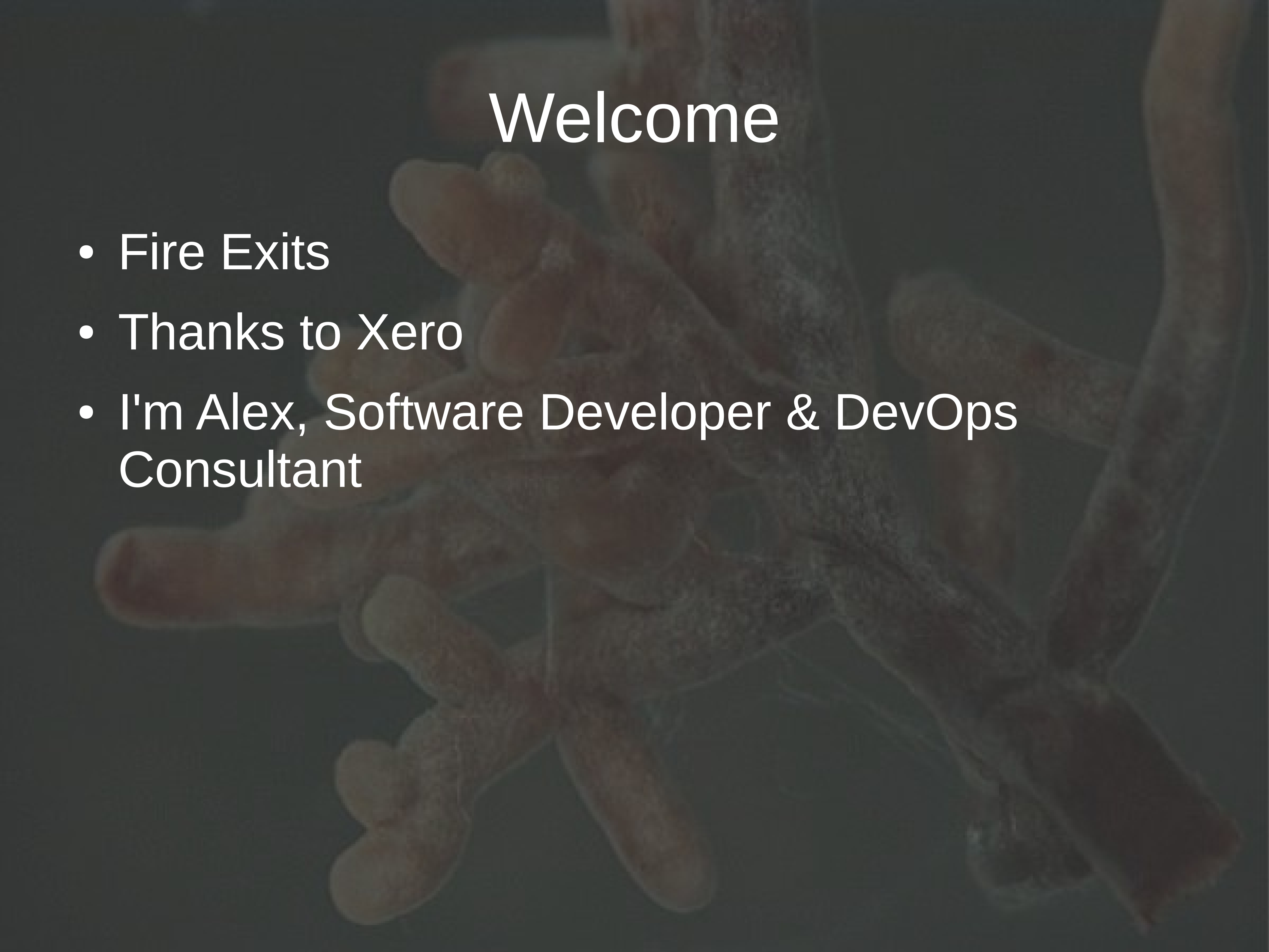
What is DevOps Anyway?

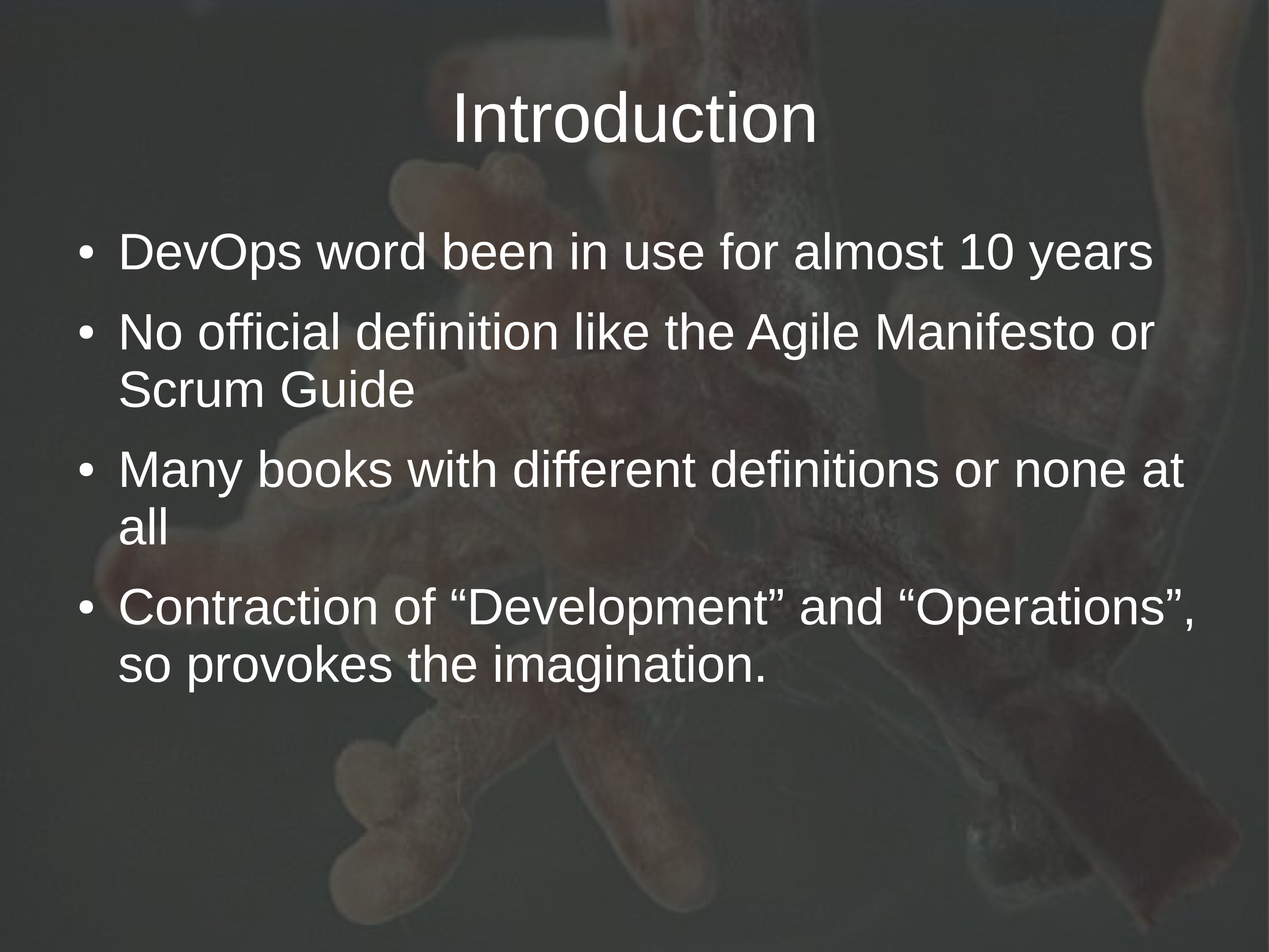
Alexander Taler
alex@deref.co.nz
Auckland, 2017-11-20

Welcome



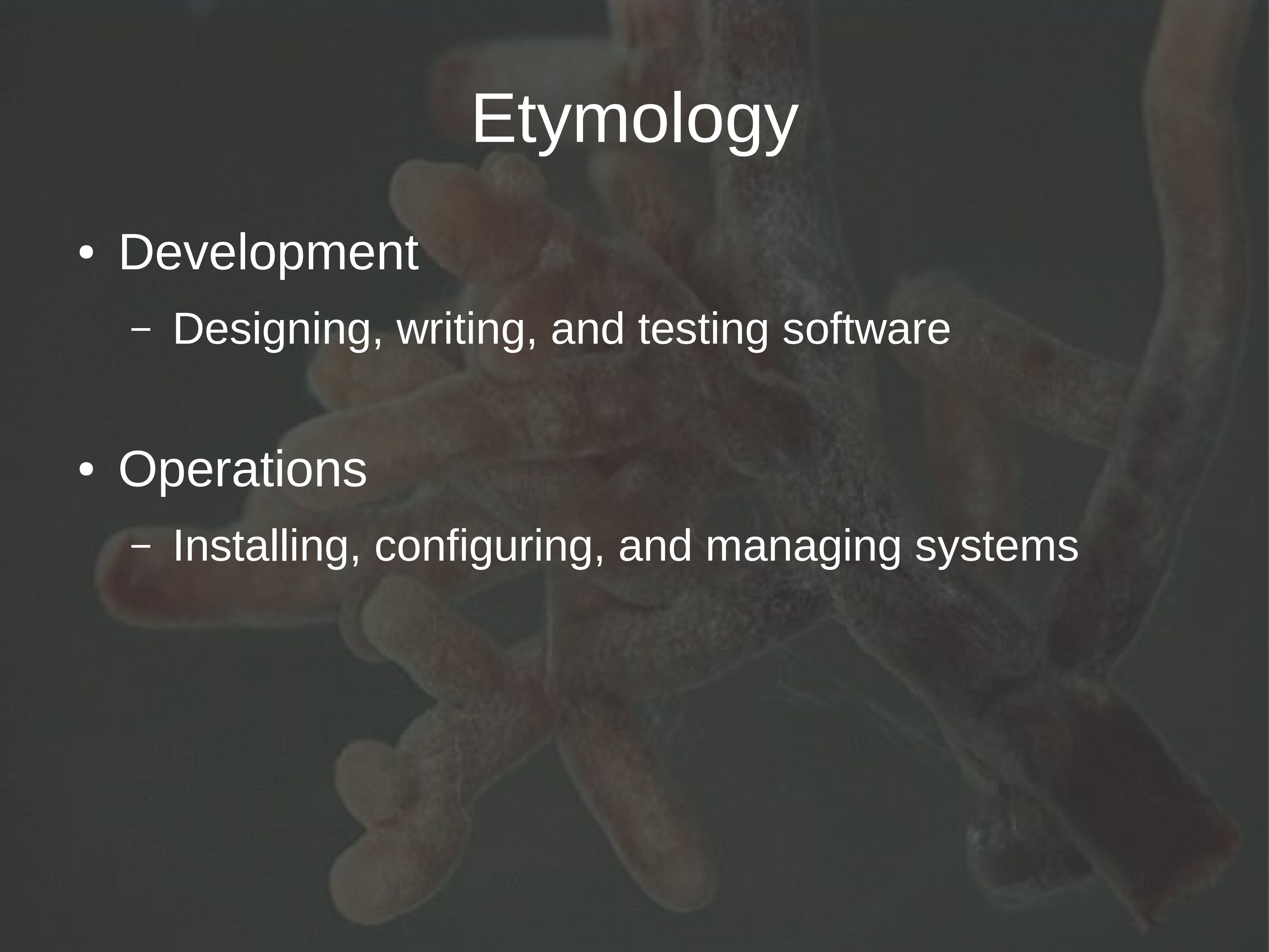
- Fire Exits
- Thanks to Xero
- I'm Alex, Software Developer & DevOps Consultant

