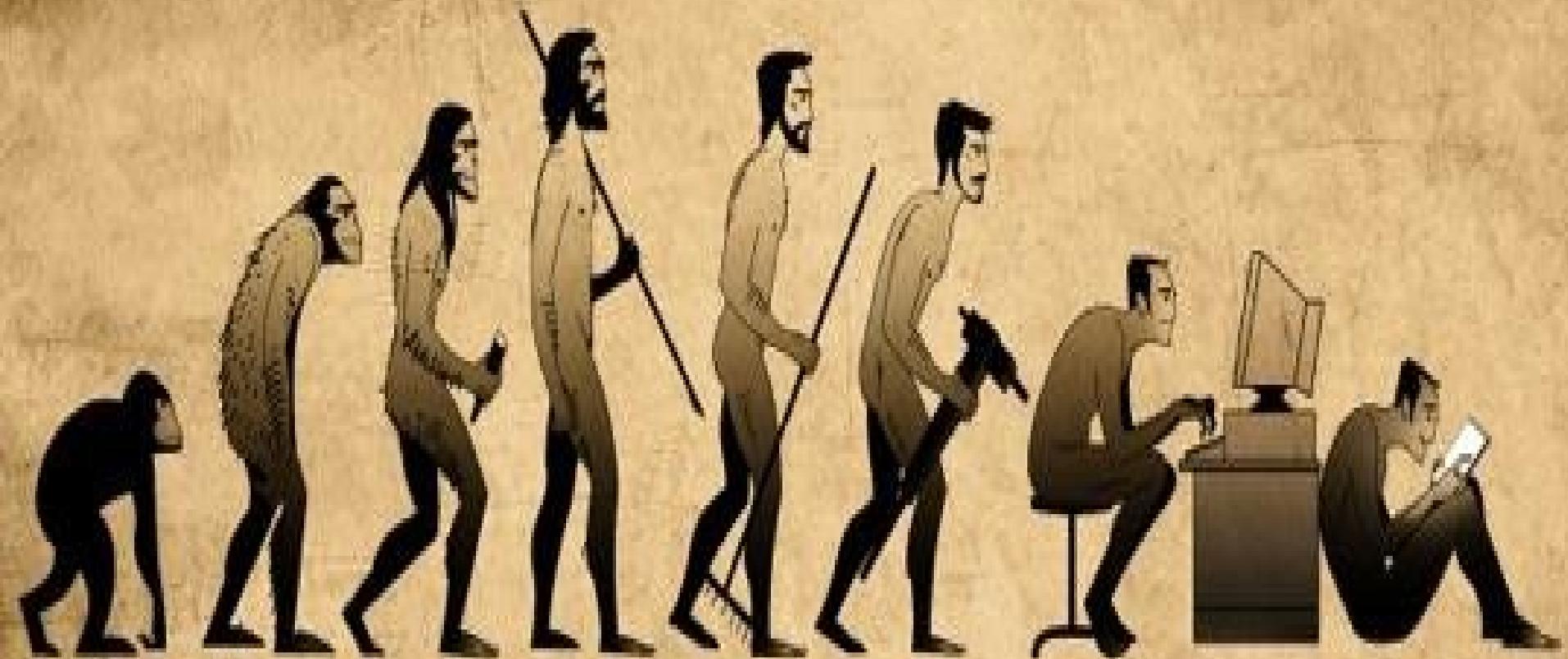


Toolbelt Psychology

Mike Place
SaltStack

mp@saltstack.com







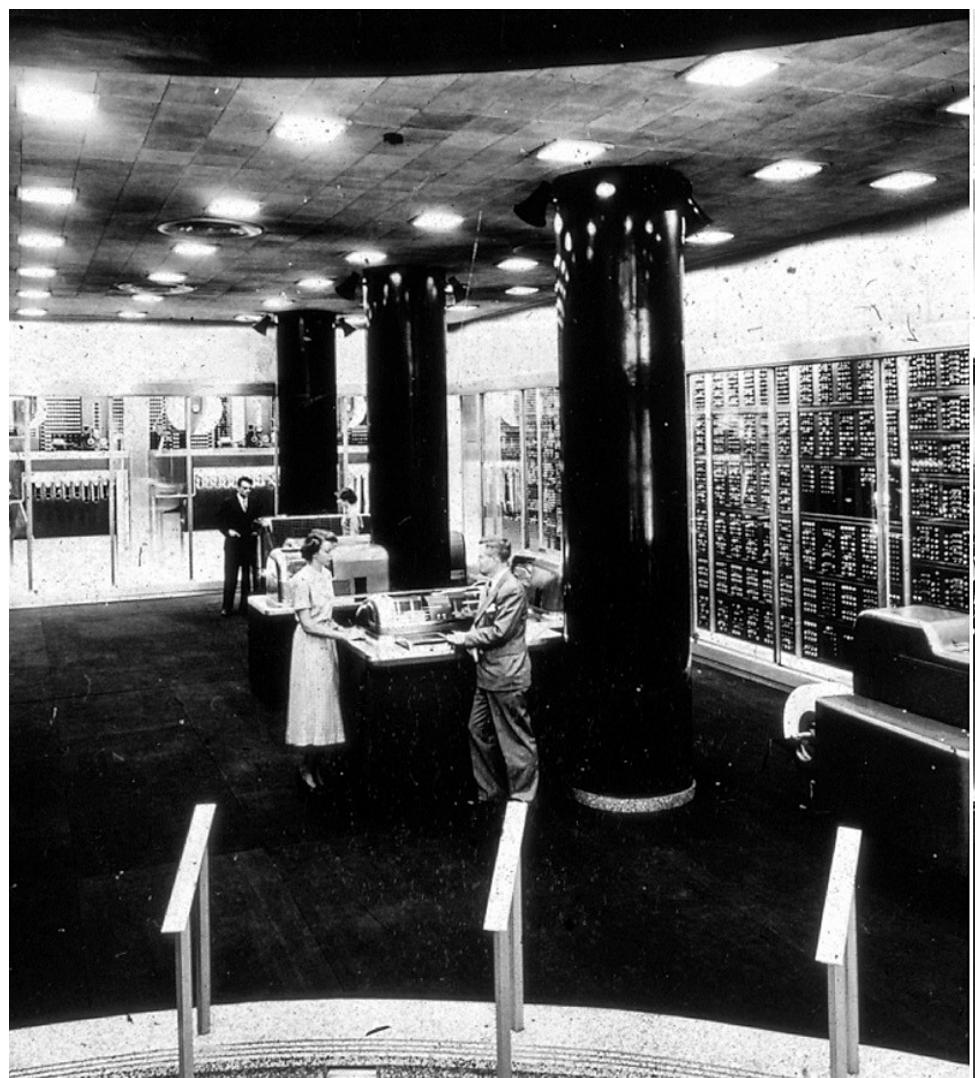




Tools

When we examine the history of an era, we too often restrict ourselves to the things that we built.

