Clearing the Way
For SRE in the Enterprise

Damon Edwards
@damonedwards
Digital Agile DevOps CI/CD
Cloud Docker Kubernetes Microservices

That is cool
Wow
I wish I could work there
“Great for Dev, but what about Ops?”

Digital and DevOps

Formation of

"Go faster!"
“Open up!”

Ops

Shorter Time-to-Market
Improved Quality

Business Idea

Dev

Ops

Fast Feedback from Users

Running Services

Availability
Auditing

Security
Compliance

“Lock it down!”
Our transformation has largely ignored Ops. Any ideas?

Have you heard of SRE? Google does it.
Jane Doe
Systems Administrator
We have SysAdmins

Jane Doe
Systems Administrator
They should be SREs!
They should be SREs!
Your new title is SRE.
Now write code and be better at ops.
<table>
<thead>
<tr>
<th>SysAdmins</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overloaded. Constant firefighting.</td>
</tr>
<tr>
<td>Waiting in ticket queues for everything.</td>
</tr>
<tr>
<td>Things break. Break again. And again.</td>
</tr>
<tr>
<td>Everyone is busy, but it doesn’t get any better.</td>
</tr>
</tbody>
</table>

**Executive View**

Everything takes too long, cost too much, and break too often!
SysAdmins

- Overloaded. Constant firefighting.
- Waiting in ticket queues for everything.
- Everyone is busy, but it doesn’t get any better.

Executive View
Everything takes too long, cost too much, and break too often!

(False) SRE

- Overloaded. Constant firefighting.
- Waiting in ticket queues for everything.
- Everyone is busy, but it doesn’t get any better.

Executive View
Everything takes too long, cost too much, and break too often!
Changing job titles or adding individual skills doesn’t make systems administrators SREs.
Principles of SRE are what set SRE apart
Principles of SRE are what set SRE apart

1. SRE needs Service Level Objectives, with consequences
Principles of SRE are what set SRE apart

1. SRE needs Service Level Objectives, with consequences
SLO and Error Budgets: Tools for Shared Responsibility

Error Budget*

Service Level Objective

Service Level Indicator

(*Use this to improve the service)
SLO and Error Budgets: Tools for Shared Responsibility

DEV  BIZ  Ops

Service Level Objective  Error Budget

(*Use this to improve the service)
Principles of SRE are what set SRE apart

1. SRE needs Service Level Objectives, with consequences
Principles of SRE are what set SRE apart

1. SRE needs Service Level Objectives, with consequences
2. SREs have time to make tomorrow better than today
Principles of SRE are what set SRE apart

1. SRE needs Service Level Objectives, with consequences
2. SREs have time to make tomorrow better than today
3. SRE teams have the ability to regulate their workload
Principles of SRE are what set SRE apart

1. SRE needs Service Level Objectives, with consequences
2. SREs have time to make tomorrow better than today
3. SRE teams have the ability to regulate their workload
Forces That Undermine SRE Principles

Silos

Queues

Excessive Toil

Low Trust
Forces That Undermine SRE Principles

Silos

Queues

Excessive Toil

Low Trust
I need X

Silos

Backlog
Information
Tools
Priorities
Silos

Silo A

- Backlog
- Information
- Tools
- Priorities

I need X

Silo B

- Backlog
- Information
- Tools
- Priorities

I do X

Requests for X
Silos cause disconnects and mismatches

I need X

Requests for X

I do X

Silo A

Silo B

Context

Process

Tooling

Capacity

Backlog

Information

Priorities

Backlog

Information

Priorities

Backlog

Information

Priorities

Context

Process

Tooling

Capacity

RUNDECK
Silos Interfere with feedback loops

1. Dev → Ops

2. Dev → Ops

3. Dev → Ops
Silos Interfere with feedback loops

1. Producer: Ops → Ops
2. Producer: Ops → Ops
3. Producer: Ops → Ops
Silos create labor pools of functional specialists

Requests fulfilled by semi-manual or manual effort

Primary management focus is on protecting team capacity
Silos Undermine SRE Principles

1. Org has Service Level Objectives, with consequences?

2. SREs have time to make tomorrow better than today?

3. SRE teams have the ability to regulate their workload?
Silos Undermine SRE Principles

1. Org has Service Level Objectives, with consequences?
   *Disjointed silos make meaningful SLOs and shared responsibility almost impossible*

