

How I Learned To Stop Worrying And Love Production Support...Even on Feature Teams



DAVID FRINK



David Frink

Agile Coach & Consultant

Release Train Engineer <- Production Support Manager <- Dev
Manager <- Developer

Eliassen Group, Ipreo, Teradata, SciQuest, Frink Consulting
PSM, ICP-ACC, ICP-ATF, ICP, SAFe 4 RTE

www.dfrink.com

david@dfrink.com

Who Saves More Lives?

FIREFIGHTERS OR FIRE MARSHALS?



Who Is In The Room?

My Production Support Journey

My Story – Background

Production Support Manager

Dedicated team (of dedicated people)

Enterprise SaaS

- ~250 customers, ~12k commands/min, ~50k unique users per day

24/7 on call

Triage, bug fixes, code promotions, application restarts, tuning, report/data requests, etc.

My Story – Before/After

BEFORE STATE

Morale issues

Not hitting guideline dates

2-3 unplanned restarts per week

Burnout – too few people in on-call rotation

Assumption manager would make many decisions

AFTER STATE

Good work-life balance

More people in on-call rotation

95%+ guideline dates met

Continuous improvement culture

Over a month without unplanned restart

Team made most decisions



Agenda

Agenda

Production Support Defined

Common (Anti)Patterns for Production Support

Team Structure – Pros/Cons

How to do Production Support Well

Blameless Postmortem Technique

Conclusion

Production Support Defined

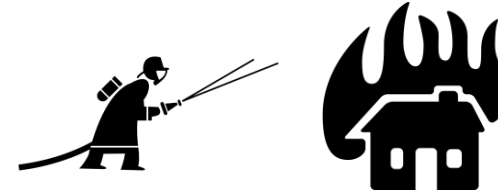
What is Production Support?

Technical teams, supporting a live application (Tier 3 Support)

Incident Management, Problem Management, Support Requests

Incident Management (Emergencies)

- Emergency response
- Triage and quick fix or work around the problem



Problem Management (Bugs)

- Root cause
- Long-term fix



Also, some investigation/report requests (Support Requests)

Common (Anti)Patterns for Production Support

Incident Management (Emergencies)

- Who is on-call?
- Email for emergencies
- Poor communication
- Whack-a-mole
- Insufficient tools & Monitoring

Problem Management (Bugs)

- Multiple intake methods
- No reproducible steps

Planning

- Impacts predictability of features
- Don't visualize prod support work

Metrics & Measurement

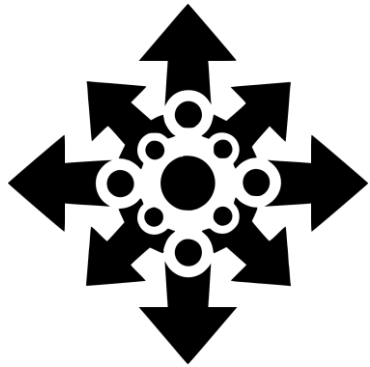
- Not having any

Continuous Improvement

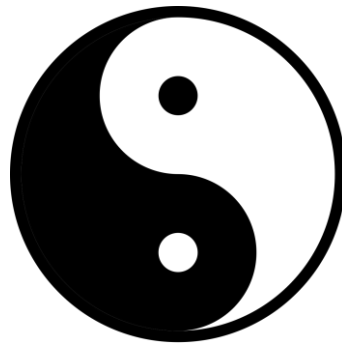
- No training in triage
- No systems thinking

Team Structure – Pros/Cons

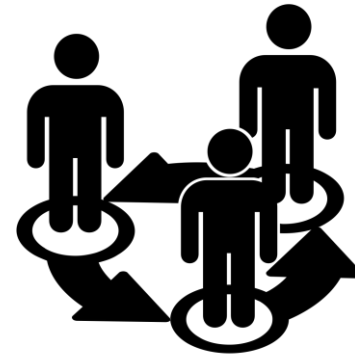
Production Support Team Structures



Chaos



Hybrid
(Features &
Support)



Dedicated,
Rotating



Dedicated,
Permanent

Your experience?