To understand a history, we need to both look at the tools that were used to make them. (Sometimes they even outlast what they built!)



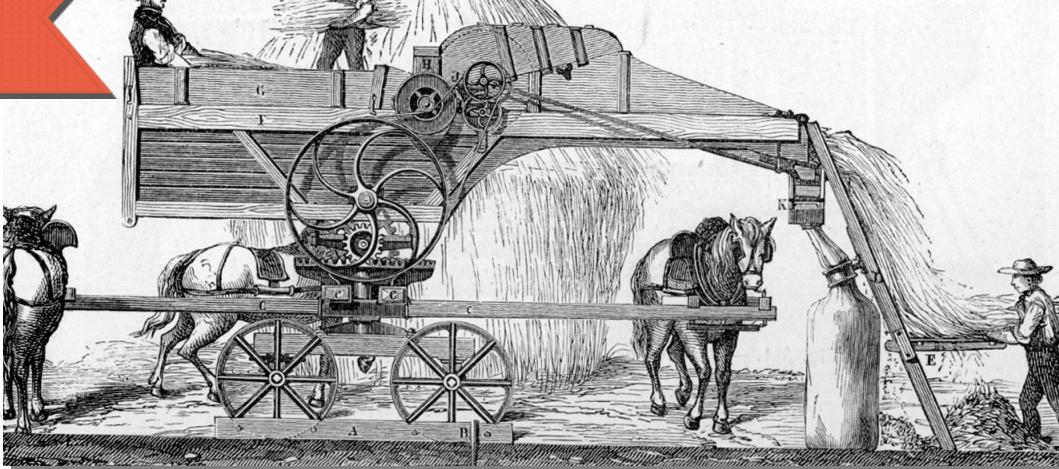


NON-RETURNABLE

LECTURE - DEMONSTRATION
← 10 11 →

Automate!