2. SREs have time to make tomorrow better than today?

3. SRE teams have the ability to regulate their workload?
Silos Undermine SRE Principles

1. Org has Service Level Objectives, with consequences?
   - Disjointed silos make meaningful SLOs and shared responsibility almost impossible

2. SREs have time to make tomorrow better than today?
   - Siloed labor pools, disconnected processes and tools, and slow feedback loops tend to consume all available capacity

3. SRE teams have the ability to regulate their workload?
Silos Undermine SRE Principles

1. Org has Service Level Objectives, with consequences?
   - Disjointed silos make meaningful SLOs and shared responsibility almost impossible

2. SREs have time to make tomorrow better than today?
   - Siloed labor pools, disconnected processes and tools, and slow feedback loops tend to consume all available capacity

3. SRE teams have the ability to regulate their workload?
   - Struggling to keep up with demand and unable to protect capacity
Forces That Undermine SRE Principles

- Silos
- Queues
- Toil
- Low Trust
How do we cover for our cross-silo disconnects and mismatches?
How do we cover for our cross-silo disconnects and mismatches?
We all know how well that works
Request queues are an expensive way to manage work

Queues Create...

- Longer Cycle Time
- Increased Risk
- More Variability
- More Overhead
- Lower Quality
- Less Motivation

Adapted from Donald G. Reinertsen, The Principles of Product Development Flow: Second Generation Lean Product Development
What do queues do to value streams?
What do queues do to value streams?
What do queues do to value streams?

Queues disintegrate and obfuscate value streams.
Tickets queues become “snowflake makers”
Tickets queues become “snowflake makers”

Silo A

Silo B

Snowflakes

(each unique, technically acceptable but un reproducible and brittle)
Ticket Queues Undermine SRE Principles

1. Org has Service Level Objectives, with consequences?

2. SREs have time to make tomorrow better than today?

3. SRE teams have the ability to regulate their workload?
Ticket Queues Undermine SRE Principles

1. Org has Service Level Objectives, with consequences?
   - Tickets reinforce siloed behaviors and obfuscate the value stream

2. SREs have time to make tomorrow better than today?

3. SRE teams have the ability to regulate their workload?
Ticket Queues Undermine SRE Principles

1. Org has Service Level Objectives, with consequences?
   🗑 Tickets reinforce siloed behaviors and obfuscate the value stream

2. SREs have time to make tomorrow better than today?
   🗑 Longer cycle time, more variability, more overhead, lower quality, and more snowflakes consume available capacity

3. SRE teams have the ability to regulate their workload?
Ticket Queues Undermine SRE Principles

1. Org has Service Level Objectives, with consequences?
   - Tickets reinforce siloed behaviors and obfuscate the value stream

2. SREs have time to make tomorrow better than today?
   - Longer cycle time, more variability, more overhead, lower quality, and more snowflakes consume available capacity

3. SRE teams have the ability to regulate their workload?
   - Queues obfuscate the pressure being put on request fulfillers
Forces That Undermine Operations

- **Silos**
- **Queues**
- **Toil**
- **Low Trust**
Toil is the enemy of SRE
Toil is the enemy of SRE

“Toil is the kind of work tied to running a production service that tends to be manual, repetitive, automatable, tactical, devoid of enduring value, and that scales linearly as a service grows.”

-Vivek Rau
Google
## Toil vs. Engineering Work

<table>
<thead>
<tr>
<th></th>
<th>Toil</th>
<th>Engineering Work</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lacks Enduring Value</td>
<td>Builds Enduring Value</td>
<td></td>
</tr>
<tr>
<td>Rote, Repetitive</td>
<td>Creative, Iterative</td>
<td></td>
</tr>
<tr>
<td>Tactical</td>
<td>Strategic</td>
<td></td>
</tr>
<tr>
<td>Increases With Scale</td>
<td>Enables Scaling</td>
<td></td>
</tr>
<tr>
<td>Can Be Automated</td>
<td>Requires Human Creativity</td>
<td></td>
</tr>
</tbody>
</table>
Excessive toil prevents fixing the system

