust their plans instead of following through on a big, upfront plan that will end in disappointment. In addition, what if these teams are allowed to use a percentage of their capacity for these improvement projects without having to go through big, formal PMO approval processes? And what if they were required to show the results of their improvement projects with metrics that truly demonstrate how they were using this capacity to remove waste and inefficiencies? What if we did all this in the hope that the organization feels good about letting teams invest in the next step of their continuous improvement journey? Do we think these steps to reduce uncertainty would help people to avoid hitting the pause button and stopping potential improvements?

V. CORROBORATING EVIDENCE

“Sometimes one person, no matter how knowledgeable or assured, is not enough. Some things just need more proof. More evidence to overcome the translation problem and drive change. Sure, one person endorsed something, but what does their endorsement say about whether I’ll like it? To overcome this barrier, catalysts find reinforcements. Corroborating evidence.”

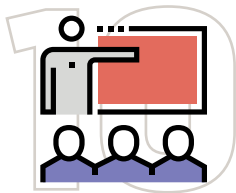


“If **multiple sources** say or do something, it’s harder not to listen. Because, now, there’s corroborating evidence.”



“The more **similar the source** is, the more diagnostic their experiences, preferences and opinions are as a source of information.”

SOFTWARE ANTI-PATTERNS



10. CONFERENCES FOR CORROBORATING EVIDENCE:

Conferences that provide corroborating **evidence from organizations that are very different from your own organization.**

Either by comparing websites to very different applications or comparing very mature transformations to organizations that are just getting started.

WHAT IF, INSTEAD...

We not only sent people to conferences to learn what is going on in the industry, we also encouraged them to identify those in our own organizations who are leading improvement projects? What if we encouraged these leaders to develop communities in their own organizations to share what they are doing and learn from each other? Do we think this would help provide better corroborating evidence and reduce the resistance to change for those just starting?

SUMMARY

The most difficult part of a digital transformation is getting people to invest in needed improvements and embrace new ways of working. It is all about organizational change management. It is unfortunate that experts who focus on influencing change management, like Professor Berger, would describe so much of what I see in my work with software organizations as anti-patterns. I hope this white paper has convinced you to consider different approaches to influencing change instead of repeating anti-patterns and finding resistance again and again. I hope you are encouraged to look outside of the software industry to experts like Professor Berger for ideas. The intent of this white paper was to provide a brief introduction to the ideas from Berger's "Catalyst," and to describe how these ideas can apply to digital transformations. I hope you will also take the time to read the book; it has much more detail, research and examples to better help you influence change. But most importantly, I hope you use these ideas to influence your digital transformation with much greater success.

Acclaimed author and in-demand speaker and executive, Gary Gruver brings a proven track record of transforming software development and delivery processes in large organizations. Gary's demonstrable experience includes leading transformations at HP, resulting in 2-3X productivity improvements on a product team of 800, and implementing continuous delivery at scale for Macy's digital retail dot com business. Today, Gary works with organizations leading digital transformation to help them achieve impactful results in their businesses through consulting, speaking, webinars, presentations, and books. Gary is the author of *Engineering the Digital Transformation* and *Starting and Scaling DevOps in the Enterprise*, and co-author of *Leading the Transformation: Applying Agile and DevOps Principles at Scale* and *A Practical Approach to Large-Scale Agile Development: How HP Transformed LaserJet FutureSmartFirmware*.



Gary Gruver is the CEO of Gruver Consulting, an acclaimed author and an in-demand speaker. Gary brings a proven track record of transforming software development and delivery processes in large organizations.

Speak with the Author:
[Gary Gruver](mailto:gary@garygruver.com) at gary@garygruver.com