DevOps in a Chaotic Situation

“A practical guide for when everything is on fire, and why it’s all going to be OK”

Dan Jandreau & Ben Hamilton, 2018
Agenda

- Introductions
- What is DevOps?
- The flavors of DevOps
- Story Time
- Takeaways
> whoami

Dan Jandreau
- IDEXX since 2015
- Intern turned DevOps
- Likes serverless architecture a little too much
- Twitter: @_dandro_
> whoami

Ben Hamilton
- IDEXX since 2013
- DevSecOps
- Just wants you to stop putting private ssh keys in Github
- Twitter: @possiblyben
What is DevOps?
What is DevOps?

- A fancy word managers throw around?
- Something you put on your LinkedIn to look fancy?
- A person who knows *gasp* “the cloud”? 
- Tech buzzword?
What is DevOps?

- A fancy word managers throw around?
- Something you put on your LinkedIn to look fancy?
- A person who knows *gasp* “the cloud”?
- Tech buzzword?
- A mindset
- A partner
- A teammate
- An expert
What is DevOps?
What is DevOps?
What is DevOps?
What is DevOps?

• Software Development
What is DevOps?

- Software Development + IT Professionals
What is DevOps?

• Software Development + IT Professionals

• Automating the SDLC
What is DevOps?

- Software Development + IT Professionals
- Automating the SDLC
- Infrastructure as Code
The Flavors of DevOps (explained in GIFs)
The Flavors of DevOps (explained in GIFS)

The Siloed Expert

Um, we have sort of a problem here.
The Flavors of DevOps (explained in GIFS)

*The Integrated Expert*

THAT'S A BOLD STRATEGY COTTON
LET'S SEE IF IT PAYS OFF FOR HIM
The Flavors of DevOps (explained in GIFS)

The Team

TEAMWORK MAKES THE DREAMWORK
Welcome to DevOps.
Here’s your Fire Extinguisher
A story: IDEXX Communicator
A Use Case

- The Product:
  - Direct messaging between a Veterinary Clinic and its patients
A Use Case

- The Product:
  - Direct messaging between a Veterinary Clinic and its patients

- The Goal:
  - Create a lightweight responsive javascript web application
  - Small feedback loops
  - Fast delivery
A Use Case

- The Product:
  - Direct messaging between a Veterinary Clinic and its patients
- The Goal:
  - Create a lightweight responsive JavaScript web application
  - Small feedback loops
  - Fast delivery
- The Constraints:
  - Kanban methodology
  - Rapid Iteration / Rapid Development
  - Deep integration with two other IDEXX development teams
A Use Case

What this sounds like

FUN, FUN, FUN!
A Use Case

What it’s actually like
The Challenges
Speed vs. Quality

- What I learned:
  - Throw your best practices out the window
  - Get ready for not even plans A or B, but plans C and D
  - All of it is going to be bad. All of it.
  - It’s OK to write subpar code
Speed vs. Quality

- What I learned:
  - Throw your best practices out the window
  - Get ready for not even plans A or B, but plans C and D
  - All of it is going to be bad. All of it.
  - It’s OK to write subpar code

- Example:
  - Non-immutable infrastructure to create environments
  - Involved build / deploy process
  - Re-architecting about once a week
Speed vs. Quality

- What I learned:
  - Throw your best practices out the window
  - Get ready for not even plans A or B, but plans C and D
  - All of it is going to be bad. All of it.
  - It’s OK to write subpar code

- Example:
  - Non-immutable infrastructure to create environments
  - Involved build / deploy process
  - Re-architecting about once a week

- Solutions:
  - Documentation is your friend
  - Make your decisions knowing the future
Changing many processes at once

- What I learned:
  - Experienced Devs + Untrained languages = Friction
  - Communication is key
  - Mapping ideas to previous workflows is ok to baseline
  - Things may feel slow, but they’re actually moving quickly
Changing many processes at once

- What I learned:
  - Experienced Devs + Untrained languages = Friction
  - Communication is key
  - Mapping ideas to previous workflows is ok to baseline
  - Things may feel slow, but they’re actually moving quickly

- Example:
  - Java to Javascript
  - Agile to Kanban
  - Integrated Devops
Changing many processes at once

- **What I learned:**
  - Experienced Devs + Untrained languages = Friction
  - Communication is key
  - Mapping ideas to previous workflows is ok to baseline
  - Things may feel slow, but they’re actually moving quickly

- **Example:**
  - Java to Javascript
  - Agile to Kanban
  - Integrated Devops

- **Solutions:**
  - Listen to your Devs, work on compromise
  - Work closely with product
  - Always keep yourself in check
Multiple team alignment

- What I learned:
  - Assumptions are bad
  - Your way isn’t the best way
  - Alignment helps give clarity
Multiple team alignment

- **What I learned:**
  - Assumptions are bad
  - Your way isn’t the best way
  - Alignment helps give clarity

- **Example:**
  - Needing data from API’s that aren’t available yet
  - Similar but yet different implementations
Multiple team alignment

- What I learned:
  - Assumptions are bad
  - Your way isn’t the best way
  - Alignment helps give clarity

- Example:
  - Needing data from API’s that aren’t available yet
  - Similar but yet different implementations

- Solutions:
  - Twice a week check in
  - Work with leadership on needs
  - Always have alternate solutions
Rapid Iteration / Rapid Development

- What I learned:
  - Small feedback loops are important
  - Company firsts do not allow you to leverage domain knowledge
  - DevOps can be included in this!

- Examples:
  - Next day backlog stories
  - DevOps being involved very early on in the process
  - Iteration of architecture

- Solutions:
  - Create a close relationship with your product owner
  - Work with your dev manager / tech lead on an appropriate time for DevOps to be involved
  - Be ok with scrapping everything
Rapid Iteration / Rapid Development

- What I learned:
  - Small feedback loops are important
  - Company firsts do not allow you to leverage domain knowledge
  - DevOps can be included in this!
Rapid Iteration / Rapid Development

- What I learned:
  - Small feedback loops are important
  - Company firsts do not allow you to leverage domain knowledge
  - DevOps can be included in this!

- Examples:
  - Next day backlog stories
  - DevOps being involved very early on in the process
  - Iteration of architecture
Rapid Iteration / Rapid Development

- **What I learned:**
  - Small feedback loops are important
  - Company firsts do not allow you to leverage domain knowledge
  - DevOps can be included in this!

- **Examples:**
  - Next day backlog stories
  - DevOps being involved very early on in the process
  - Iteration of architecture

- **Solutions:**
  - Create a close relationship with your product owner
  - Work with your dev manager / tech lead on an appropriate time for DevOps to be involved
  - Be ok with scrapping everything
Conclusions
Conclusions

1. Fires will inevitably happen.
Conclusions

1. Fires will inevitably happen.
2. The closer you are with your developers and product, the happier you will be.
Conclusions

1. Fires will inevitably happen.
2. The closer you are with your developers and product, the happier you will be.
3. You will never get it right the first try.
Conclusions

1. Fires will inevitably happen.
2. The closer you are with your developers and product, the happier you will be.
3. You will never get it right the first try.
4. Iteration leads to innovation.
Thanks Everyone!

LinkedIn:
https://www.linkedin.com/in/danieljandreau
https://www.linkedin.com/in/hamiltonbenjamin/

Email:
Dan-Jandreau@idexx.com
Benjamin-Hamilton@idexx.com

Web: https://careers.idexx.com/
IDEXX is green
Please project (don’t print) this presentation.