Toil at manageable percentage of capacity

Reduce toil

Toil

Engineering Work

Improve the business
Excessive toil prevents fixing the system

Toil at manageable percentage of capacity

- Reduce toil
- Improve the business

Toil at unmanageable percentage of capacity ("Engineering Bankruptcy")

- No capacity to reduce toil
- No capacity to improve business
Excessive Toil Undermines SRE Principles

1. Org has Service Level Objectives, with consequences?

2. SREs have time to make tomorrow better than today?

3. SRE teams have the ability to regulate their workload?
Excessive Toil Undermines SRE Principles

1. Org has Service Level Objectives, with consequences?
   - Buried in toil keeps team from contributing engineering work to uphold their end of the shared responsibility deal

2. SREs have time to make tomorrow better than today?

3. SRE teams have the ability to regulate their workload?
Excessive Toil Undermines SRE Principles

1. Org has Service Level Objectives, with consequences?
   - Buried in toil keeps team from contributing engineering work to uphold their end of the shared responsibility deal

2. SREs have time to make tomorrow better than today?
   - Buried in toil… no capacity for engineering work to reduce toil.

3. SRE teams have the ability to regulate their workload?
Excessive Toil Undermines SRE Principles

1. Org has Service Level Objectives, with consequences?
   - Buried in toil keeps team from contributing engineering work to uphold their end of the shared responsibility deal

2. SREs have time to make tomorrow better than today?
   - Buried in toil... no capacity for engineering work to reduce toil.

3. SRE teams have the ability to regulate their workload?
   - Buried in toil... no capacity for engineering work to reduce toil.
Forces That Undermine Operations

Silos

Queues

Toil

Low Trust
Where are decisions made? Who can take action?

Decisions made here
All work is contextual
All work is contextual

```
rm -rf $PATHNAME
```
All work is contextual

```
rm -rf $PATHNAME
```

Is this dangerous?
All work is contextual

rm -rf $PATHNAME
All work is contextual

```
rm -rf $PATHNAME
```

@@ -1,2 +1,2 @@
-<!-- Status: Ok --> +<!-- Status: OK -->
All work is contextual

```
rm -rf $PATHNAME
```
All work is contextual

```
rm -rf $PATHNAME
```
All work is contextual

rm -rf $PATHNAME

Answer is always “it depends”
Where are decisions made? Who can take action?
Low trust + approvals = illusion of control
Low trust + approvals = illusion of control

Add up the total number of approval requests and
Low trust + approvals = illusion of control

Add up the total number of approval requests and

...subtract the info radiators (“I need to be in the loop”)
Low trust + approvals = illusion of control

Add up the total number of approval requests and

...subtract the info radiators (“I need to be in the loop”)

...subtract the CYAs (“Prove you followed the process”)

Ticket System
Low trust + approvals = illusion of control

Add up the total number of approval requests and

...subtract the info radiators (“I need to be in the loop”)

...subtract the CYAs (“Prove you followed the process”)

...subtract the too removed to judge (“mostly guessing”)

Ticket System
Low trust + approvals = illusion of control

Add up the total number of approval requests and

...subtract the info radiators ("I need to be in the loop")

...subtract the CYAs ("Prove you followed the process")

...subtract the too removed to judge ("mostly guessing")
Low trust + approvals = illusion of control

Add up the total number of approval requests and

...subtract the info radiators (“I need to be in the loop”)

...subtract the CYAs (“Prove you followed the process”)

...subtract the too removed to judge (“mostly guessing”)

Ticket System
Low trust + approvals = illusion of control

Add up the total number of approval requests and

...subtract the info radiators (“I need to be in the loop”)

...subtract the CYAs (“Prove you followed the process”)

...subtract the too removed to judge (“mostly guessing”)

How many are you left with?
Low trust + approvals = illusion of control

Add up the total number of approval requests and

...subtract the info radiators (“I need to be in the loop”)

...subtract the CYAs (“Prove you followed the process”)

...subtract the too removed to judge (“mostly guessing”)

How many are you left with?
How many were the right call?
Low trust + approvals = illusion of control

Add up the total number of approval requests and

...subtract the info radiators ("I need to be in the loop")

...subtract the CYAs ("Prove you followed the process")

...subtract the too removed to judge ("mostly guessing")