DevOps Definitions



Introduction

- DevOps word been in use for almost 10 years
- No official definition like the Agile Manifesto or Scrum Guide
- Many books with different definitions or none at all
- Contraction of “Development” and “Operations”, so provokes the imagination.

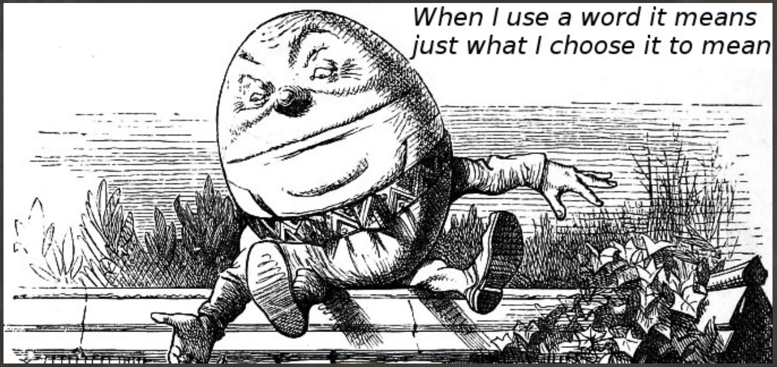


Etymology

- Development
  - Designing, writing, and testing software

- Operations
  - Installing, configuring, and managing systems

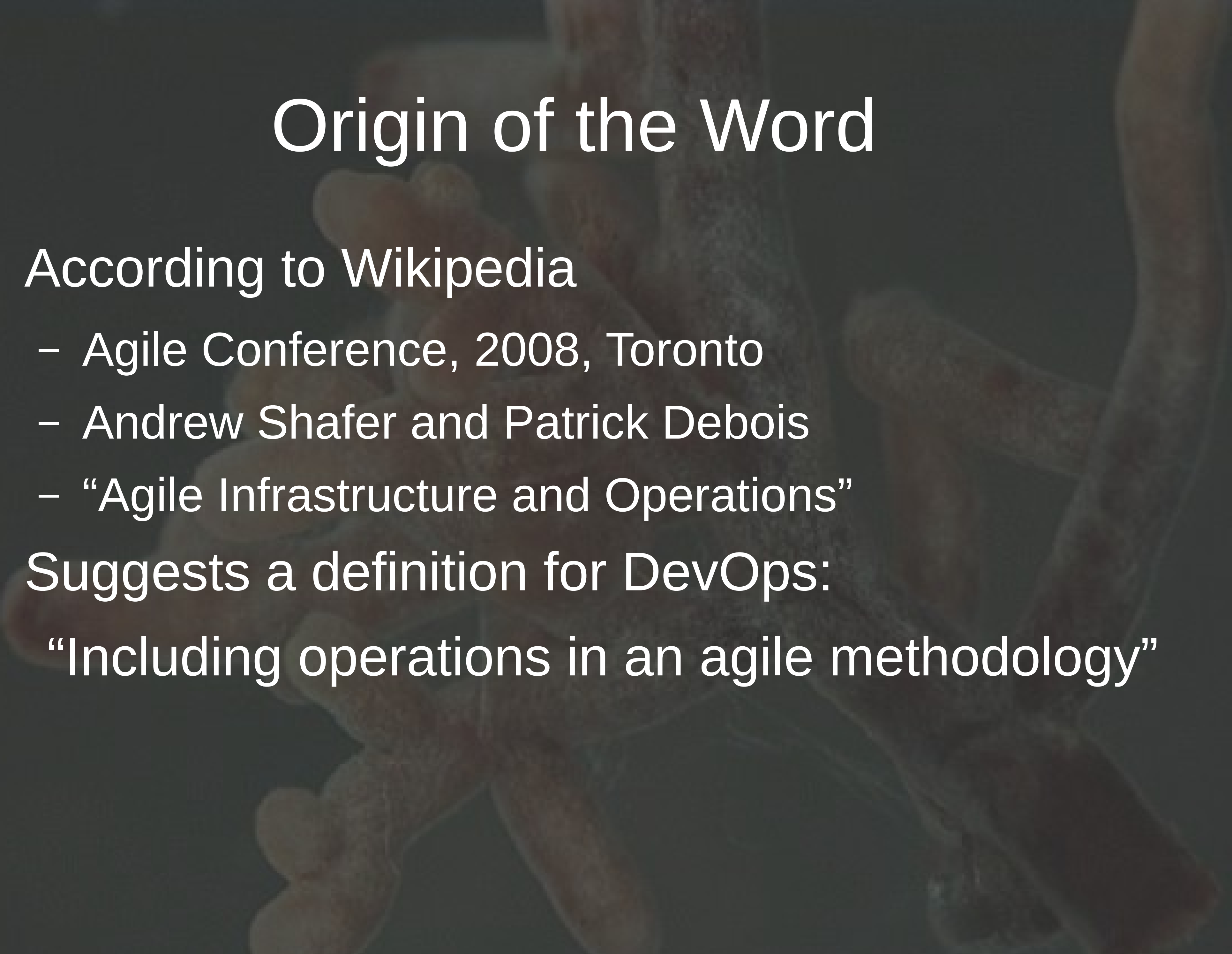
How is “DevOps” Used?



- Commonly:
  - “DevOps Engineer”
  - “DevOps Team”
  - “We do DevOps”
  - “DevOps Tools”

- Literature
  - “DevOps Ethics”
  - “DevOps Principles”
  - “DevOps Practices”
  - “DevOps Behaviours”
  - “DevOps Outcomes”

Origin of the Word

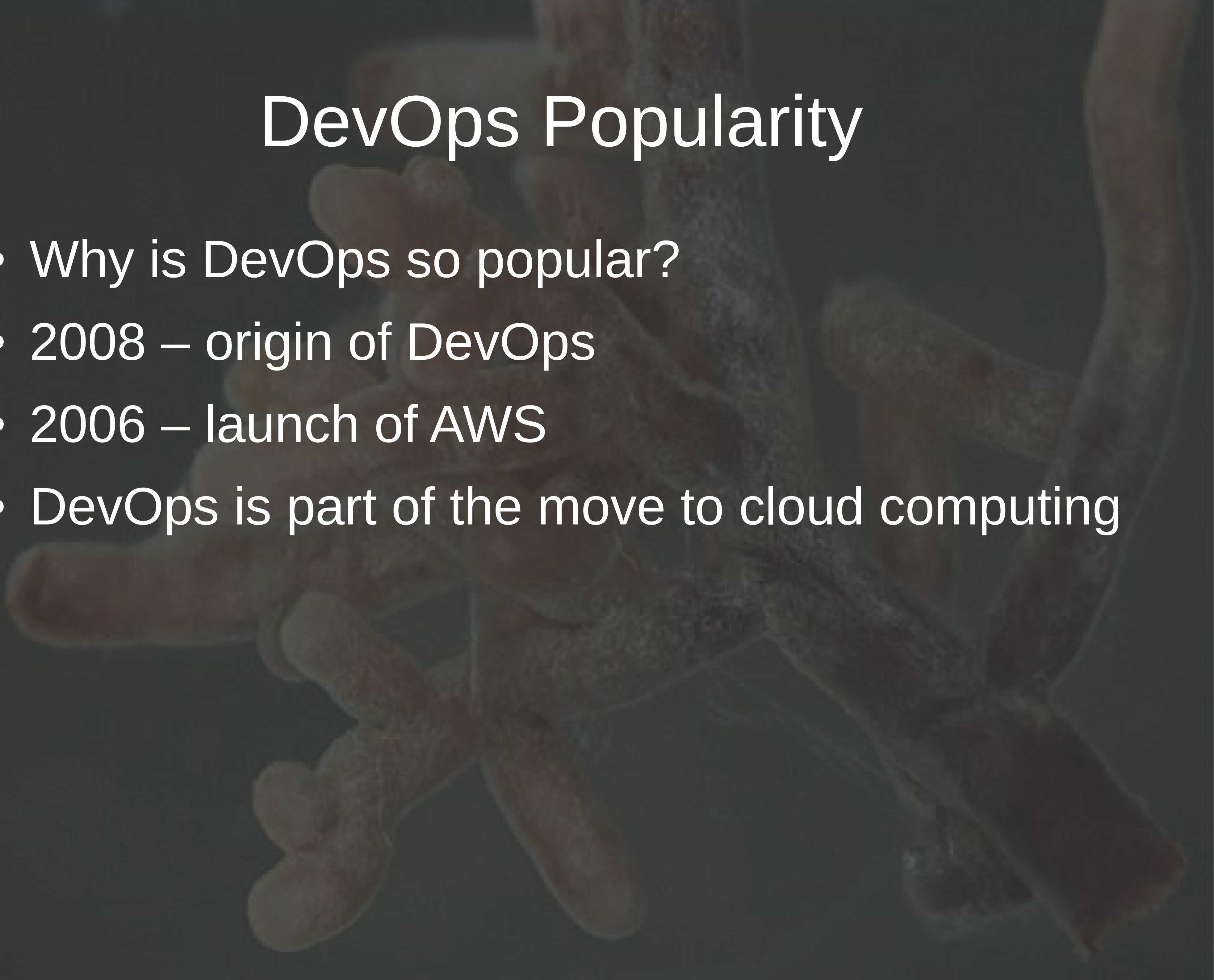


- According to Wikipedia
  - Agile Conference, 2008, Toronto
  - Andrew Shafer and Patrick Debois
  - “Agile Infrastructure and Operations”
- Suggests a definition for DevOps:

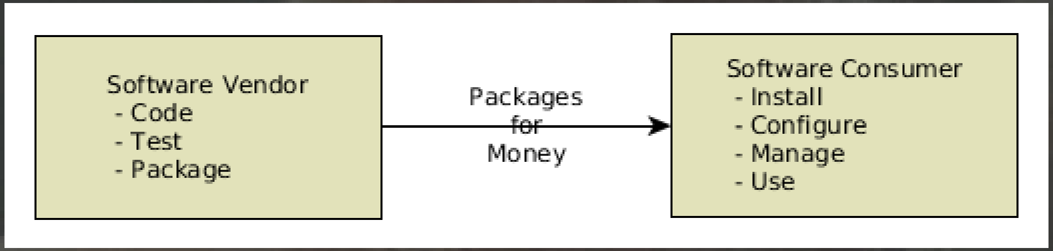
  “Including operations in an agile methodology”

DevOps Popularity

- Why is DevOps so popular?
- 2008 – origin of DevOps
- 2006 – launch of AWS
- DevOps is part of the move to cloud computing



Before Cloud



Software Vendor
- Code
- Test
- Package

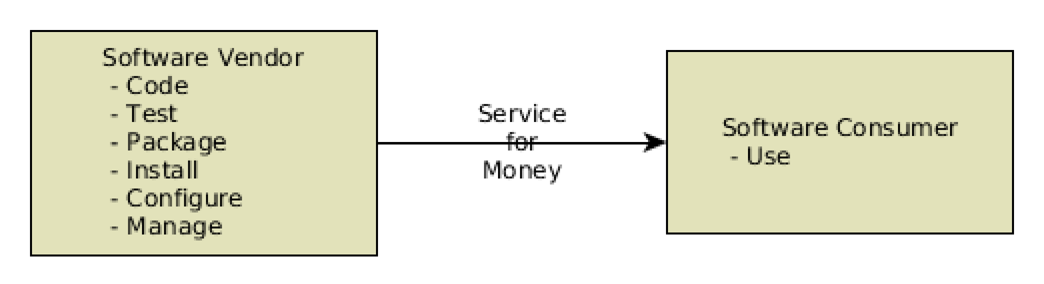
Packages for Money

Software Consumer
- Install
- Configure
- Manage
- Use

- Development and Operations were separated
  - Financially
  - Networkologically
  - Geographically
  - Legally
  - Culturally

After Cloud

- SaaS is the Norm



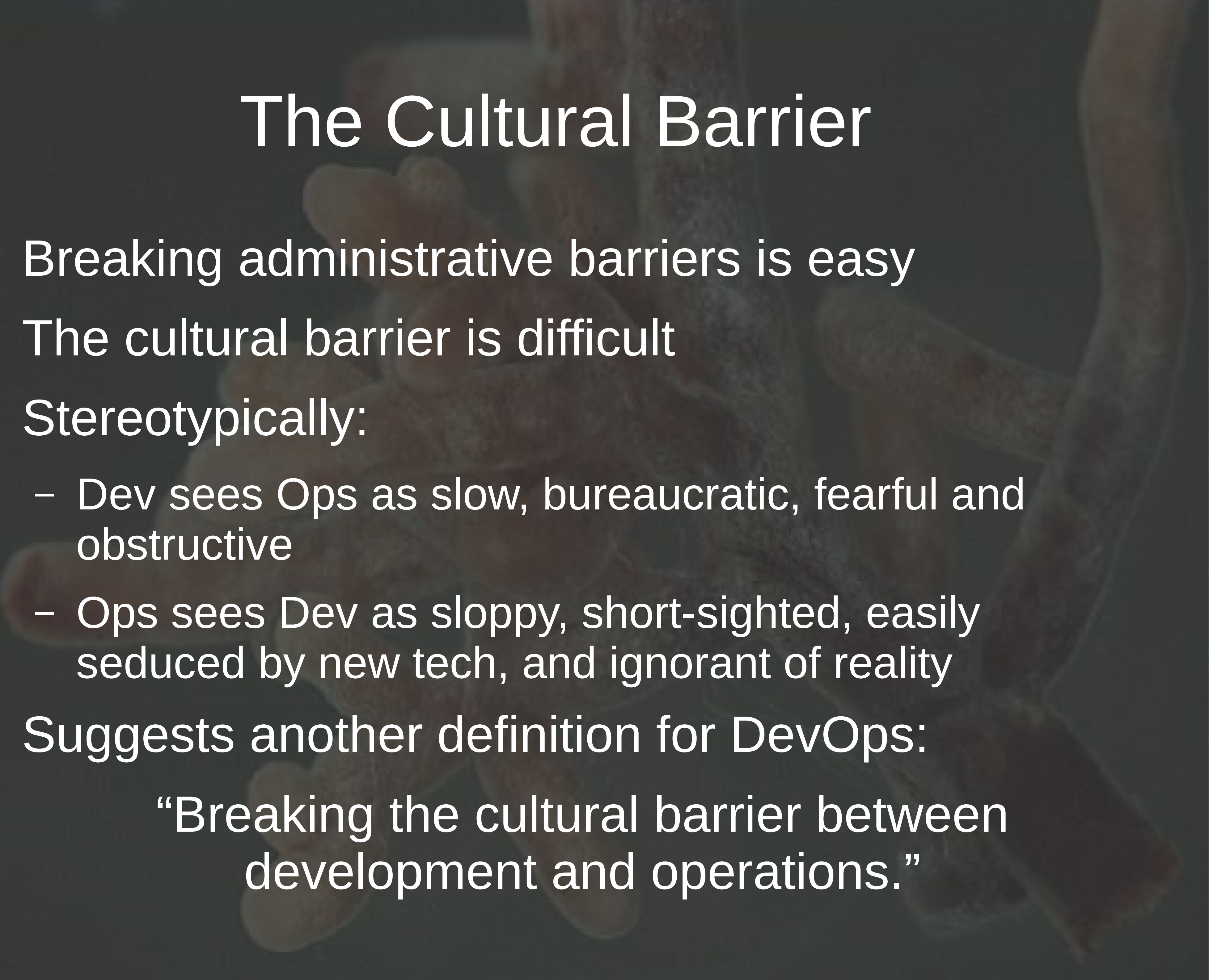
Software Vendor
- Code
- Test
- Package
- Install
- Configure
- Manage

Service for Money

Software Consumer
- Use

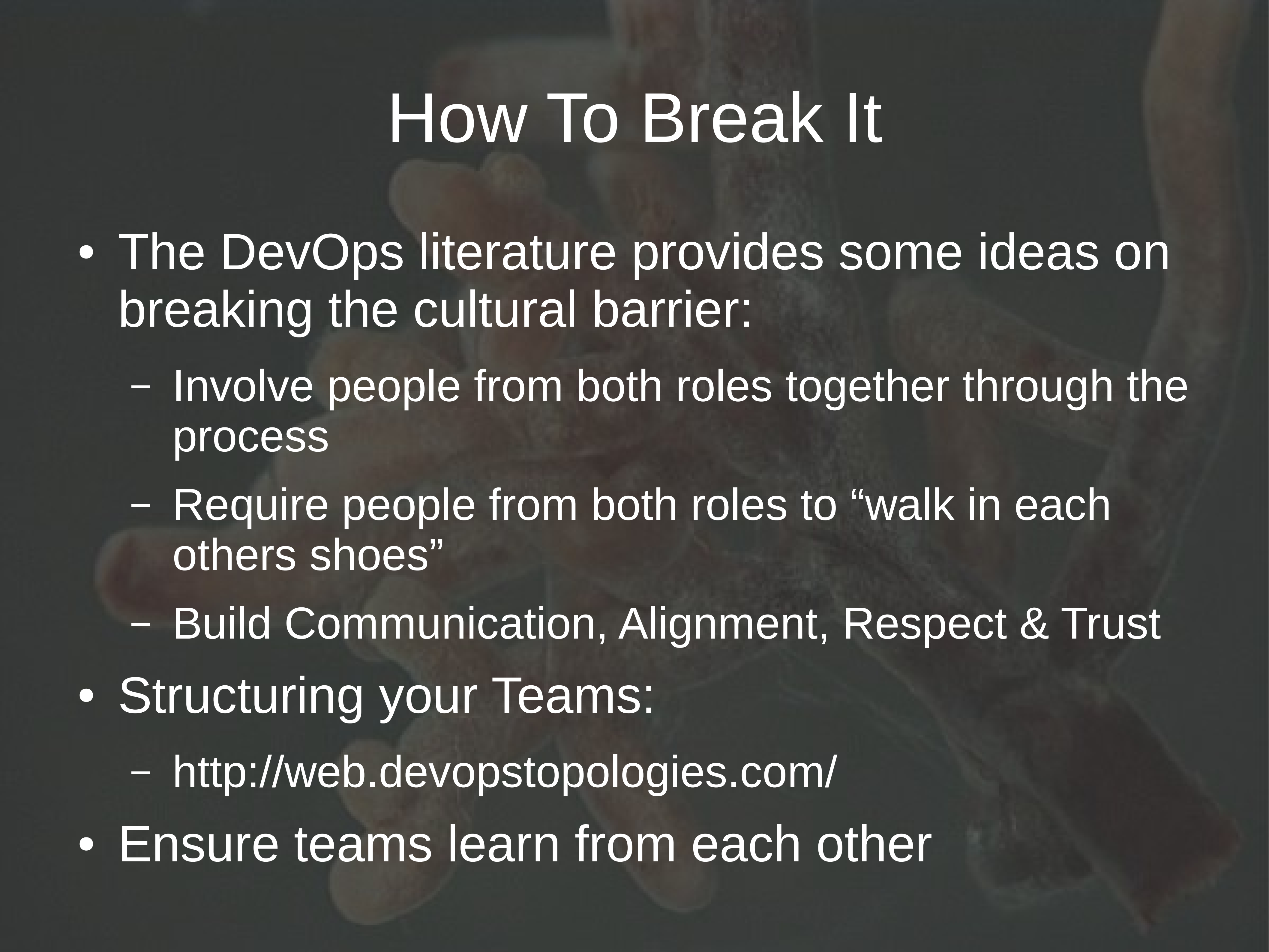
- Barriers between Development and Operations are broken
- Except, the cultural barrier