What improves?



Speed: The ability to produce more in less time

Reliability: The ability to produce goods at a consistent rate.

Quality: The ability to standardize on a given form and ensure that form is easily reproducible.

Cost: The ability to lower the cost of each unit produced.

Automate!

What does
not
improve?

Abstraction: Can be a positive but also a negative. Understanding of lower layers could decrease.

Craftsmanship: When we automate, we

Safety: Automation makes it exceedingly easy to destroy everything.



DON'T

AUTOMATE

WHAT

YOU

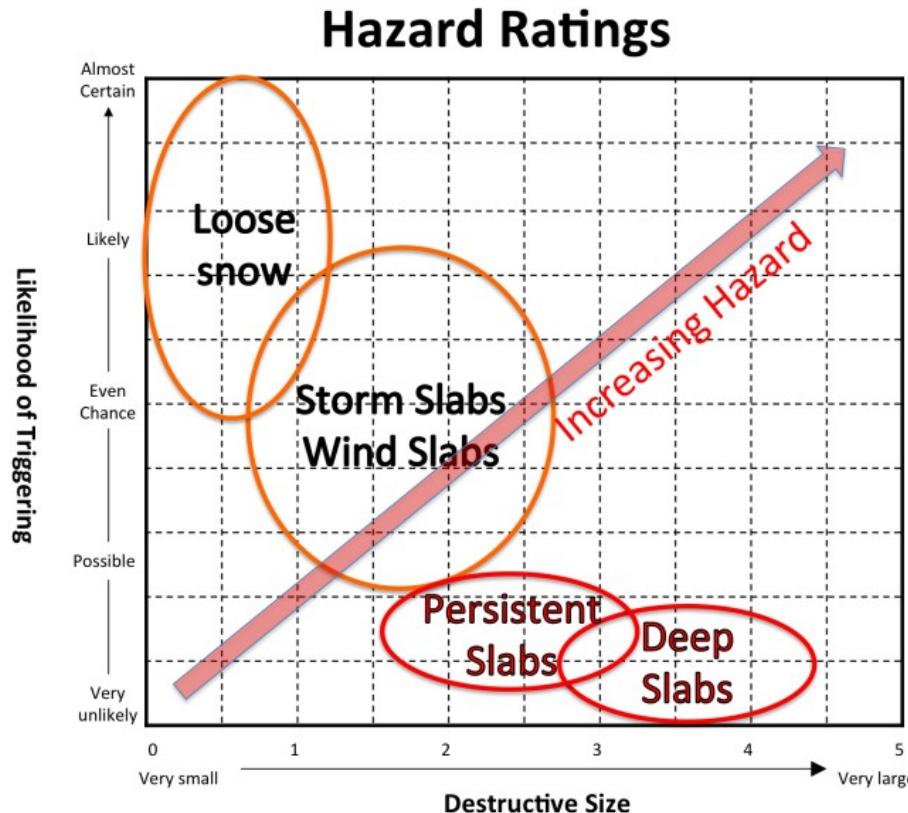
DON'T

UNDERSTAND



Probability and Consequences

What is the problem we are trying to solve?





Write programs that **do one thing** and do it well.

Write programs that **work together**.

Write programs to **handle text streams**, because that is a universal interface.

UNIX Program Design

“Much of the power of the UNIX operating system comes from a style of program design that makes programs easy to use and, more important, easy to combine with other programs. This style has been called the use of software tools, and depends more on how the programs fit into the programming environment and how they can be used with other programs than on how they are designed internally. [...] This style was based on the use of tools: using programs separately or in combination to get a job done, rather than doing it by hand, by monolithic self-sufficient subsystems, or by special-purpose, one-time programs.”

- *Brian Kernighan and Rob Pike, "Program Design in the UNIX Environment"*

UNIX Program Design

“As a programmer, it is your job to put yourself out of business. What you do today can be automated tomorrow.”