How many are you left with?
How many were the right call?
How many got rejected?
Low trust + approvals = illusion of control

Add up the total number of approval requests and

...subtract the info radiators ("I need to be in the loop")

...subtract the CYAs ("Prove you followed the process")

...subtract the too removed to judge ("mostly guessing")

How many are you left with?
How many were the right call?
How many got rejected?
Low Trust Undermines SRE Principles

1. Org has Service Level Objectives, with consequences?

2. SREs have time to make tomorrow better than today?

3. SRE teams have the ability to regulate their workload?
Low Trust Undermines SRE Principles

1. Org has Service Level Objectives, with consequences?
   - Cultures of low trust have a really difficult time with shared responsibility

2. SREs have time to make tomorrow better than today?

3. SRE teams have the ability to regulate their workload?
Low Trust Undermines SRE Principles

1. Org has Service Level Objectives, with consequences?
   - Cultures of low trust have a really difficult time with shared responsibility

2. SREs have time to make tomorrow better than today?
   - People closest to problems know what to fix but tasking, priorities, and decisions are largely out of their control

3. SRE teams have the ability to regulate their workload?
Low Trust Undermines SRE Principles

1. Org has Service Level Objectives, with consequences?
   - 🚫 Cultures of low trust have a really difficult time with shared responsibility

2. SREs have time to make tomorrow better than today?
   - 🚫 People closest to problems know what to fix but tasking, priorities, and decisions are largely out of their control

3. SRE teams have the ability to regulate their workload?
   - 🚫 People aren’t trusted to plan or design their own work
Forces That Undermine Operations

Silos

Queues

Toil

Low Trust
So what can we do differently?
Lean on Lean to find what to fix

1. Map the end-to-end flow of information and artifacts (using a recent delivery or event)

2. Identify what slows lead times, undermines quality, and impacts flow

3. Identify countermeasures and create improvement storyboards (justification/plan)
Lean on Lean to find what to fix

1. Map the end-to-end flow of information and artifacts (using a recent delivery or event)

2. Identify what slows lead times, undermines quality, and impacts flow

3. Identify countermeasures and create improvement storyboards (justification/plan)

All processes should be studied with an improvement disciple
Lean on Lean to find what to fix

1. Map the end-to-end flow of information and artifacts (using a recent delivery or event)

2. Identify what slows lead times, undermines quality, and impacts flow

3. Identify countermeasures and create improvement storyboards (justification/plan)

- All processes should be studied with an improvement disciple
- Incidents are just as much a "process" as delivery
Lean on Lean to find what to fix

1. Map the end-to-end flow of information and artifacts (using a recent delivery or event)

2. Identify what slows lead times, undermines quality, and impacts flow

3. Identify countermeasures and create improvement storyboards (justification/plan)

- All processes should be studied with an improvement disciple
- Incidents are just as much a “process” as delivery
- Look to Lean for proven improvement techniques (value stream mapping, waste analysis, improvement kata)
Lean on Lean to find what to fix

- Map the end-to-end flow of information and artifacts (using a recent delivery or event)

- Identify what slows lead times, undermines quality, and impacts flow

- Identify countermeasures and create improvement storyboards (justification/plan)

- All processes should be studied with an improvement disciple

- Incidents are just as much a “process” as delivery

- Look to Lean for proven improvement techniques (value stream mapping, waste analysis, improvement kata)

- Make it a part of your organization’s discipline
Get rid of as many silos as possible

Old Silo A  Old Silo B  Old Silo C  Old Silo D
Get rid of as many silos as possible
Get rid of as many silos as possible

Key 1: get rid of as many handoffs as possible
Get rid of as many silos as possible

Key 1: get rid of as many handoffs as possible

Key 2: “Horizontal” shared responsibility, *not* everyone do everything!
Shared responsibility matters more than org model

“Netflix" Model

Cross-Functional Team 1

Cross-Functional Team 2

Cross-Functional Team n
Shared responsibility matters more than org model

"Netflix" Model

Cross-Functional Team 1
Cross-Functional Team 2
Cross-Functional Team n

"Google" Model

Development Team 1
Development Team 2
Development Team n
SRE Team

Clear handoff requirements
Error budget consequences
Shared responsibility matters more than org model

"Netflix" Model

- Cross-Functional Team 1
- Cross-Functional Team 2
- Cross-Functional Team n

Same high-quality, high-velocity results!