The Cultural Barrier



- Breaking administrative barriers is easy
- The cultural barrier is difficult
- Stereotypically:
  - Dev sees Ops as slow, bureaucratic, fearful and obstructive
  - Ops sees Dev as sloppy, short-sighted, easily seduced by new tech, and ignorant of reality
- Suggests another definition for DevOps:

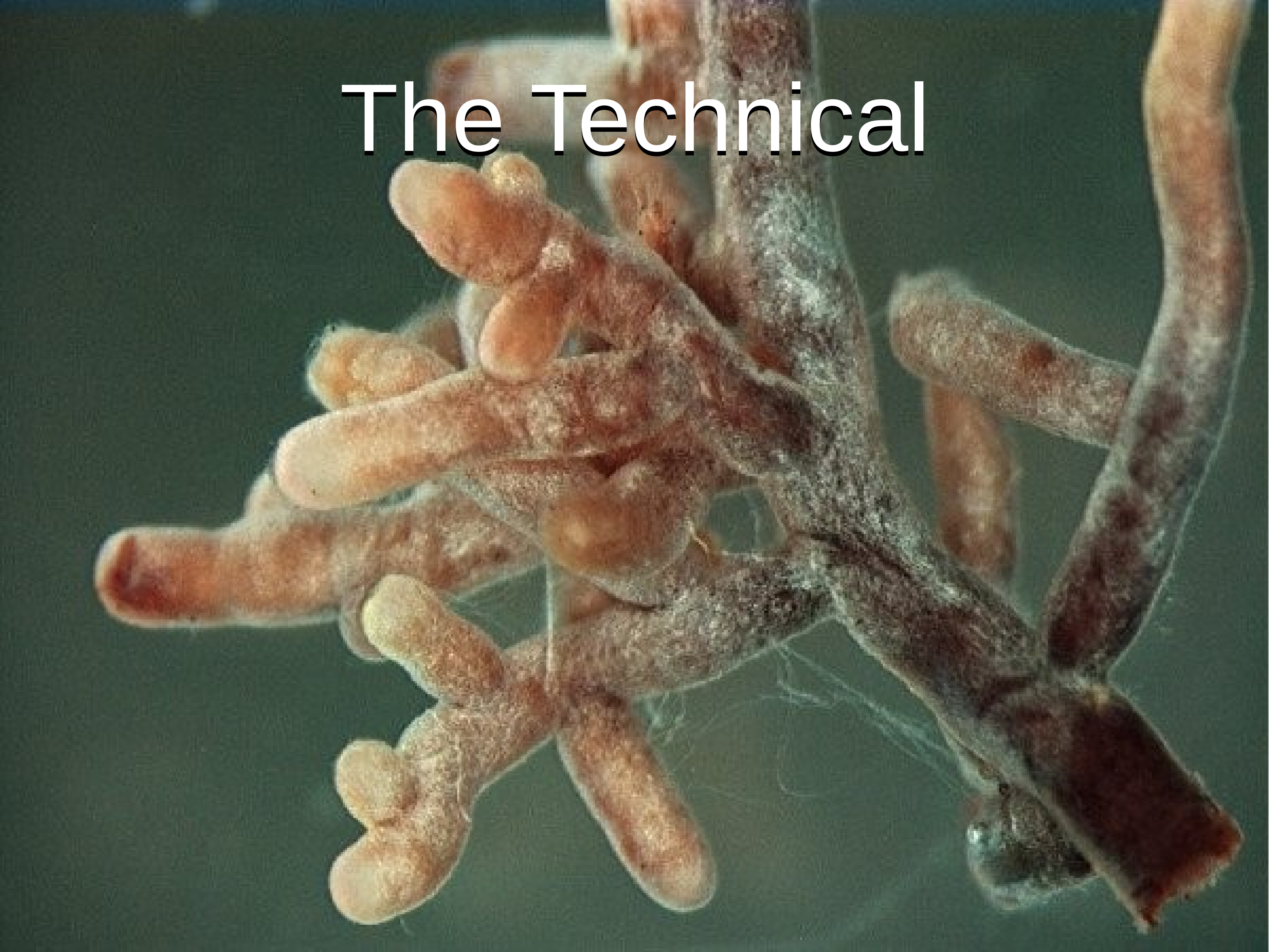
“Breaking the cultural barrier between development and operations.”



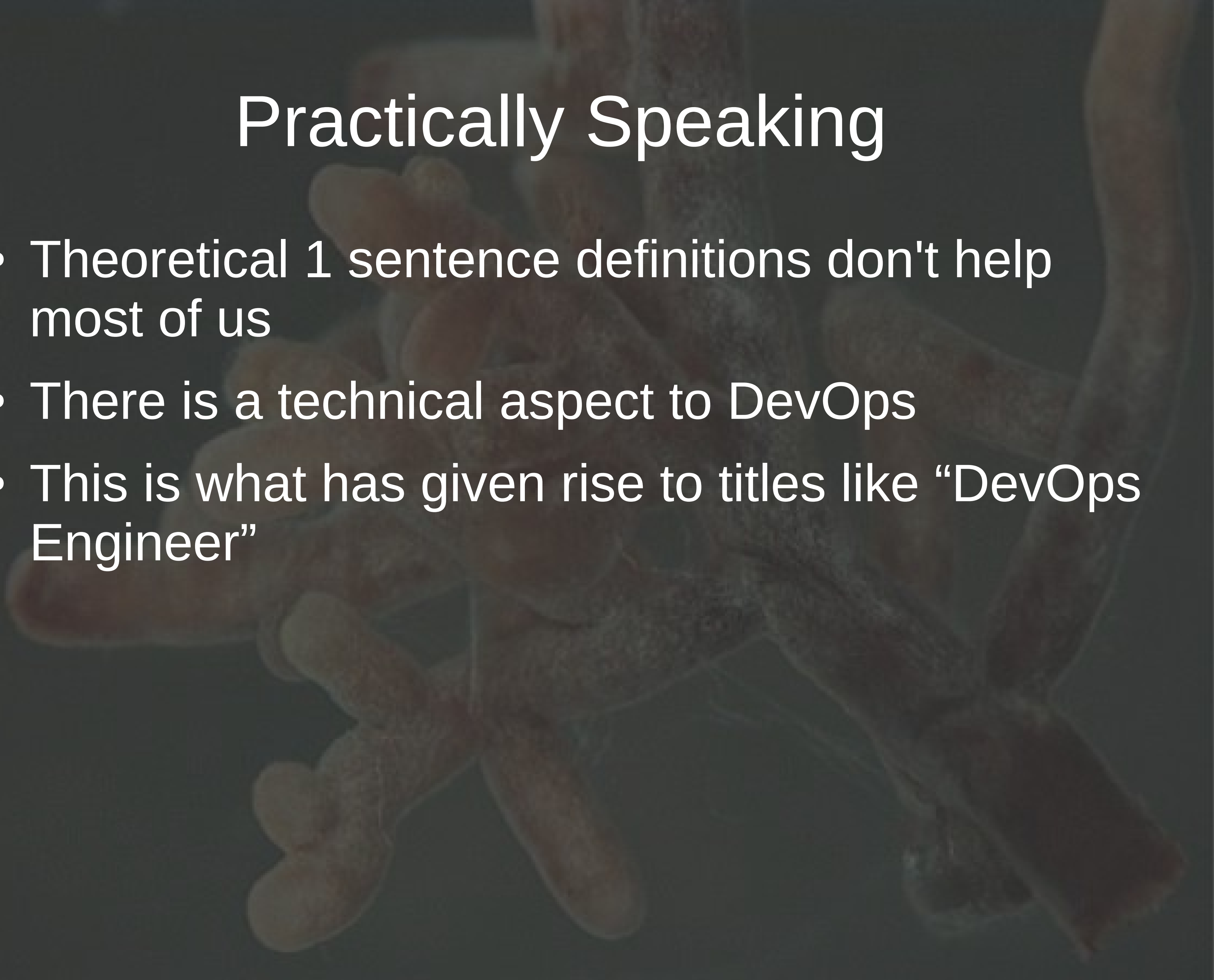
How To Break It

- The DevOps literature provides some ideas on breaking the cultural barrier:
  - Involve people from both roles together through the process
  - Require people from both roles to “walk in each others shoes”
  - Build Communication, Alignment, Respect & Trust
- Structuring your Teams:
  - http://web.devopstopologies.com/
- Ensure teams learn from each other

