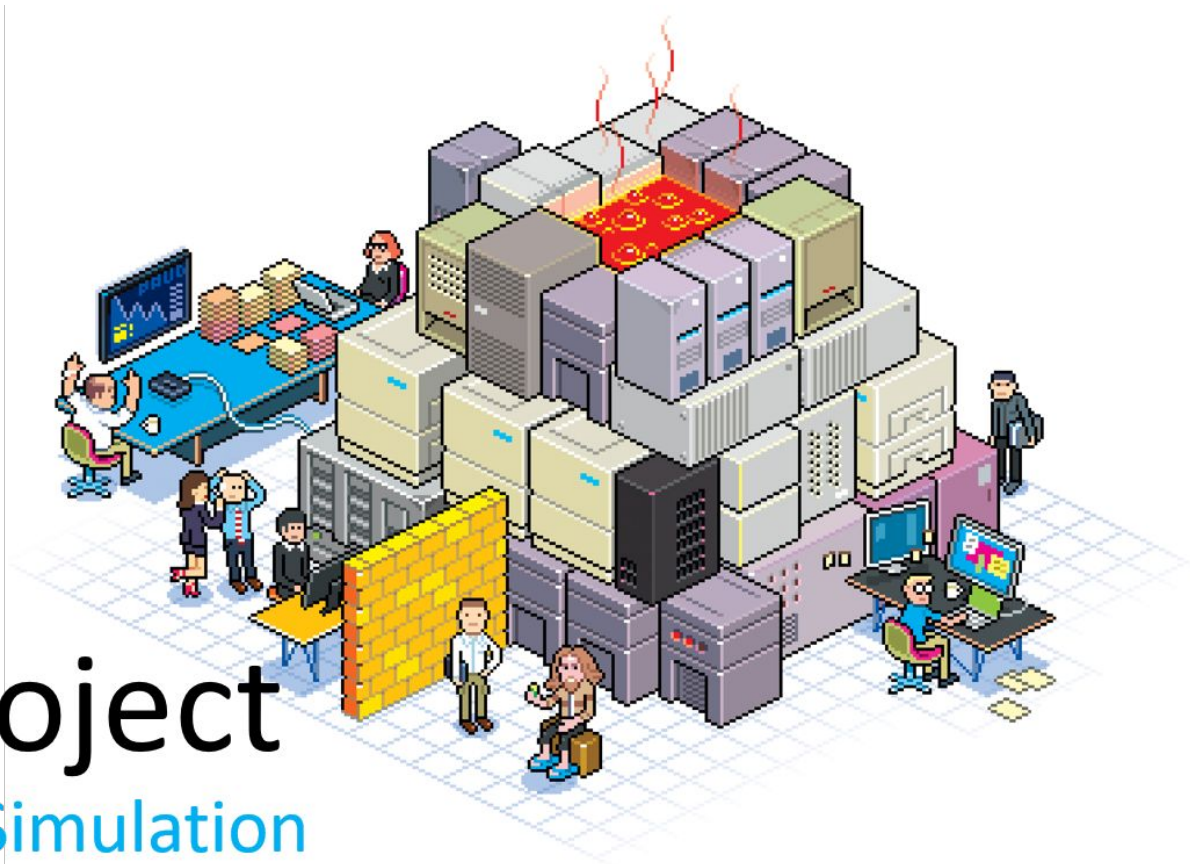


The Phoenix Project

A DevOps Business Simulation

by Gene Kim, Kevin Behr and George Spafford



Presented by Katharina Hofschien

One of the top novels for IT professionals



What to expect?

A story of a failing company that is able to turn the ship around by implementing

the 3 Ways:

1. Flow
2. Feedback
3. Culture of continuous learning

These 3 Principles are at the **basis of all DevOps patterns.**

Meet the Company: **Parts Unlimited**

Manufacturing and retail company specialized in automotive parts:

- Struggling to keep up with competition
- lots of layoffs
- stuck in the 1920s

According to Steve Masters the CEO:

"We're years late delivering, our investors and Wall Street are howling and now my board is losing confidence in our ability to hit our commitments.

*I'll be blunt, the way things are going,
I'll be out of a job in six months. [...]*

What's at stake here is not just my job but the nearly four thousand employees who work here Parts Unlimited."