"Google" Model

- Development Team 1
- Development Team 2
- Development Team n
- SRE Team

Clear handoff requirements

Error budget consequences
Why focus on getting rid of handoffs?
Why focus on getting rid of handoffs?

1. Your people are your most valuable assets
Why focus on getting rid of handoffs?

1. Your people are your most valuable assets
2. The SRE skillset is expensive
Why focus on getting rid of handoffs?

1. Your people are your most valuable assets
2. The SRE skillset is expensive
3. Stay out of their way!
SREs are expensive, stay out of their way!

Not this:

This:
SREs are expensive, stay out of their way!

Not this:

This:

Reduce friction:
SREs are expensive, stay out of their way!

Not this:

This:

Reduce friction:

Invest in the right instrumentation

SRE OODA Loop

Observe Orient Decide Action
SREs are expensive, stay out of their way!

Not this:

- Backlog
  - Ticket Queue
  - Ticket Queue
  - Ticket Queue
  - Ticket Queue
  - Ticket Queue

This:

- Backlog
  - Ticket Queue

Reduce friction:

- Invest in the right instrumentation
- Invest in collaboration, checklists, investigatory tools

**OODA Loop**

- Observe
- Orient
- Decide
- Action
SREs are expensive, stay out of their way!

Not this:

This:

Reduce friction:

- Invest in the right instrumentation
- Invest in collaboration, checklists, investigatory tools
- Empower them to make decisions!
SREs are expensive, stay out of their way!

Not this:

This:

Reduce friction:

Invest in the right instrumentation

Empower them to take action!

Empower them to make decisions!

Invest in collaboration, checklists, investigatory tools

Orient

Observe

SRE OODA Loop

Decide

Action
What about the handoffs you can’t get rid of?

Old Silo A  Old Silo B  Old Silo C  Old Silo D

Cross-Functional Team 1

Cross-Functional Team 2

Cross-Functional Team $n$

Specialist Capabilities

Specialist Capabilities

Specialist Capabilities
What about the handoffs you can’t get rid of?

Old Silo A  Old Silo B  Old Silo C  Old Silo D

Cross-Functional Team 1

Cross-Functional Team 2

Cross-Functional Team $n$

Ticket Queue  Ticket Queue  Ticket Queue

Specialist Capabilities  Specialist Capabilities  Specialist Capabilities
What about the handoffs you can’t get rid of?
Operations as a Service: Turn handoffs into self-service

Cross-Functional Product Team 1
Ops (embedded)

Cross-Functional Product Team 2
Ops (embedded)

Cross-Functional Product Team n
Ops (embedded)

Operations as a Service
Ops (builds & operates)

Ops Capability
SRE, Dev, or Specialist

Ops Capability
SRE, Dev, or Specialist

Ops Capability
SRE, Dev, or Specialist
Operations as a Service: Works with any org model

Development Team 1
Development Team 2
Development Team n

Ops/SRE Team

Operations as a Service

Ops Capability
SRE, Dev, or Specialist

Ops Capability
SRE, Dev, or Specialist

Ops Capability
SRE, Dev, or Specialist

Ops (builds & operates)
Operations as a Service: Popular Uses for SRE

"I could fix it, if I could get to it"
Operations as a Service: Popular Uses for SRE

"I could fix it, if I could get to it"
Operations as a Service: Popular Uses for SRE

“Avoiding the dogpile”

I think it's a problem with dbcluster07-store2.uswest.acme
Operations as a Service: Popular Uses for SRE

“Avoiding the dogpile”

I think it's a problem with dbcluster07-store2.uswest.acme

"$ top"
“I don’t read wikis. I’m an expert.”

Service has changed. This flag is now required or bad things will happen!

Pause monitoring first or we all get woken up!

Later...

I’ve done this before. I’ve got this.

“restart -doit -now”

Environment
Operations as a Service: Popular Uses for SRE

“I don’t read wikis. I’m an expert.”

Service has changed. This flag is now required or bad things will happen!

Pause monitoring first or we all get woken up!

Later...

I've done this before. I've got this.

“restart -doit -now”

Environment

Later...

Update Restart Job

I've done this before. I've got this.