Find a partner/small group

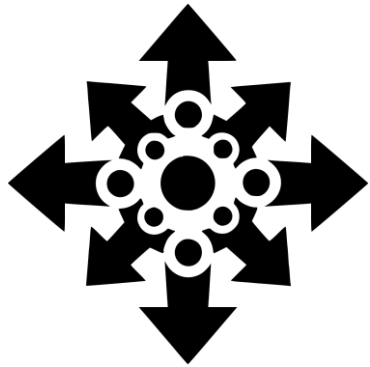
Within your group discuss how your team is structured to handle production support

Discuss pros/cons of your approach

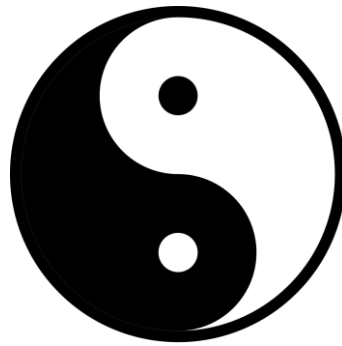
3-4 minutes

Raised hand will be the sign to stop the discussion

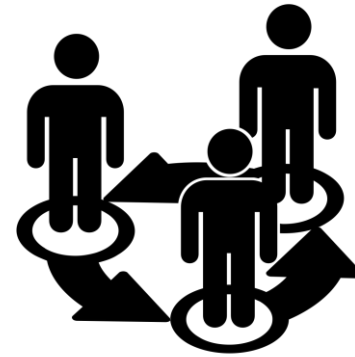
Production Support Team Structures



Chaos



Hybrid
(Features &
Support)



Dedicated,
Rotating



Dedicated,
Permanent

Dedicated, Permanent Production Support Team



PROS

Feature teams are protected from interruption

People become experts at triage and investigation

Team creates better tooling & monitoring

Depth of knowledge about how the application behaves in the real world

People who like firefighting can gravitate here

CONS

Burnout

Very hard to hire people into this team

No feedback loop to the teams causing the production problems

Dedicated, Rotating Production Support Team



PROS

More developers get experience with triage and investigation

Prod Support mindset feeds back to feature teams

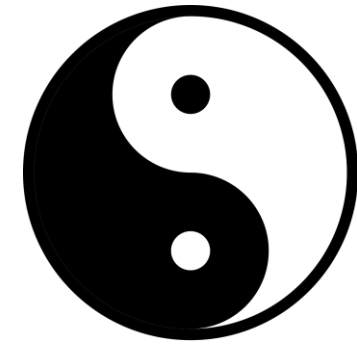
CONS

Constant newbies means some things will get mis-handled

Attitude of developers as they rotate on

Not everyone is good under pressure

Hybrid Team (Features and Production Support)



PROS

People creating the features support the features

Learning from real-world feeds back into current and future stories

Higher incentive to build in monitoring/alerting up-front

CONS

Messes with predictability

Competes with Feature work

Not enough people for 24/7 support (team too small/inexperienced)

Burnout/attitude towards work

Not everyone is good under pressure

Which Model Is Best?

“It depends”

Based on your application, architecture, domain, customers (b2b vs b2c)...

Choose what you need to optimize for

Do you need full-time firefighters? 

Or would volunteer firefighters be sufficient?

My Belief

Hybrid model yields the best long-term results

- Accelerates learning, connects team to customers

A dedicated team may be more **efficient** at reacting and closing issues



However, a hybrid team may be more **effective** in the long run as they bake-in quality, scalability and resiliency to make their own lives easier.



Might you need both?

How To Do Production Support Well

What Enables Good Production Support?

Start with **Kanban Principles & Properties**

Foundational Principles

- Start with what you do now
- Agree to pursue incremental, evolutionary change
- Respect the current process, roles, responsibilities & titles

5 Core Properties

1. Visualize the workflow
2. Limit WIP
3. Manage Flow
4. Make Process Policies Explicit
5. Improve Collaboratively

Foundational Principles of Kanban

Start with what you do now

- New responsibility

Agree to pursue incremental, evolutionary change

- Production Support isn't something to simply survive
- Team must treat it as a discipline and want to improve at it

Respect the current process, roles, responsibilities & titles

Visualize the workflow

What are the new statuses/“state changes” with Production Support?

- Beyond “New, In Progress, Done”

Workflow before/after the team

- Think full value stream

Prod Support work must be visible and on equal footing with Feature work

- This is “real” work – don’t keep it hidden

Limit Work in Progress (WIP)

Prod Support is unplanned, urgent and non-optional

- This can expand to fill available time

Not all Prod Support is an emergency

Classes of Service

- Emergencies
- Bugs (ASAP, Next Maintenance Release, Next Major Release)
- Support Requests

WIP – shows overload and bottlenecks

Manage Flow

Emergencies – single incoming communication channel

- (It isn't email)
- Paging solutions (Pager Duty, OpsGenie, etc.)