Bill Palmer: new VP of IT operations



Laura from HR:

“Effective as of this morning, Luke (former CIO) and Damon (former VP IT-Ops) are no longer with the company. This went all the way to the top, with Steve getting involved.

He’s chosen you to be the next VP of IT operations.”

Bill reluctantly goes from being a midrange manager of one IT ops team to being the VP.

In his previous team he ran a tight ship but will soon discover the mayhem of being at the intersection between IT ops and the rest of the company.

The Phoenix project

The **last hope** to close the gap with competition and regain profitability.

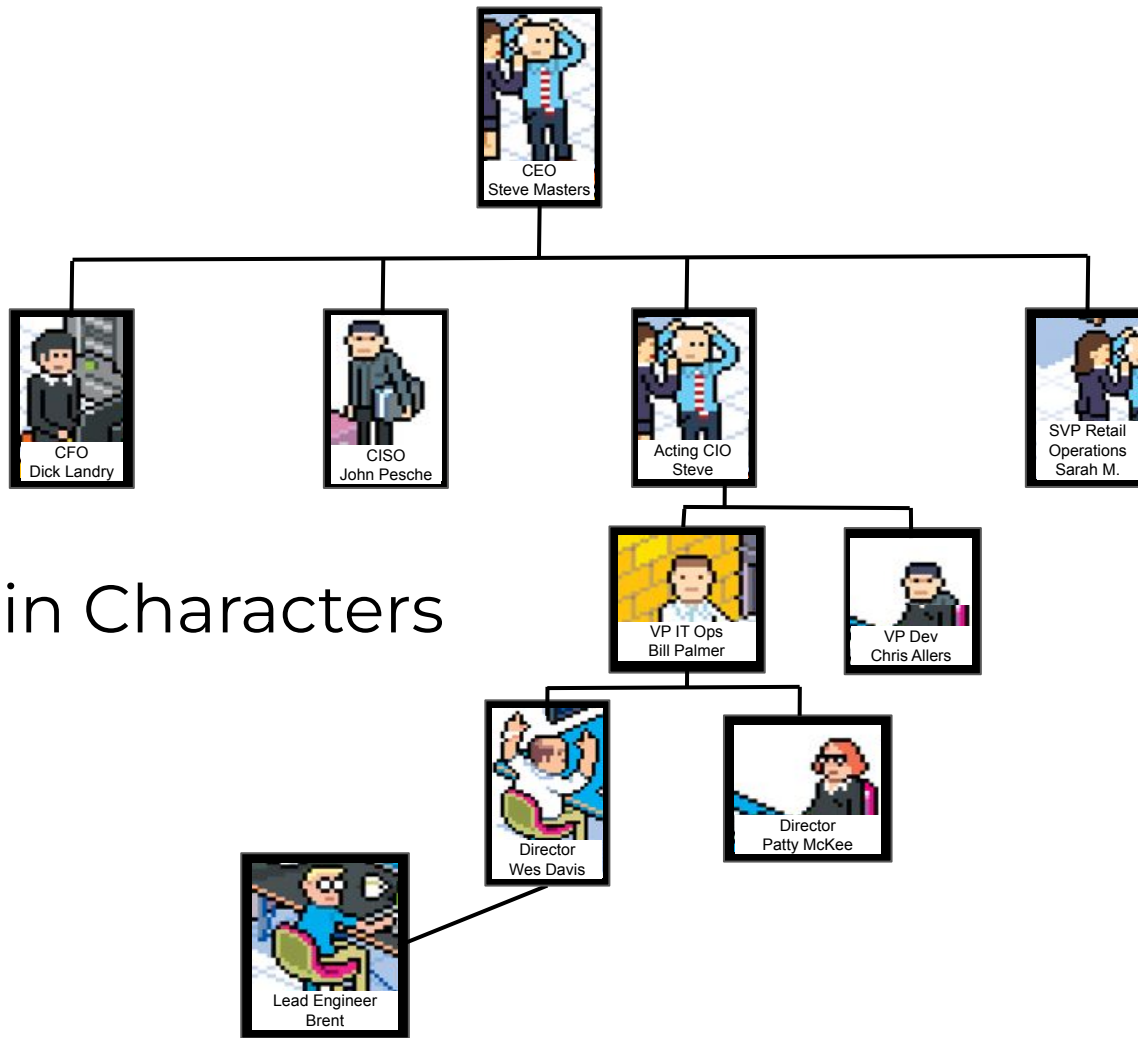
Project Phoenix will allow:

- customers to buy from the company wherever they want, whether online or in-store.
- Additional features for better marketing and recommendation
- And more

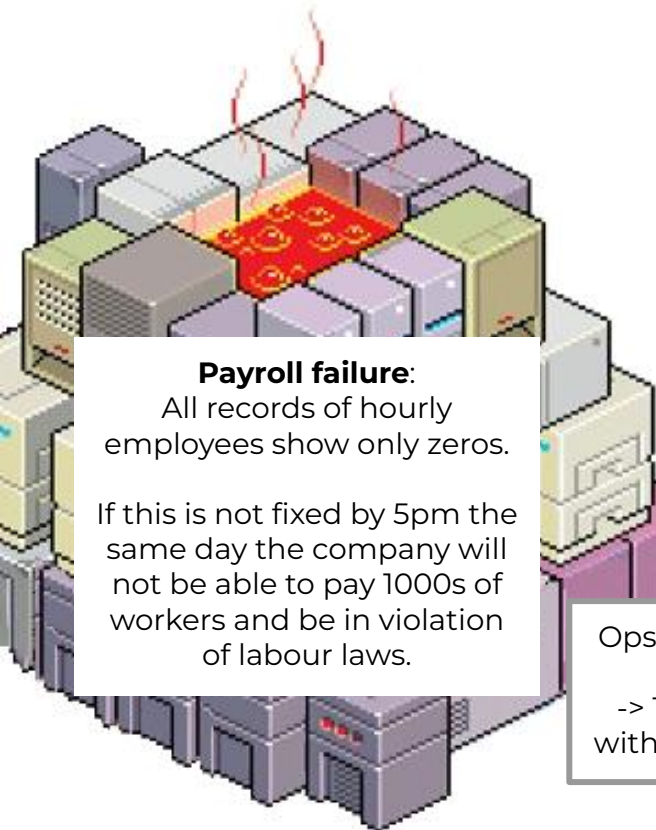
It has already **cost millions** and is running **months late**.



Our Main Characters



Bill's first day: What caused the payroll to crash?



Payroll failure:

All records of hourly employees show only zeros.

If this is not fixed by 5pm the same day the company will not be able to pay 1000s of workers and be in violation of labour laws.

Ops/Brent rolls back SAN upgrade
-> it completely crashes

Failures on lots of DBs.
Error messages: in some unknown alphabet.

Ops upgrades the SAN firmware
-> 15 min later errors with the payroll pop up

Further investigation from finance reveals:
all social security numbers are weird icons and symbols.
Unlikely to have been caused by SAN.

Further investigation reveals:
An IT Security guy put pressure on a developer to make last minute changes before his vacation:

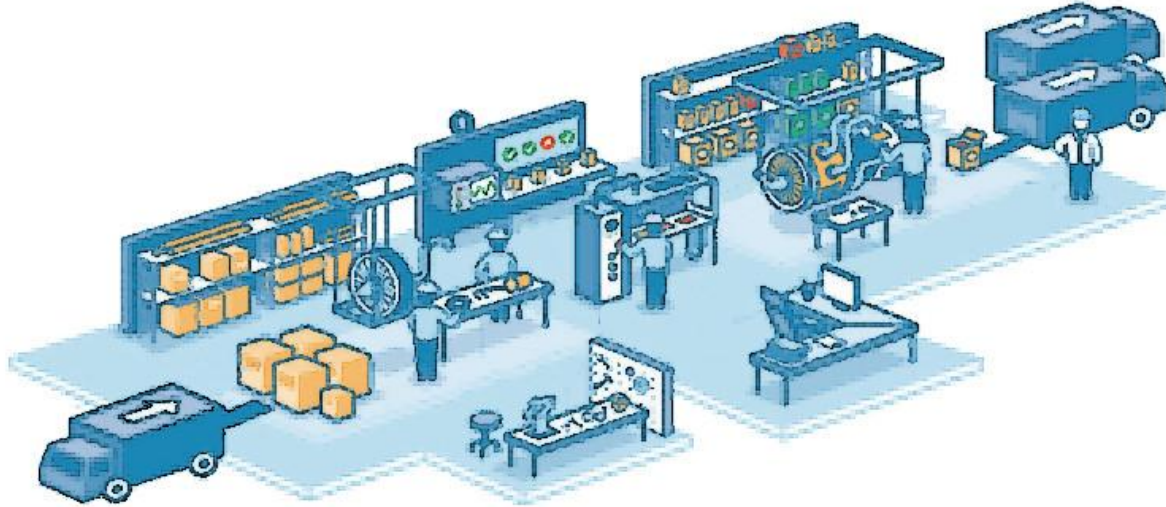
"anonymize the social security number so the company is in compliance for security audit next week."