The Technical

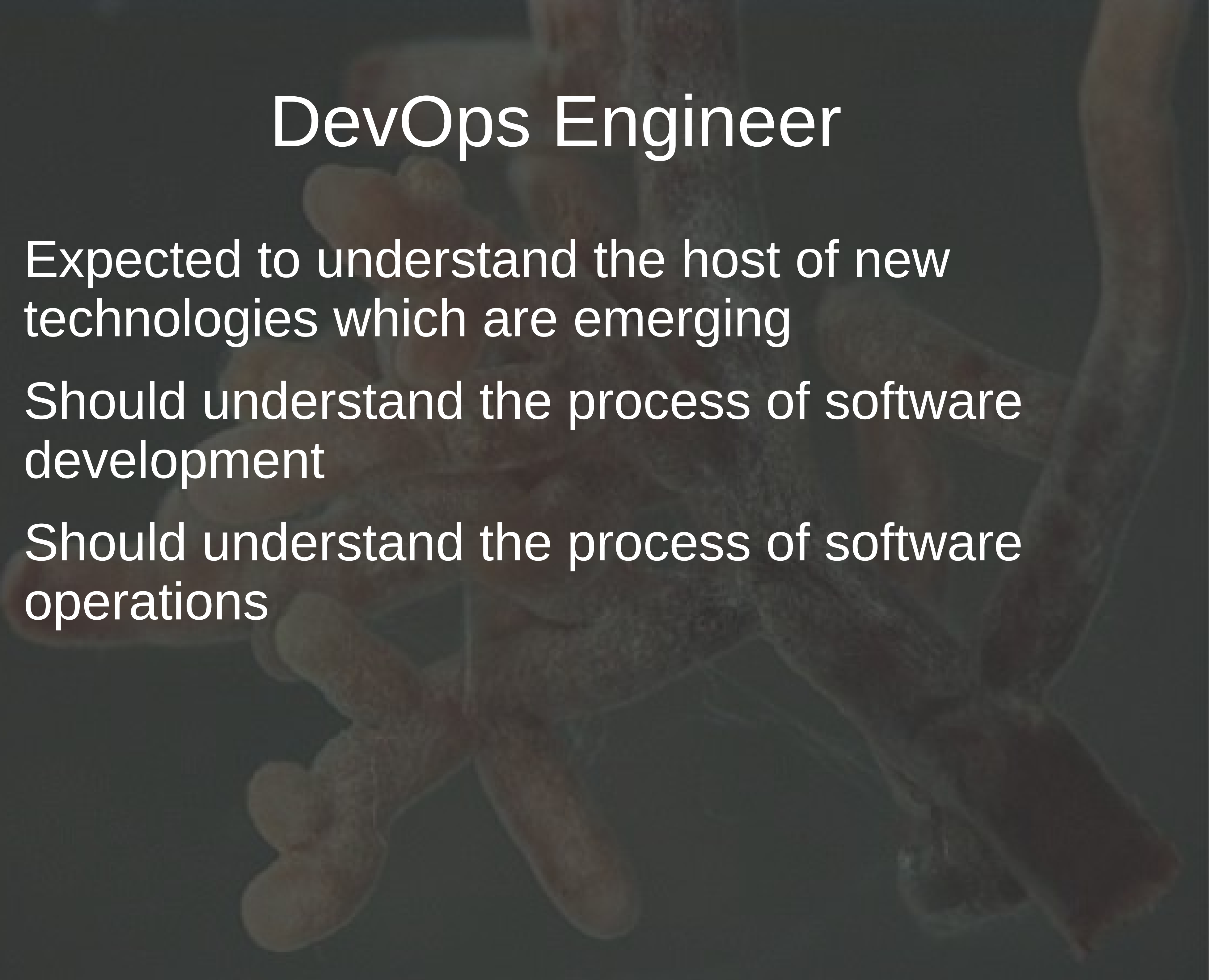


Practically Speaking



- Theoretical 1 sentence definitions don't help most of us
- There is a technical aspect to DevOps
- This is what has given rise to titles like “DevOps Engineer”

DevOps Engineer

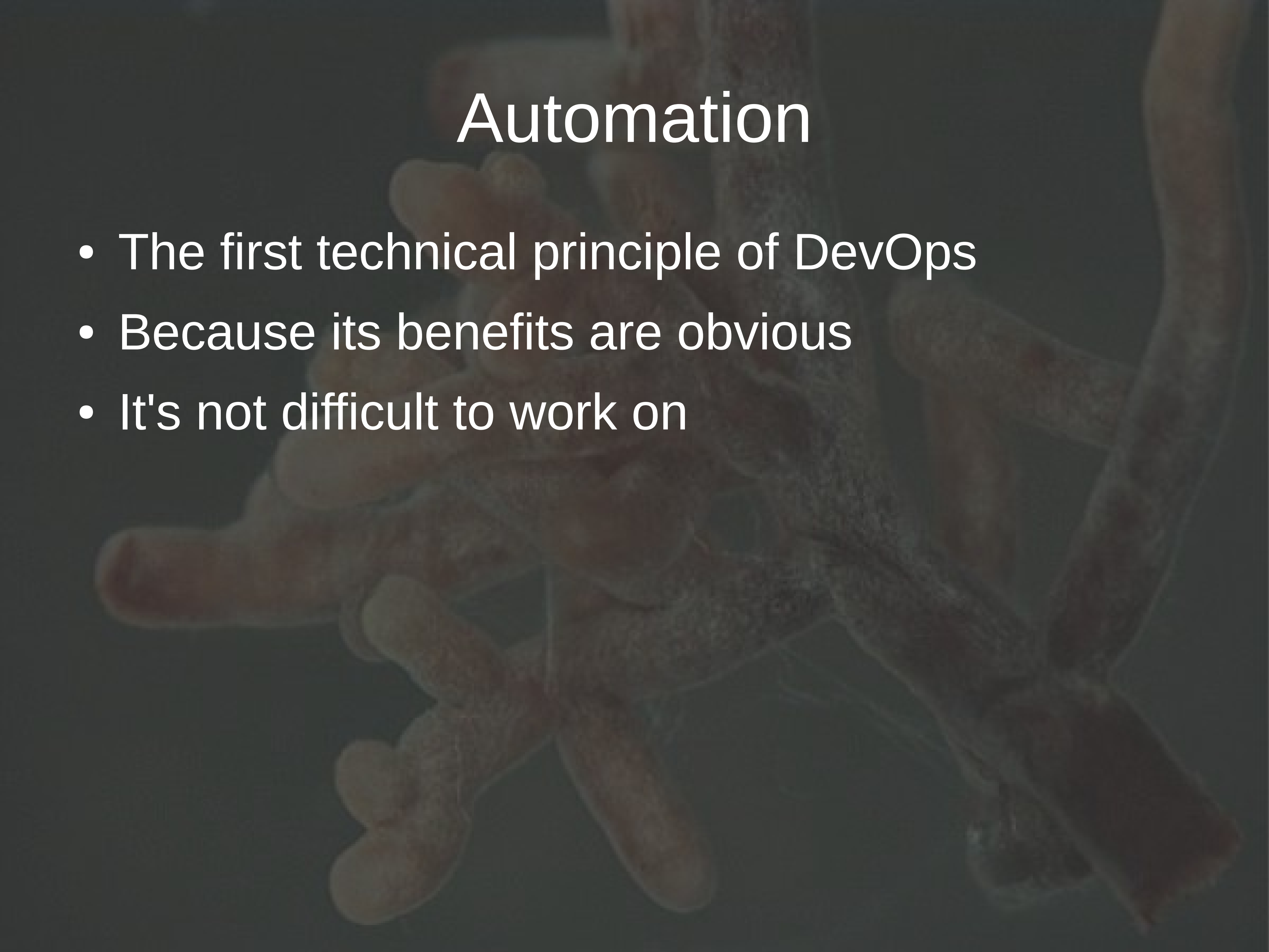


- Expected to understand the host of new technologies which are emerging
- Should understand the process of software development
- Should understand the process of software operations

Sharing Tools

- Possibly the most fun of all
- Ops adopt tools
  - Code management
  - Code review
  - Test automation
- Dev adopt tools
  - Monitoring & reporting
  - Infrastructure automation

1 - Automation

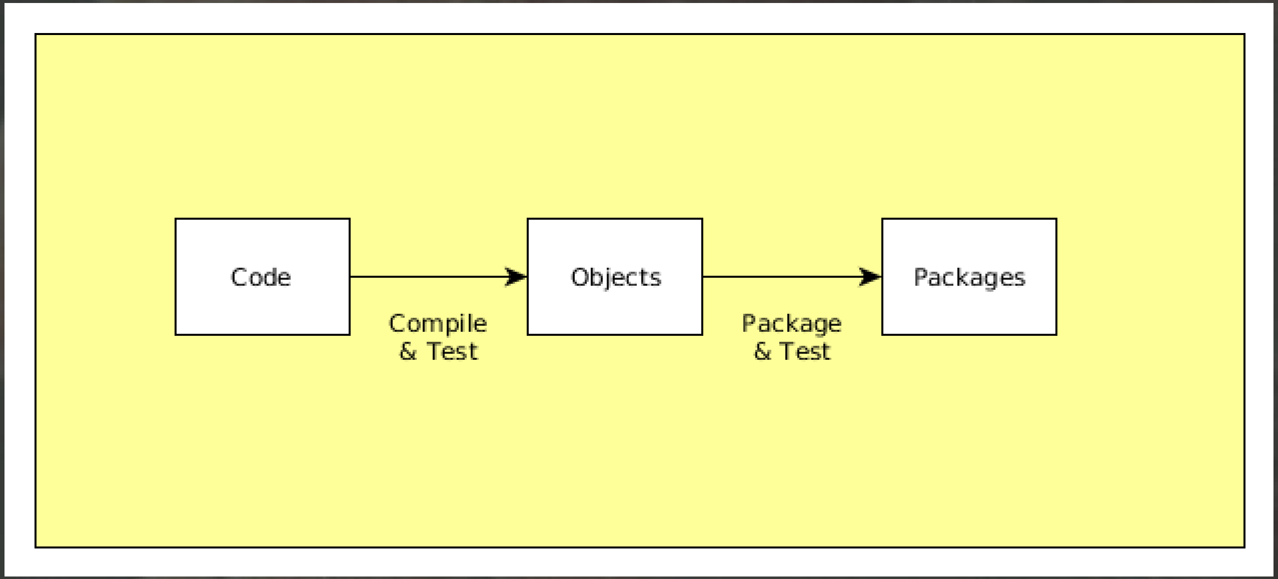


Automation

- The first technical principle of DevOps
- Because its benefits are obvious
- It's not difficult to work on

