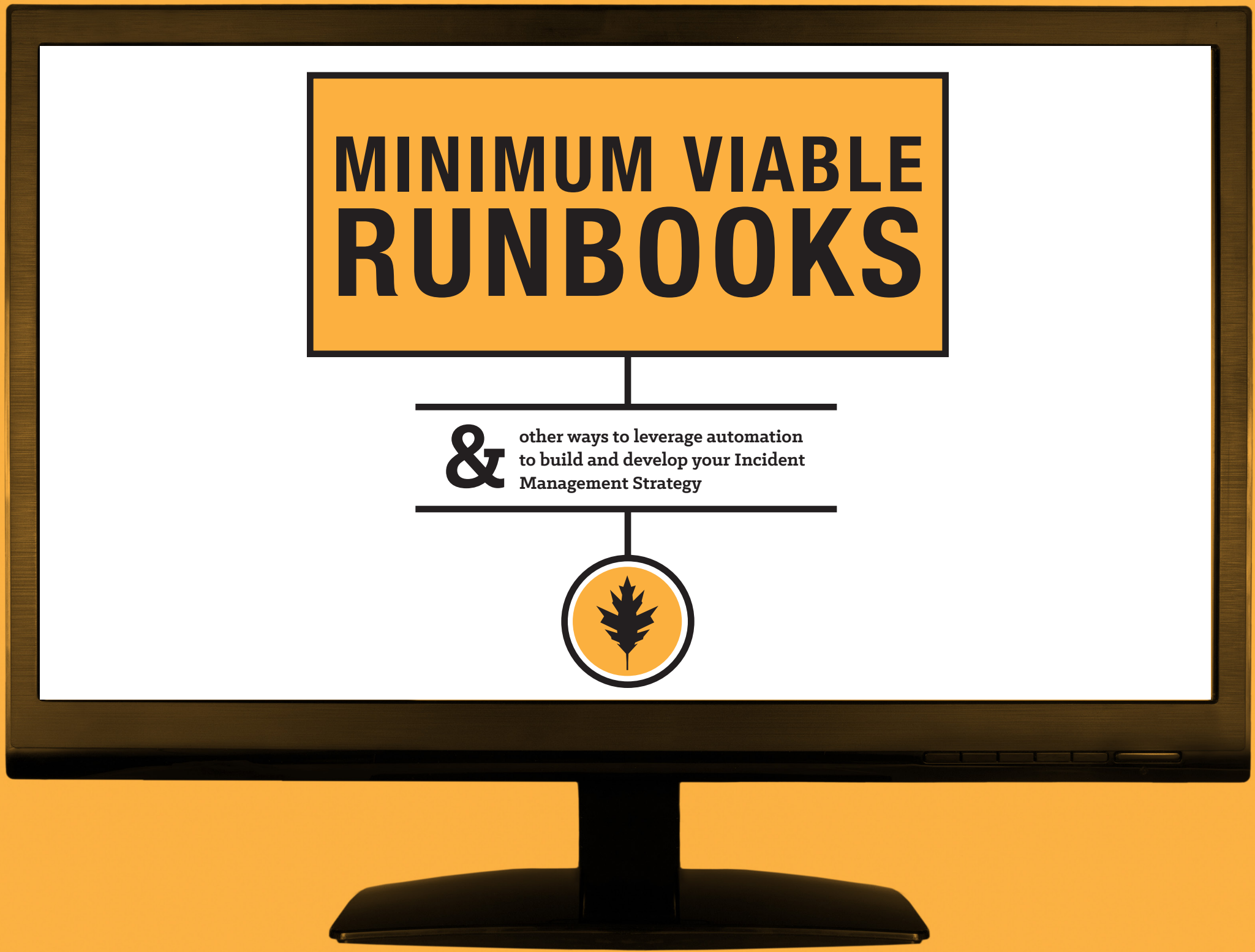

MINIMUM VIABLE
RUNBOOKS
&
other ways to leverage automation
to build and develop your Incident
Management Strategy

RUNBOOK

run·book *noun* 1. In a computer system or network, a 'runbook' is a routine compilation of procedures and operations that the system administrator or operator carries out. System administrators in IT departments and NOCs use runbooks as a reference. Runbooks can be in either electronic or in physical book form. Typically, a runbook contains procedures to begin, stop, supervise, and debug the system. It may also describe procedures for handling special requests and contingencies. An effective runbook allows other operators, with prerequisite expertise, to effectively manage and troubleshoot a system. Through runbook automation, these processes can be carried out using software tools in a predetermined manner.

a: It is impossible to create a runbook for every single incident that can happen, let alone predict the next incident.

b: "We should be using runbooks but we don't"

max

min

MINIMUM VIABLE PRODUCT

MVP

In product development, the minimum viable product (MVP) is the product with the highest return on investment versus risk. The term was coined and defined by Frank Robinson, and popularized by Steve Blank, and Eric Ries.