Without testing.

Finance had to pay all employees based on the previous payroll

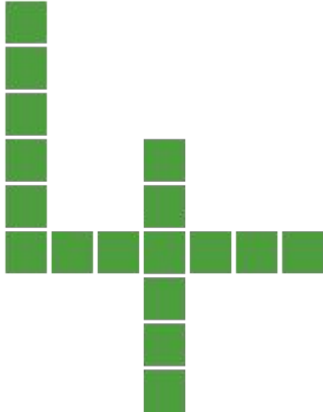
Failure splashed across news the next day

Bill meets Erik aka “Yoda of the 3 Ways”



Erik tells Bill that IT is a lot like Manufacturing plants and that Bill needs to look at them closely to learn how to improve the situation.

What is work?



Types of work:

1. Business work (eg. new order system)
2. Internal It projects (eg. automation)
3. Changes (eg. deploys, improve performance)
4. Unplanned work (eg. site down, server crash)

1st Way: Systems Thinking (Dev to Ops)

- Understand the flow of work
- Always seek to increase flow
- Never unconsciously pass defects downstream
- Never allow local optimization to cause global degradation

DevOps patterns derived from this

1. **Pattern:** Define the work and make it visible (eg. with Kanban boards)
2. **Pattern:** Limit Work In Progress (WIP)
3. **Pattern:** Reduce Batch sizes and reduce Handoffs



What/ Who is a constraint?

Brent

The weakest link in the chain of flow of work.

Bottleneck: where the flow is limited

“Any improvement on anything that is not the constraint or bottleneck is an illusion”

1. Identify the constraint
2. Exploit the constraint
3. Subordinate the constraint
4. Elevate the constraint
5. Prevent from becoming the constraint



The 2nd way: Feedback loops (Ops to Dev)



- Understand and respond to the needs of all customers, internal and external
- Shorten and amplify all feedback loops
- Create quality at the source

4. Pattern: Embed Dev into IT ops (eg include all dev and Ops when there are incident escalations and also For blameless post mortem analysis)

5. Pattern: Ensure App monitoring and metrics

3rd Way: Culture of Continuous Learning

Foster a culture that rewards:

- Taking risks and learning from failure
- Repetition leads to mastery or the Kata of Improvement

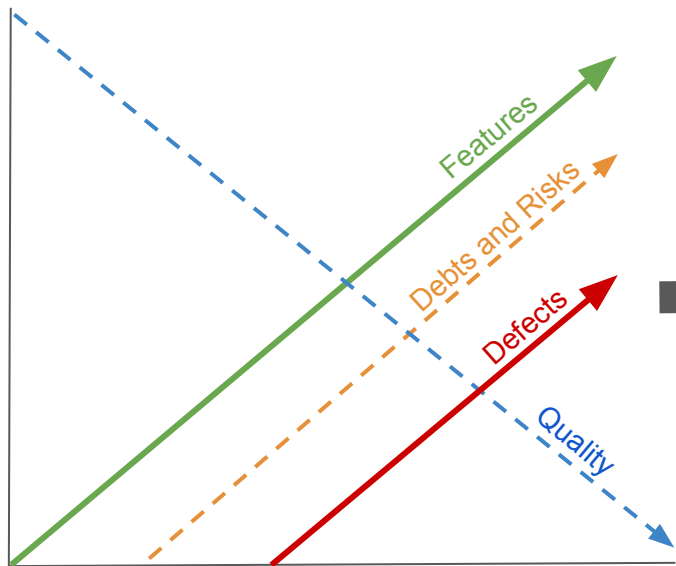
5. Pattern: Inject Failures often to make rollouts smoother and less painful