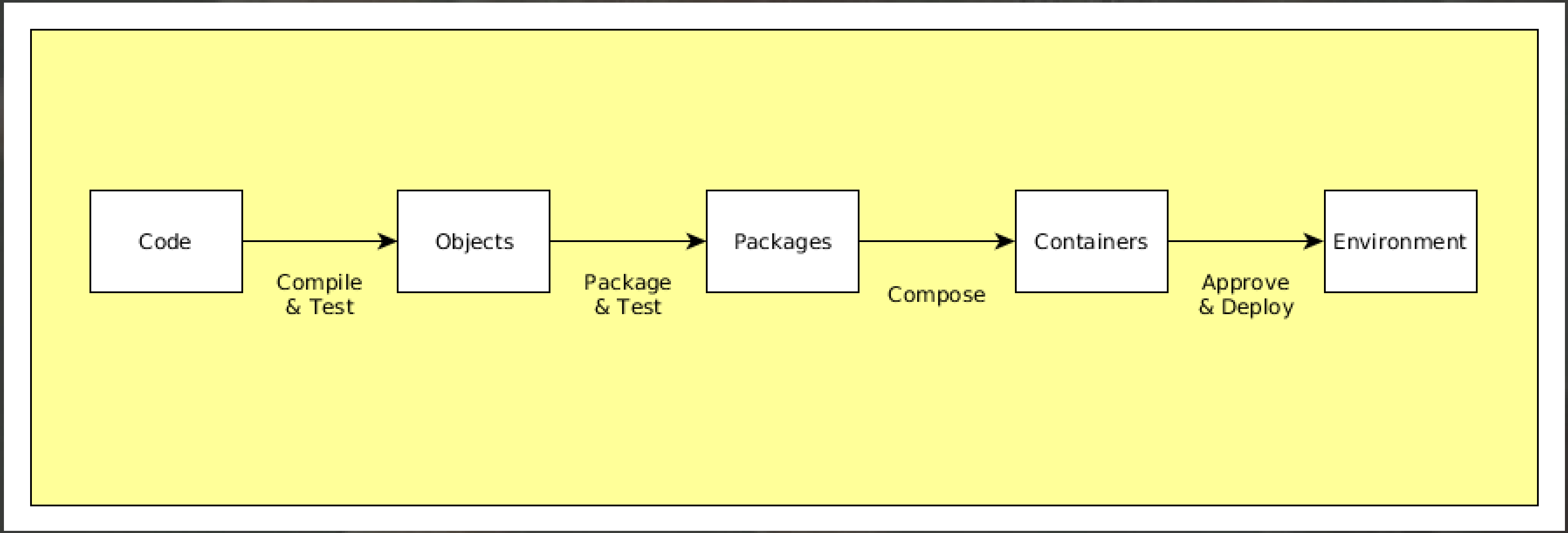
Automation – The Build Chain

- Development teams have had build chains for decades



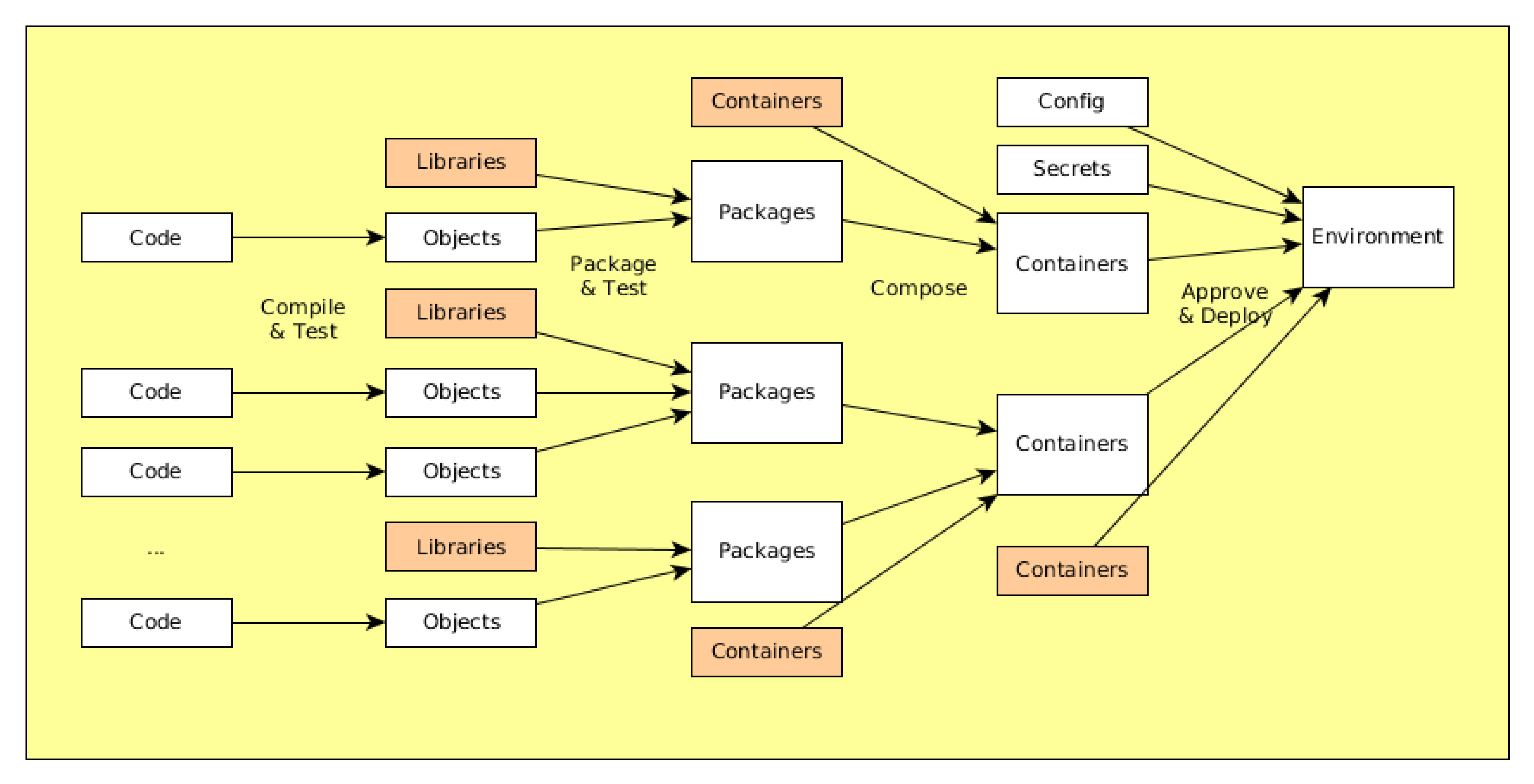
Automation – Deployment Pipeline

- DevOps extends the build chain to running code in an environment
- Now we call it a deployment pipeline