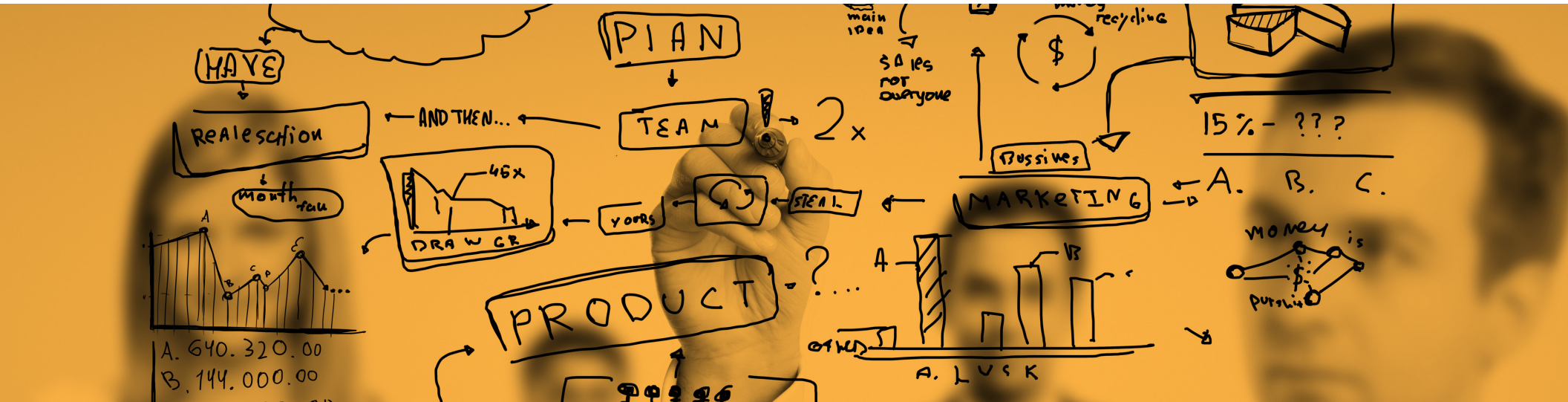
An MVP is not a minimal product,* it is a strategy and process directed toward making and selling a product to customers. It is an iterative process of idea generation, prototyping, presentation, data collection, analysis and learning.

* http://web.archive.org/web/20140323181121/http://radoff.com/blog/2010/05/04/minimum-viable-product-rant/

MINIMUM VIABLE RUNBOOK

MVR

In IT Process development, the minimum viable runbook (MVR) is the runbook with the highest return on valuable information versus time spent creating it. It is a strategy and process directed toward making and implementing runbooks for your IT team. It is an iterative process of idea generation, prototyping, automation, presentation, information capturing, analysis and learning.



A DAY LATE AND A RUNBOOK SHORT

AM

Your teams are busy. There is little value in carving out time to have a dedicated team member build runbooks, design processes and gather information from previous successes when you have 24x7 operations that need their attention.

Incidents are inevitable, outages are unforeseeable, and frustration can quickly become an internal IT cultural norm. Thus, runbooks are important in providing your teams with contextual documents to support their efforts

How do we help these teams to be successful in this very important, but not urgent, task of creating runbooks?

How do we ensure that our teams are prepared for the next incident?

How do we ensure consistency in our recovery processes?

What actions are we taking to truly lower downtime?

DEVELOP YOUR INCIDENT MANAGEMENT STRATEGY

LEAN STYLE

When using the minimum viable approach to building your runbooks, you want to leverage digital automation to capture your team's activity, collaboration, and actions.

By having a record of what actions and activities have worked, you can then leverage the recorded actions (that fixed the problem) to build your Minimum Viable Runbook. Simply review the conversations, actions or collaborations that were previously captured, and restate them into an "action-plan" as clear steps to resolution.

WHAT DOES IT LOOK LIKE

Your runbook at a minimum should include:

THE WHAT The first thing your team member should review in order to confirm the problem (which monitoring tools, status sites, charts). We want to save brain cycles and reduce confusion by focusing on the very first THOUGHT that is needed to start the problem solving.

THE HOW We now move right into the next step of building the very first ACTION that should be made to remediate the problem (i.e. server resets, QOS policy adjustments, etc.)