- *Doug McIlroy, invented the UNIX pipe*



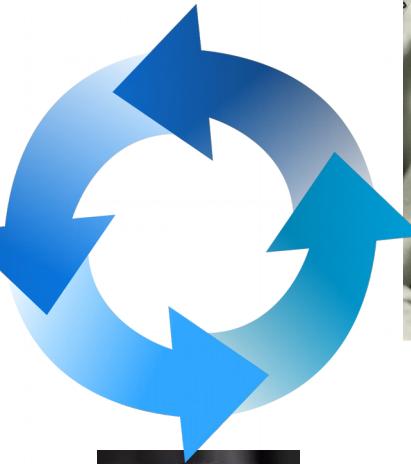
In the Beginning...

There were no differences between "dev" and "ops" because there was only one.

-John Allspaw (Etsy)

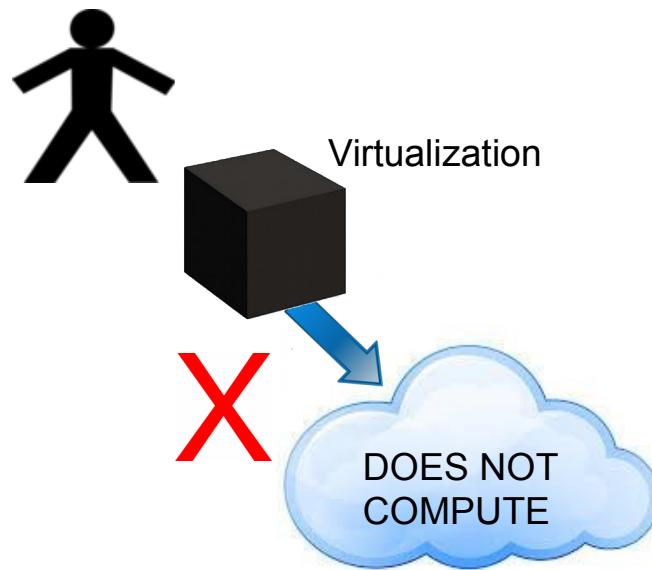
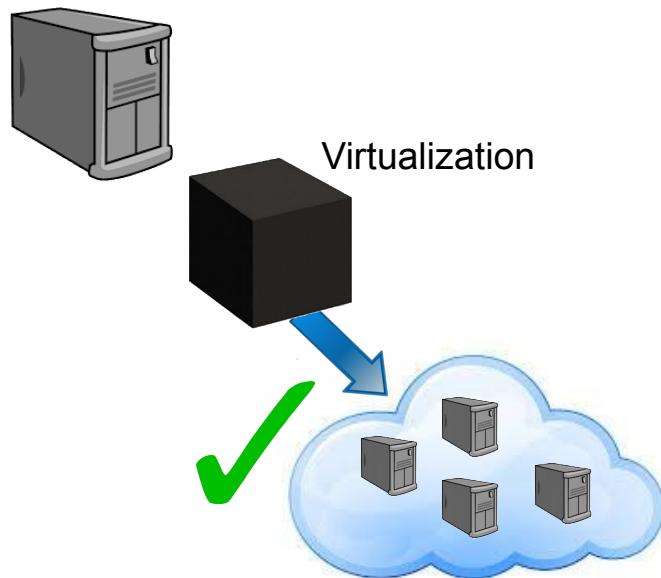


Then Ops Happened!



A Problem of Scale

- Humans don't virtualize quite as well as machines do.



EVERYBODY IS OUT DEVOPSING



**AND I'M JUST SITTING HERE
SYSADMINING**

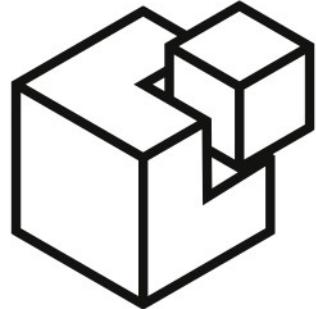
“We also use [that] limitation purposely **as a tool** to understand things, to form the illusion of mastery and control over a limited scale of things, because by being able to isolate only a part of the world, we reduce a hopeless problem to a manageable one.”
— Mark Burgess, In Search of Certainty: The Science of Our Information Infrastructure