“restart”

OaaS

Environment
Operations as a Service: Popular Uses for SRE

“Uneven and hidden skills”

1. I don’t know how to do X.
2. I know how to do X.
3. I know how to do Y.
4. I don’t know how to do Y.
Operations as a Service: Popular Uses for SRE

“Uneven and hidden skills”

I don’t know how to do X.
I know how to do X.
I don’t know how to do Y.
I know how to do Y.

“Do X”
“Define X Procedure”

“Do Y”
“Define Y Procedure”

“Do X+Y”
Operations as a Service: Popular Uses for SRE

“Let me do that for you again… and again”
Operations as a Service: Popular Uses for SRE

“Let me do that for you again… and again”
Use tickets only for what they are good for
Use tickets only for what they are good for

1. Documenting true problems/issue/exceptions
Use tickets only for what they are good for

1. Documenting true problems/issues/exceptions
2. Routing for necessary approvals
Use tickets only for what they are good for

1. Documenting true problems/issues/exceptions
2. Routing for necessary approvals

Not as a general purpose work management system!
But won’t Security or Compliance stop you?

Operations as a Service

Cross-Functional Product Team 1

Ops (embedded)

Cross-Functional Product Team 2

Ops (embedded)

Cross-Functional Product Team n

Ops (embedded)

Ops Capability

SRE, Dev, or Specialist

Ops Capability

SRE, Dev, or Specialist

Ops Capability

SRE, Dev, or Specialist

On Demand

On Demand

On Demand

On Demand

Build-in Security Here

Build-in Compliance Here

Operations as a Service

Ops (builds & operates)
But what about ITIL®?
But what about ITIL®?

- Ask ITIL people and they say SRE is ITIL compatible
But what about ITIL®?

• Ask ITIL people and they say SRE is ITIL compatible

• Ask people who have seen ITIL implemented and they say “how?”
But what about ITIL®?

• Ask ITIL people and they say SRE is ITIL compatible

• Ask people who have seen ITIL implemented and they say “how?”

• Agile+DevOps+SRE have self-regulation and shared responsibility features that seem to undermine ITIL command and control nature
But what about ITIL®?

• Ask ITIL people and they say SRE is ITIL compatible

• Ask people who have seen ITIL implemented and they say “how?”

• Agile+DevOps+SRE have self-regulation and shared responsibility features that seem to undermine ITIL command and control nature

• ITIL “Standard Change” is often focus of discussion, but it still implies approval model
But what about ITIL®?

- Ask ITIL people and they say SRE is ITIL compatible
- Ask people who have seen ITIL implemented and they say “how?”
- Agile+DevOps+SRE have self-regulation and shared responsibility features that seem to undermine ITIL command and control nature
- ITIL “Standard Change” is often focus of discussion, but it still implies approval model
- Straight talk: are we doing contortions to defend a sunk cost?
“Shift Left” the ability to take action
“Shift Left” the ability to take action

Push the ability to take action this direction

or

1°  2°  3°  4°
“Shift Left” the ability to take action

Push the ability to take action this direction

or

1° escalate
2° escalate
3° escalate
4°

OaaS

Enablement and tooling
Reduce Toil
Reduce Toil

1. Track toil levels for each team
Reduce Toil

1. Track toil levels for each team

2. Set toil limits for each team
Reduce Toil

1. Track toil levels for each team

2. Set toil limits for each team

3. Fund efforts to reduce toil (with emphasis on teams over toil limits)
Start a book club
Recap

**SRE is more than a title**

Leverage the Operations as a Service design pattern

"Shift-Left" control and decision making.

Focus on removing silos and queues

Leverage the Operations as a Service pattern

Understand the forces undermining SRE

Focus on removing silos and queues

Reduce toil to create capacity to change

---

**ITIL Book 1**

**ITIL Book 2**

**ITIL Book 3**

**ITIL Book 4**

**ITIL Book 5**

Quality! is job #1

---

Sys

Admin

CAB CALENDAR

Your new title is SRE.
Now write code and be better at ops.

---

Dilbert characters © Scott Adams  www.dilbert.com
Let's talk…

@damonedwards
damon@rundeck.com
@damonedwards

Dive Deeper Into Operations as a Service:
https://www.rundeck.com/oaas