



FAIL OFTEN, FAIL EARLY

Accelerating Your Speed to Value:
How to Automate and Monitor
Your Middleware Environments.

Written by MidVision and Nastel

FAIL OFTEN | FAIL EARLY

Contributors: Helen Beal - Head of Sales and Marketing at MidVision | Charley Rich - VP of Marketing at MidVision | Dave Sayers - Director of Technical Services at MidVision

It was Confucius who said: “our greatest glory is not in never failing, but in rising every time we fail” and it is a widely held belief that we learn through our mistakes, ultimately attaining wisdom, and that a cautiously led life constrains us from many possible opportunities and successes. And thus these philosophies can be applied to the management of middleware releases - you need to deliver ever more functional, propagating, applications to your business and you know that things will go wrong, systems will fail, releases will fail, you will fail. But you will fix it. The key is to be prepared for failure, to know that it will occur often, to encourage it to happen early in the process so you can catch it, so you don't get caught out.

At MidVision and Nastel we innovate technology that makes the life of IT operations staff easier, making failure more predictable, more fixable. Our application release automation and performance management tools do more than just automate and monitor your middleware. They reduce your company's time to market, scale to enterprise levels, provide cross-project visibility, enable compliance and security and increase resource utilization.

During your day to day work, managing releases of the varying multitude of applications your business requires to run effectively, service your customers, transact your earnings, keep you competitive, you might be wondering: “How do I know I failed?” and when you do: “Do I have enough time to take action and repair this before our customers are impacted? Or I get a visit from the CIO.”

What you need is an Early Warning System. You need to know your Mean Time to Know (MTK) and your Mean Time to Repair (MTR) - that is, on average, how long is it before you find out you failed, and how long until you fix it, either by rolling back to a previously successful release, or making the new one work. You know you will fail, but you need to know as soon as possible when you have, giving you the maximum amount of time to fix it, preferably before anyone else sees.

If you are conducting all of your application release deployments manually or using a bunch of scripts, the chances are the first time you know you've failed is when someone screams at you. And you're going to struggle to troubleshoot when you've done everything this way, which means that you're going to be taking some time, more time than you want, possibly with your CIO stood behind you, breathing down your neck, to fix the damn thing.

So how do you build your early warning system? We see four parts to the process that will keep you out of the hot water:

DEPLOY | MONITOR | ANALYZE | IMPROVE

DEPLOY

Let's start with deployment. Like we said before, if you're deploying applications and releases manually or using scripts you're asking for trouble. You'll have no way of recording what's been done to the systems and when you have tens, hundreds or even thousands of applications running there's a lot that can go wrong. Using a tool to automate deployment doesn't just drastically reduce the time it takes to release applications (making your business more agile and happy) it also means that you have an audit trail of activity meaning you can triage the systems when they break quickly and easily (keeping your CIO upstairs). Using snapshot and compare capabilities means you can see instantly what change has been made, you can make a call whether to fix the problem then and there, or press a button to redo the last successful release (pew). Not only this, but recording all activity, all change, whether authorized or not, means your organization can be more comfortable about compliance (helping your CFO to sleep at night).

MONITOR

Fantastic, so you have successfully deployed your new application. It's time to go and make a cup of tea. But while you're gone, your new application attracts a gazillion new customers and your carefully architected virtualized environment takes a massive performance hit. Is the first thing you hear about this from the call centre when they hear an irate voice? Or when the marketing team sees a complaint on a forum? Or does your smart-phone inform you there's a problem while you wait for the kettle to boil? You shouldn't have to wait to fail. You should know immediately at the point of failure that there is a problem. You should know first. Or even better, you should know before it happens.

ANALYZE

We're not suggesting you develop psychic abilities, but we are saying you should get predictive. Or even better, pre-emptive. The knowledge you are going to fail is one thing, the knowledge of when and how you are going to fail is something else entirely.

IMPROVE

Your business wants you to do more. You want to do more. We all want to grow and develop and do better, right? And it's a whole lot easier if you know where you're starting from. And even more satisfying if you can see your progress. More throughputs, less downtime, more done with less resources. Showing your boss all of this will put a smile on her face and could have a positive direct impact on your job title and your pay packet.