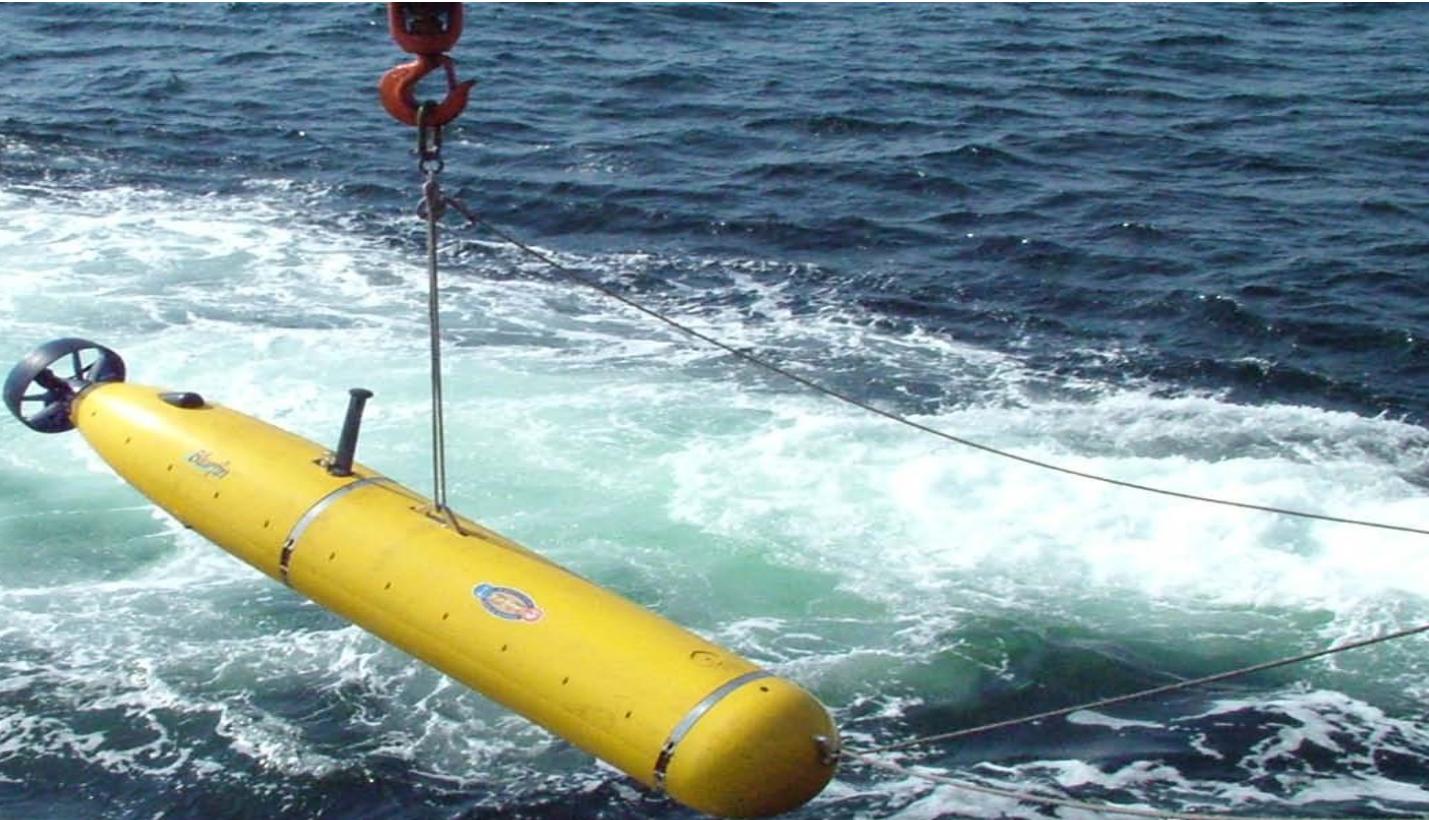








SALTSTACK





BEACONS



3 Questions to Ask Ourselves

1. How can we understand the problems that we are trying to solve by examining the tools which we use to solve them?

2. Can automation decrease complexity or can it only increase efficiency?

3. Are we using automation to augment our understanding or as a substitute for failing to understand?

Thanks.

Questions?