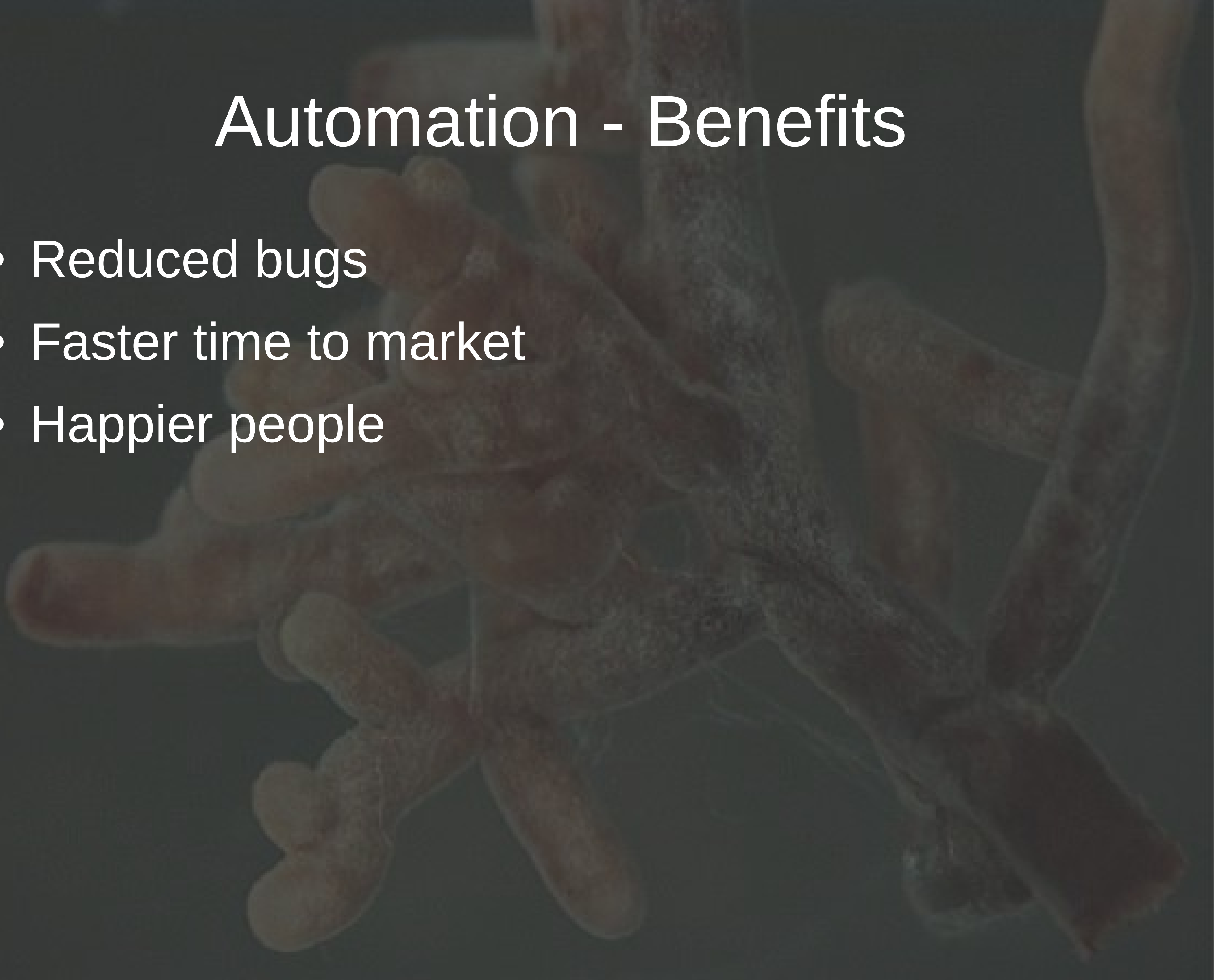
Automation – Deployment Tree

- In reality it's more complex
- Deployment Tree is probably a better term



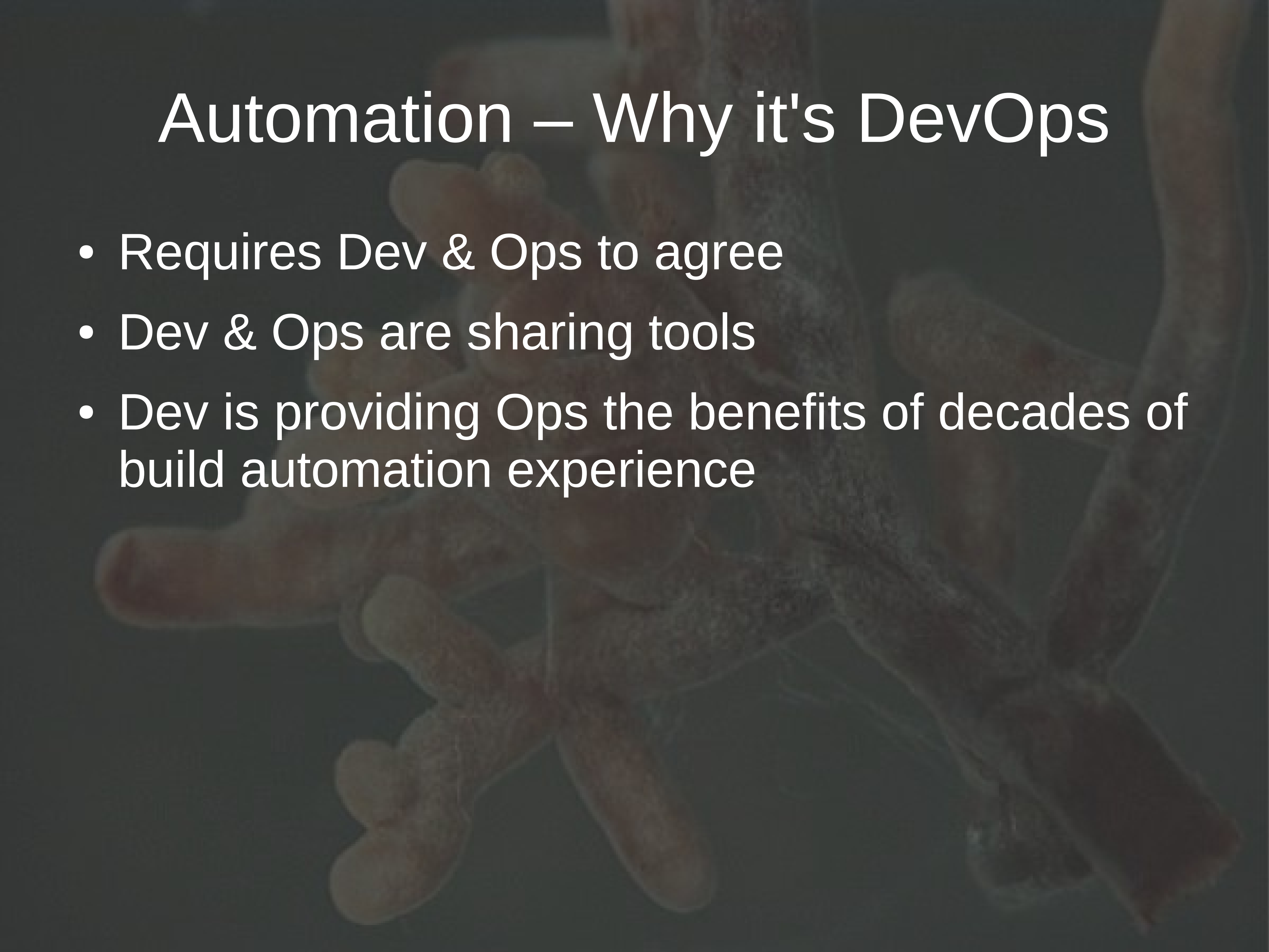
Automation - Benefits

- Reduced bugs
- Faster time to market
- Happier people

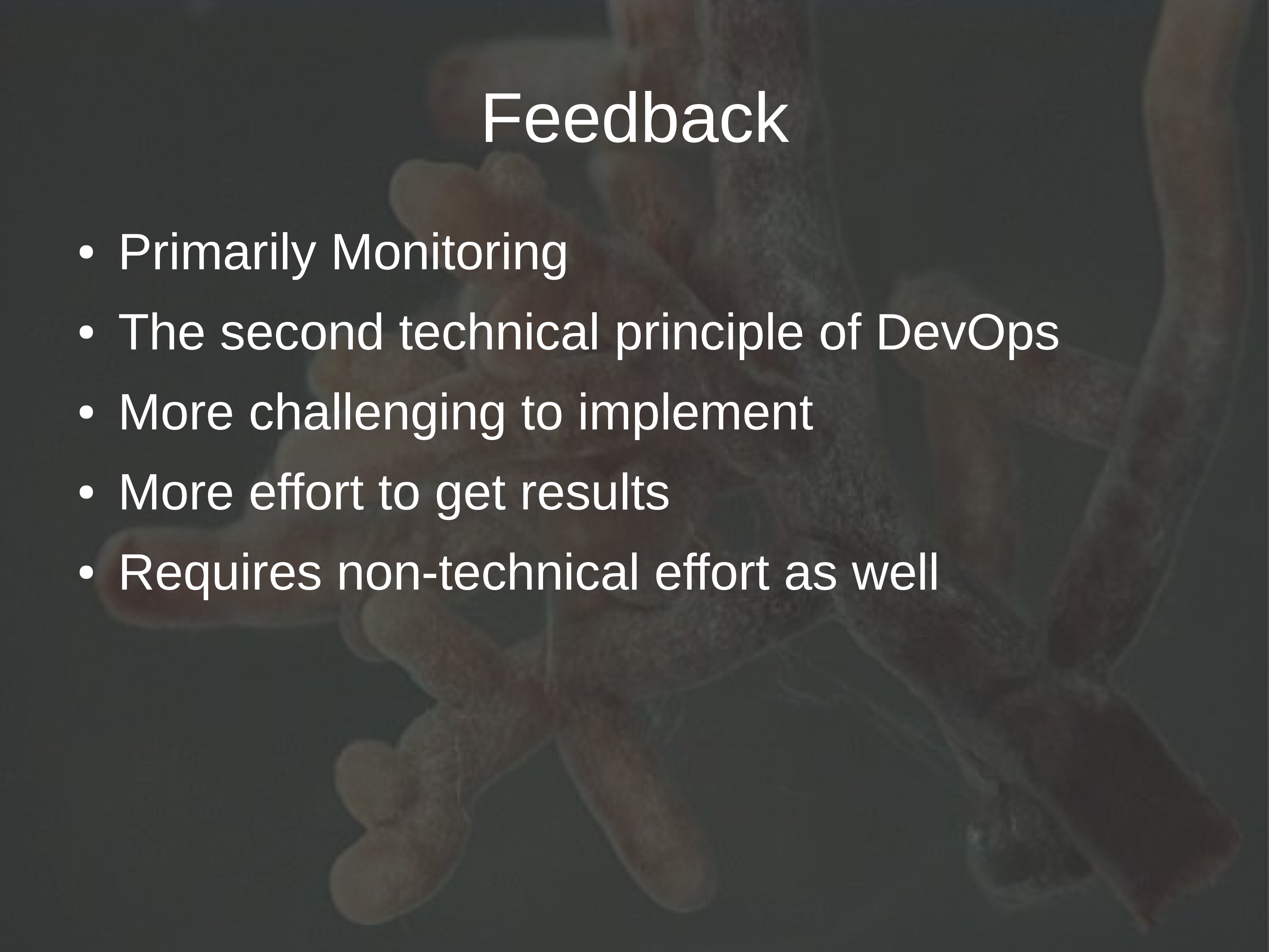


Automation – Why it's DevOps

- Requires Dev & Ops to agree
- Dev & Ops are sharing tools
- Dev is providing Ops the benefits of decades of build automation experience



2 - Feedback

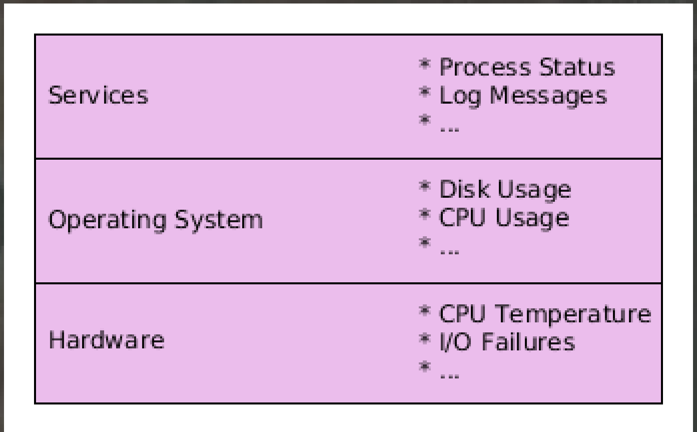


Feedback

- Primarily Monitoring
- The second technical principle of DevOps
- More challenging to implement
- More effort to get results
- Requires non-technical effort as well

Feedback - Monitoring

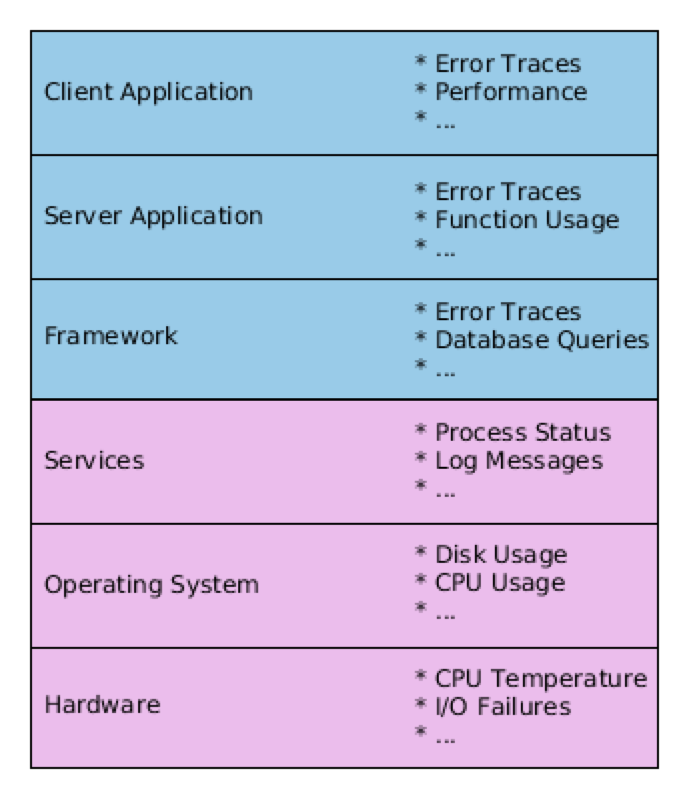
- Monitoring has been used by Ops for decades
- Focussed on the OS layer, CPU, Disk Usage, Memory, I/O



Services	* Process Status * Log Messages * ...
Operating System	* Disk Usage * CPU Usage * ...
Hardware	* CPU Temperature * I/O Failures * ...