THE WHO (no, not the band) In the case of needing to include additional team members, your runbook should include the specific teams or team members to contact if you need to escalate the incident

THE WHERE These are the tools and digital locations of where to record notes, update statuses, post questions, record activities, etc.

MAKING THE RUNBOOKS WORK, NOT SIT IN THE CORNER

This is where the rubber meets the road. You now need to have these mini runbooks be front and center, right in your teams' faces when the incident happens.

This reduces or eliminates the time it takes for your team member to "dig" for your runbooks in a file repository or a wiki link. This also reduces the stress accompanied with solving a complex problem at 4am.
Solution: Implement a "rules engine" using an "if this, then that" approach to your incoming alerts.

Your Runbooks then need to give Call-To-Action instructions, clickable links, viewable graphs (live) and static images of graphs for reference.

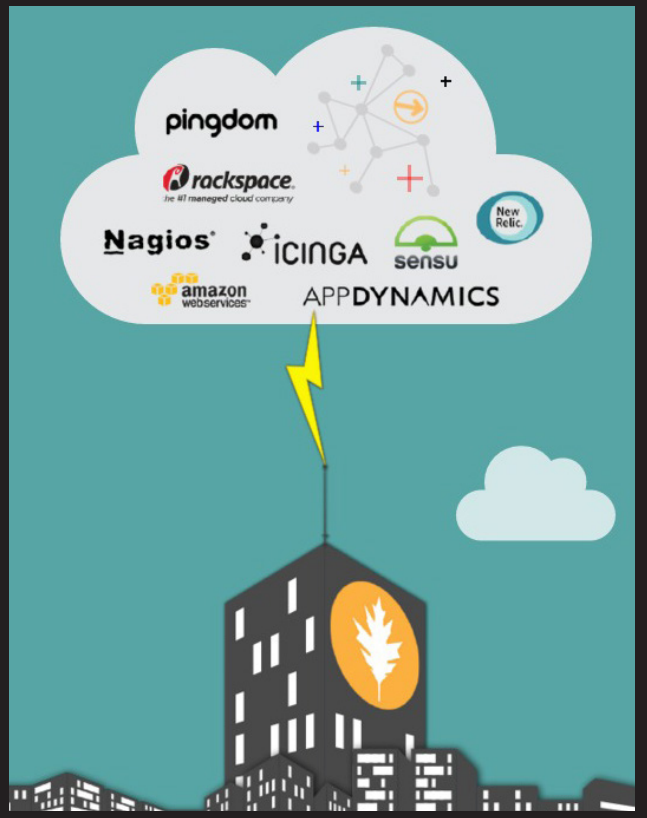
This gives the incident team member instant context on not only how to fix the problem, but how to better understand what to look for.
Solution: Have your runbooks or alerting system display contextual actions based on payload information and provide your teams with a URL that can be accessed alongside the alert using your rules engine. The URL should link to a reference document or a detailed runbook. Note: These URLs must point to a single point of reference, thus making it easier for version control and consistent messaging.

INSTRUCTIONS FOR CREATING A MVR

1. Choose a platform that allows you to send all alerts and chats into a single incident timeline. This makes it easier to capture all the information necessary to have a successful post-mortem and begin work on the MVR.

2. Implement Best Practices to ensure that teams are collaborating in the platform so that the actions/ knowledge are captured.

3. After an incident, review the timeline of alerts, chats, etc and begin to decipher what worked and what didn't. ☐ ☑



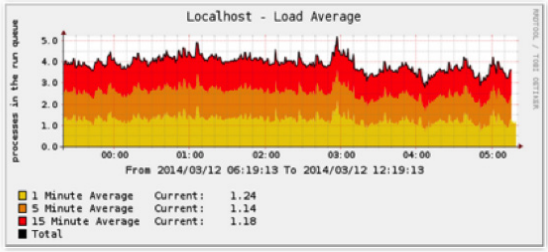
ALMOST THERE

4

Use your findings and build basic action plans with clear steps on how to resolve the issue, who to contact, what to refer to, where to find documentation, etc.

Who: 1. **Attention:** Despite the name of this alert this represents a degradation of customer experience. Notify support@victorops.com once you begin work.

What: 2. Load Average Graph:



Where: 3. Runbook

How:

AND, FINALLY...

