Purpose: share best practices and focus people on how to be really successful with DevOps

Main points: - Keep it simple stupid: complexity will come - don’t intentionally add complexity. - Everytime a developer is clever, an atrocity is committed: Keep it simple. Don’t add RegEx or Linq unless

* Software is too complicated (don’t add middleware you might need)
* Unit testing is important, but most misunderstand it
* DevOps/Agile is mostly about culture and behavior change, not tools - but most organizations butcher it to match ‘their’ process, instead of adapting to DevOps best practices

Where code falls apart (Agile/DevOps/etc).

1. Developing an idea
2. Leading the team
3. being a leader 1. striving for leadership. What is leadership? What isn’t leadership? When you are being a manager? Why protecting and advocating for your direct reports is the most important part 1. Trying to do better than your goals 1. Lead by example 1. invest in yourself and team: continuous learning and networking opportunities - retention is a trailing indicator - but people will stay longer than usual if they are still learning, even if salary is less than ideal 1. Invest in yourself: sleep is important to recharge and repair.
4. Hiring a team 1. How to conduct interviews
   1. Passion
   2. debugging tests
   3. Thinking tests
   4. No whiteboards. No mindbenders
      1. Developing the team
   5. Managing ‘clever’ developers
   6. Encouraging right incentives
5. Creating the product 1. DevOps & Agile
   1. Creating time on Backlog for technical debt and continuous improvement
      1. Code analysis
      2. Code Reviews
      3. Agreeing on formatting - using tools (EditorConfig) to take the opinion out of development
      4. Tools and technologies
      5. Monitoring for feedback
      6. Technology. When to use microservices. When not to. When to use containers. What language to use.