Feedback – New Tools

- Monitoring into the code itself
- Combining monitoring with a debugger

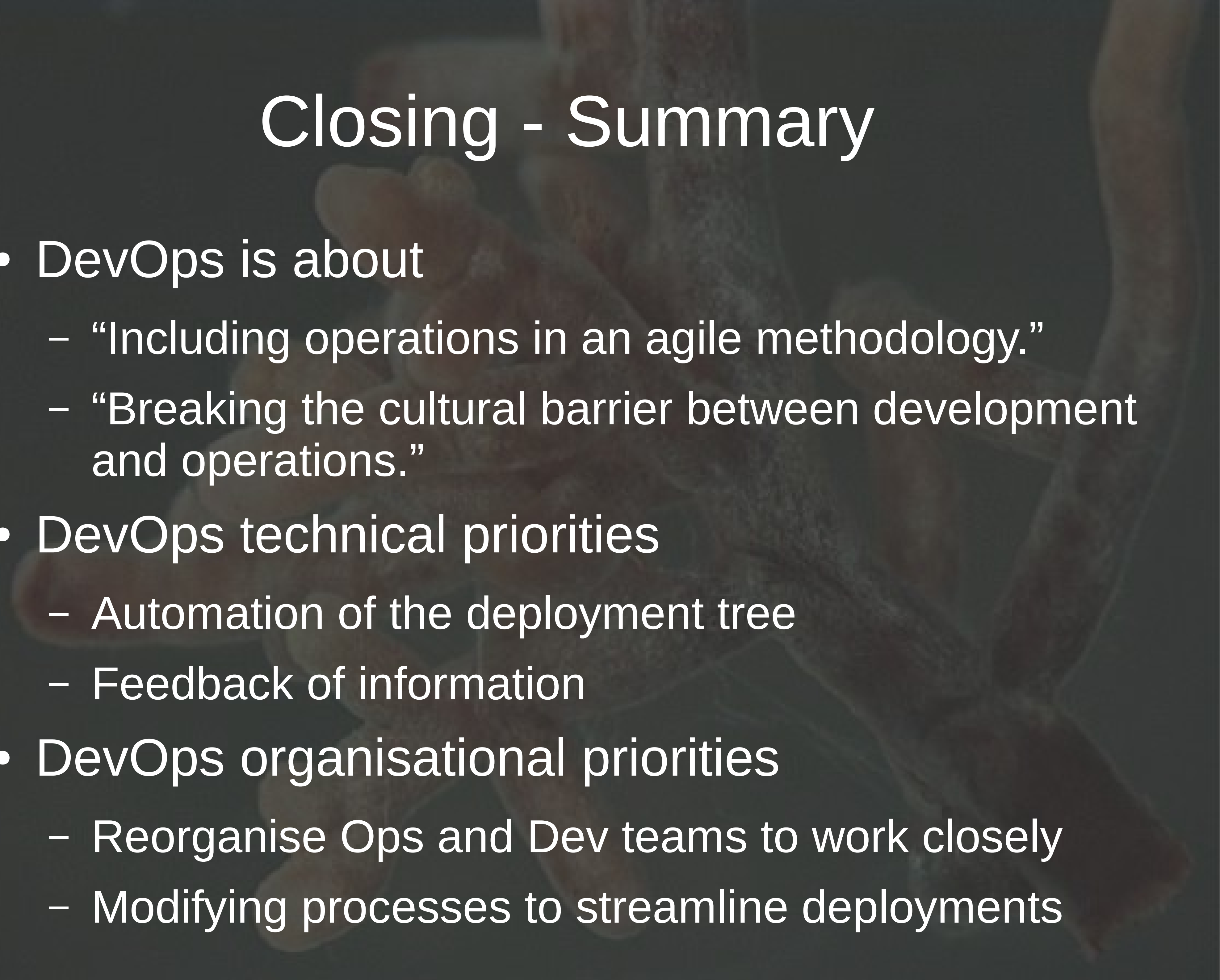


Client Application	* Error Traces * Performance * ...
Server Application	* Error Traces * Function Usage * ...
Framework	* Error Traces * Database Queries * ...
Services	* Process Status * Log Messages * ...
Operating System	* Disk Usage * CPU Usage * ...
Hardware	* CPU Temperature * I/O Failures * ...

Feedback – Why it's DevOps

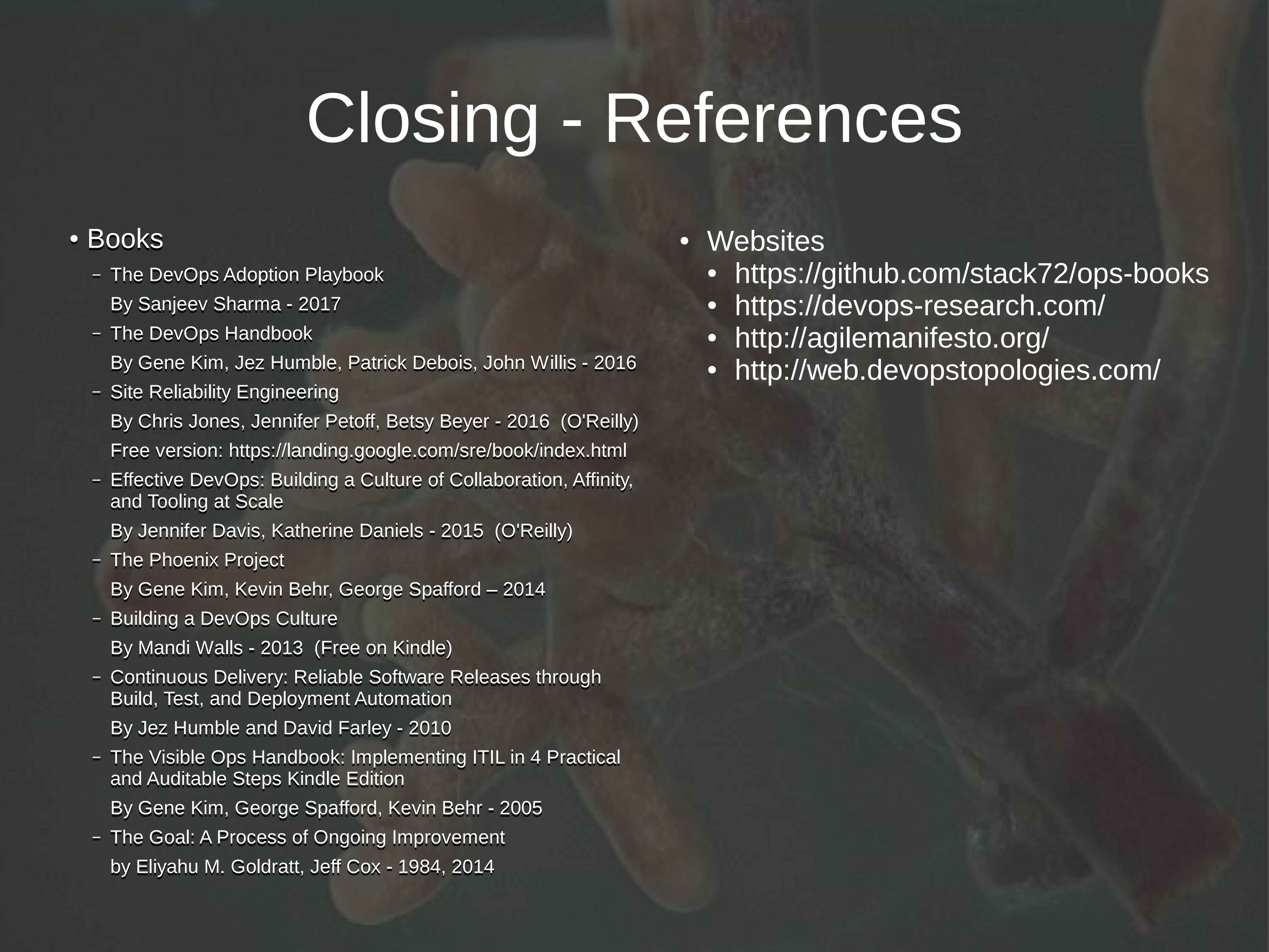
- Ops providing Dev the benefit of decades of experience with monitoring
- Application of Ops methodology to drive development
- Requires both teams to be involved

Closing



Closing - Summary

- DevOps is about
  - “Including operations in an agile methodology.”
  - “Breaking the cultural barrier between development and operations.”
- DevOps technical priorities
  - Automation of the deployment tree
  - Feedback of information
- DevOps organisational priorities
  - Reorganise Ops and Dev teams to work closely
  - Modifying processes to streamline deployments



Closing - References

- Books
  - The DevOps Adoption Playbook

    By Sanjeev Sharma - 2017
  - The DevOps Handbook

    By Gene Kim, Jez Humble, Patrick Debois, John Willis - 2016
  - Site Reliability Engineering

    By Chris Jones, Jennifer Petoff, Betsy Beyer - 2016 (O'Reilly)

    Free version: https://landing.google.com/sre/book/index.html
  - Effective DevOps: Building a Culture of Collaboration, Affinity, and Tooling at Scale

    By Jennifer Davis, Katherine Daniels - 2015 (O'Reilly)
  - The Phoenix Project

    By Gene Kim, Kevin Behr, George Spafford – 2014
  - Building a DevOps Culture

    By Mandi Walls - 2013 (Free on Kindle)
  - Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation

    By Jez Humble and David Farley - 2010
  - The Visible Ops Handbook: Implementing ITIL in 4 Practical and Auditable Steps Kindle Edition

    By Gene Kim, George Spafford, Kevin Behr - 2005
  - The Goal: A Process of Ongoing Improvement

    by Eliyahu M. Goldratt, Jeff Cox - 1984, 2014

- Websites
  - https://github.com/stack72/ops-books
  - https://devops-research.com/
  - http://agilemanifesto.org/
  - http://web.devopstopologies.com/

Closing - Attributions

- Background Image Source:
  Creative Commons Attribution 2.5 Generic License
  R. Henrik Nilsson, Erik Kristiansson, Martin Ryberg, Karl-Henrik Larsson (2005). "Approaching the taxonomic affiliation of unidentified sequences in public databases – an example from the mycorrhizal fungi". BMC Bioinformatics 6: 178. DOI:10.1186/1471-2105-6-178.
- Humpty Dumpty Image
  Public Domain
- Diagrams
  Author's