5

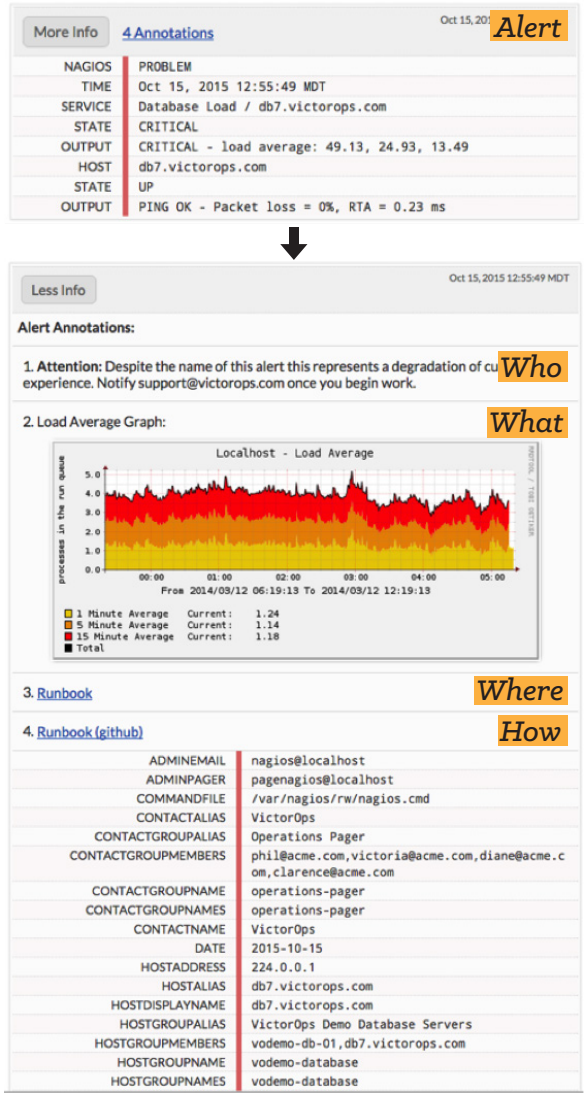
Append incoming alerts with the MVR. Make that action plan available to the team member, right when they need it most…when the same incident/alert happens again.

6

Inspect and adapt. Review and improve. Measure and share.
(Or insert any other two words that reminds you that we are here to learn and grow…oooh those are good too)

7

Repeat steps 1-6 for other team, services, tools, etc.



THE GOAL

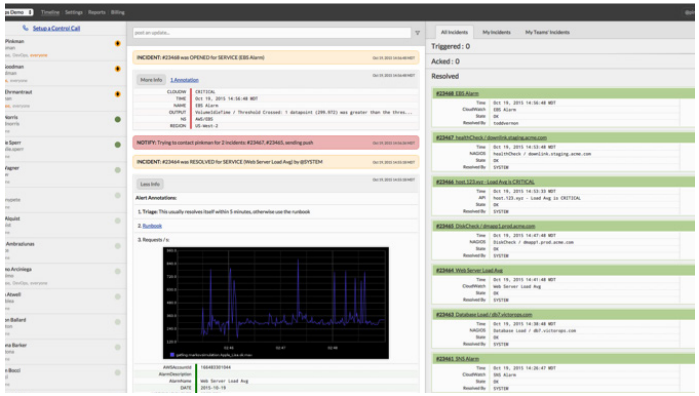
To have the minimum viable runbook available at the fingertips of the IT team member, upon the occurrence of the critical alert or incident.

There doesn't need to be that much information, but enough to help the teams start looking in the right place and thinking about the right actions.

Even if you do not have a URL link to a mature runbook, a few sentences with basic instructions beats having nothing at all.

VICTOROPS

All of the screen captures (visuals) were captured from the VictorOps On-Call and Incident Management Platform.



The "rules engine" behind the annotated/transformed alerts is a powerful feature in the VictorOps platform, called the Transmogrifier.

To learn more, visit: https://victorops.com/transmogrifier/

For a live demo of how you can begin creating your minimum viable runbooks using VictorOps, please contact sales@victorops.com
Or learn more by visiting www.VictorOps.com

LEVERAGING WIKIS AS YOUR RUNBOOK MEDIUM

The nature of wikis being editable by its users is collaborative in nature.

Wikis allow your teams to collaborate on a single document that is sharable and consistent.

The single source of record enables the team to begin exchanging tribal knowledge.

Creating URLs to wikis provides better navigation to the correct documentation rather than a folder maze on a shared drive

Having the URLs to Wikis attached to your alerts will increase visibility to your runbooks, thus making the most of your team's runbook efforts. Rather than letting the work get lost in the repository.

Increased visibility leads to more opportunity to gather more feedback for continuous improvement.

The "rules engine" behind the automated URL attachment to alerts is a powerful feature in the VictorOps platform, called the Transmogrifier.

To learn more, visit: https://victorops.com/transmogrifier/


MINIMUM VIABLE
RUNBOOKS
And remember... DevOps
is about people, not tools.