Bugs & Support Requests – Single intake method

- (It isn't email)
- Your existing issue tracking system – linked with the CRM if possible
- Review on a cadence

Dedicating people to Prod Support increases flow

Make Process Policies Explicit

Communication paths & work intake

Paging System (On-call rotation & escalation path)

Response targets (SLAs)

Definition of Ready (DoR) to accept a request – send back if not enough info

- Environment, Customer, Severity/Urgency, Product/Module, Problem Description, etc.
- Reproducible Steps – Link to where this has been reproduced internally (if possible)

Requests can't be casual or unstructured

Improve Collaboratively

Blameless Postmortems

Don't get better at reacting, get better at preventing

Full Value Stream - Bring Customer Support into the discussion

Additional Tips

Everyone Needs A Hero...

Who is the Superhero this week?

Who is the Side-kick?

Superhero responsibilities:

- On-call
- Interruptible resource – protect focus
- Work production issues/bugs (pull in others as needed)

Most important question: “What does the Superhero do if there aren’t problems?”

- Resist the urge to say “Feature work”
- “Make Production Support easier for the next Superhero”



Planning Work

“Control the damage”

Two queues of work, visualized together

Create slack for prod support

Ideas for capacity planning

- On-call person – pretend like they’re out on vacation for the sprint and adjust capacity accordingly
- Placeholder story – with story points
- Scrumban/Kanban – separate “class of service” (swimlane) with WIP limit

Other Tips

Say, “I’ve got it”

Make everything self-service

Be able to answer, “What % of the team’s capacity is Production Support”

You might need to point bugs to do this



Tracking hours is another option

Track how many, by severity and type, over time

Logging/monitors/alerts

- Ask, “could the system have reported the issue first?”
- Log aggregation (Splunk, etc.)

Blameless Postmortem

Blameless Postmortem Retrospective

Retrospective on a specific production incident

Done only after the fire is out

Inspired by “Blameless Postmortems and a Just Culture”
about Etsy

“Blameless” is a powerful word

- Do people or systems cause failures?

The focus is on **how** the problem occurred rather than **who**
did what

Be thinking of your experiences...

Blameless Postmortem Retrospective

Serves 2 purposes:

- Ideas for improvement
- Training

Do these for any Sev 1 & 2 – recommend wiki

Track improvements – in an improvements backlog

Blameless Postmortem - Format

Incident Artifacts (documents, emails, issue links)

Summary – Impact to customer

Timeline – Alerts, communications, monitor status, actions

5 Questions

- Detect – How do we detect this problem or a problem like this sooner?
- React – How do we improve our reaction to issues like these?
- Quick Fix – How do we stop the bleeding faster?
- Prevent – How do we prevent **or** reduce the impact of issues like this in the future?
- Other Areas of Risk – What other areas share this same vulnerability?



Running the Blameless Postmortem

Invite prod support team plus those who reported the issue

Introductions

Review what “blameless” means (do this every time)

Have the team talk through summary & timeline, filling in details

Walk the 5 questions

Find the ideas for improvement the team wants to try

Get owners & dates on each one

Add to improvements backlog

Disappearing Fax Provider

Support notified by supplier faxes weren't distributing – emailed Prod Support

Looked at the queue – thousands of failures

Restarted fax service – didn't help

Error code not helpful

Called the provider – they said we hadn't paid them in 6 months

Got them paid

Faxes re-enabled

Manually resent the orders (painfully)

Disappearing Fax Provider - BPM

Detect

- Monitors
- Email list of failures every hour

React

- Translate error code into plain language
- Don't send an "urgent" email

Quick Fix

- Bulk re-send option

Prevent

- Secondary fax provider
- Double check payments to 3rd party vendors – update contacts

Other areas...

- Requisitions, Invoices, etc.
- Anywhere there is a queue without a monitor

Your Turn

Find a partner/small group

Describe a recent production problem (in 30 seconds or less)

Walk the 5 questions – make note of new ideas

4-5 minutes

Raised hand will be the sign to stop the discussion

Ideas From Your Stories

Detect?

React?

Quick Fix?

Prevent?

Other Areas...?

Conclusion

Conclusion

Production Support is hard, necessary and thankless

Doing well, and embedding it into feature teams can be a competitive advantage

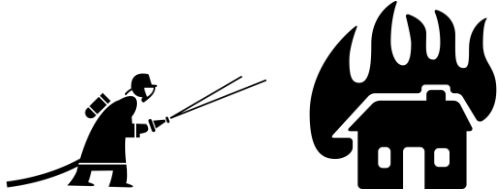
Kanban Principles & Properties

Find your Superhero

Don't use email

Blameless Postmortem is the engine for improvement

“Responding to change” includes Production Support



Thanks!

QUESTIONS?



<https://dfrink.com/prodsupportagile2018/>