6. Pattern: Break things before Production

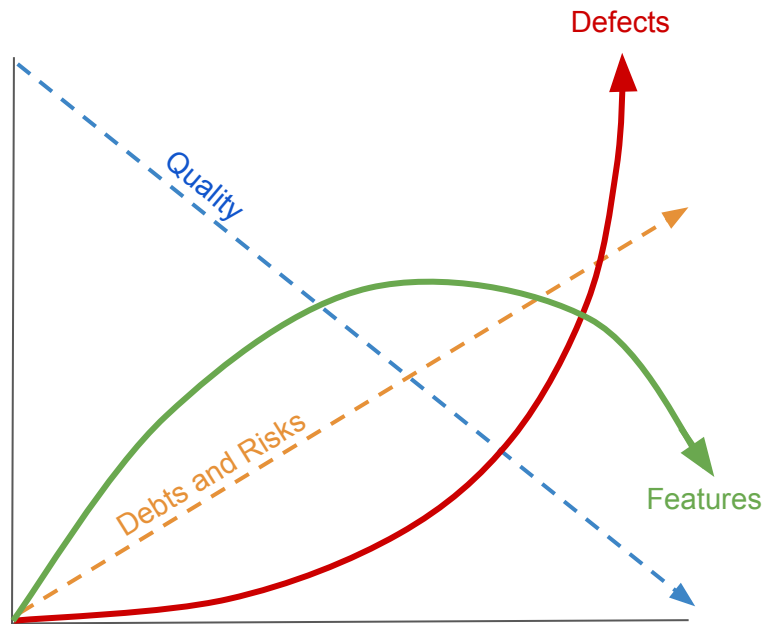
7. Pattern: Allocate 20% of cycles for technical debt reduction



Technical debt



Example: Fast push to market



How to get out of this situation?

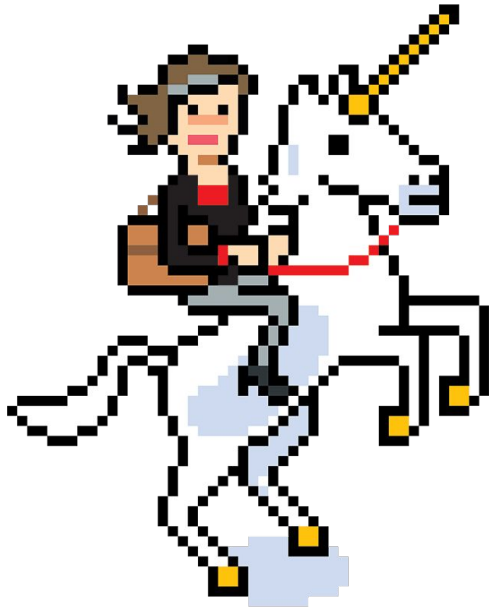
Short term: A **feature freeze**

Long term: **20%** of cycles to debt **reduction**

Rough overview of steps Bill takes

1. Cataloging work and scheduling it
2. Many outages and blow ups
3. Starting communication and trust between Dev and Ops
4. More very public disasters
5. After being undermined by CEO, Bill quits his job
6. Erik talks sense into CEO: Trust building exercise between all IT sections
7. Project freeze
8. Kanban board for critical resources
9. Protect Brent to focus on Phoenix or if needed in outage to shadow others
10. Dev, QA, OPS and Security start working together
11. Automated env creation allows easier migration to cloud for small projects

Introducing: The Unicorn Project



Bill finds out what revenue the goals are and how to reach them.

A small SWAT team starts working towards these goals

- Using Automated env creation
- Frequent deploys
- A/B testing
- Following the 3 ways and all patterns

Test Recommendations on 1% of customers was a huge success

Making the recommendations available on Black Friday lead to such a big success that inventory was out of stock

Additional resources

Free access to the Book for UniBZ students:

<https://learning.oreilly.com/library/view/the-phoenix-project/9781457191350/>

Kim Gene on why we need DevOps:

https://www.youtube.com/watch?v=877OCQA_xzE

Artwork by Mike Collins



Thank you! And feel free to ask questions.