Q & A with Gene Kim
What is *The Phoenix Project* about, and how does it relate to Kanbans?
“This book will have a profound effect on IT, just as *The Goal* did for manufacturing.” –Jez Humble, co-author *Continuous Delivery*

“This is the IT swamp draining manual for anyone who is neck deep in alligators.” –Adrian Cockroft, Cloud Architect at Netflix

“This is *The Goal* for our decade, and is for any IT professional who wants their life back.” –Charles Betz, IT architect, author “Architecture and Patterns for IT”
How were you introduced to Kanban?
From Worst to Best in 9 Months:
Implementing a Drum-Buffer-Rope Solution in Microsoft’s IT Department

By David J. Anderson & Dragos Dumitriu, Microsoft Corporation, November 2005

Abstract
This is a case study about implementing common sense changes where they were needed. It’s a story not about the brilliance of the Theory of Constraints (TOC) but rather TOC playing a role as permission giver, reinforcing the beliefs of a manager and encouraging him to do the right thing. It’s also a story about simplicity – making just a few simple changes, collecting less data, spending less time on overhead and bureaucracy and more on productive tasks.

…”lead time went from 155 days to 22 days. Lead times were so good that they created a new SLA guarantee of 25 days…”
Why are Kanban boards so prominent in *The Phoenix Project*?
The All Too Familiar Problem Statements

- We get worked “dumped on us” by everyone else
- We sometimes make promises that we can’t deliver on
- Our work is always late
- We defer work that we shouldn’t
- No one takes ‘no’ for an answer
- We act before understanding the business
- Business is dissatisfied with our work
The All Too Familiar Problem Statements

- Promises to customers and other stakeholders never met
- Too much firefighting and heroics
- We cut corners to make the promised dates
- Ever accumulating technical debt
- Less and less time to retire technical debt
- Business is dissatisfied with our work
- Quality problems cause outages, service impairments
- Business outcomes aren’t met
- Everyone loses…
High Performing DevOps Teams

- They’re more agile
  - 30x more frequent deployments
  - 8,000x faster lead time than their peers

- They’re more reliable
  - 2x the change success rate
  - 12x faster MTTR

The Three Ways

1. Dev → Ops
2. Dev → Ops
3. Dev → Ops
Why 15 min Tasks Take Weeks To Complete

Task queue time (actually, queue length)

- 50% busy: 1 day
- 75% busy: 3 days
- 90% busy: 9 days
- 95% busy: 19 days
- 99% busy: 99 days
Kanbans

- What is it, and why does it work?
  - Popularized by Taiichi Ohno (Toyota)
  - Scheduling board for lean production
  - Good: Visualizes work in a system
  - Better: Visualizes work FLOW THROUGH a system

- Outcomes: Work gets done!
  - Work takes less time to complete (i.e., reduced cycle time, on time!)
  - Better tracking of effort and costs
  - Find recurring work that we can automate
  - Find work where there’s too much time ‘waiting’ or ‘in queue’
  - Business gets what they need, when they need it

Source: Branden Williams
Write project plan for new automated code deployment

Review code for Project Hoosegow, Sprint 39
RealGeneKim

Source: Dominica DeGrandis, David J. Anderson
Deploy Smaller Changes, More Frequently *

Facebook Chat
By Eugene Letuchy on Tuesday, May 13, 2008 at 10:56pm

One of the things I like most about working at Facebook is the ability to launch products that are (almost) immediately used by millions of people. Unlike a three-guys-in-a-garage startup, we don’t have the luxury of scaling out infrastructure to keep pace with user growth; when your feature’s userbase will go from 0 to 70 million practically overnight, scalability has to be baked in from the start. The project I’m currently working on, Facebook Chat, offered a nice set of software engineering challenges:

Real-time presence notification:

The most resource-intensive operation performed in a chat system is not sending messages. It is rather keeping each online user aware of the online-idle-offline states of their friends, so that conversations can begin.

Deploy Smaller Changes, More Frequently *

- Decouple feature releases from code deployments
- Deploy features in a disabled state, using feature flags
- Require all developers check code into trunk daily (at least)
- Practice deploying smaller changes, which dramatically reduces risk and improves MTTR
The First Way: Outcomes

- Everyone’s work is visible
- Work in process is limited
- Lead times go down
- We get in the habit of daily deployments, decoupling deploys from sprint boundaries
- Determinism in the release process
- Consistent Dev, Test and Production environments, all properly built before deployment begins
- Features being deployed daily without catastrophic failures
- Decreased lead time
- Faster cycle time and release cadence
What would you want everyone to learn about Kanban?
“deployes per day”

vs.

“lead time”
Customer request → Design & Analysis
  LT: 2 weeks
  VA: 2 days

Design approval → Estimation
  %C/A: 60%
  LT: 1 week
  VA: 2 hours
  Backlog
  LT: 2 weeks
  %C/A: 60%
  LT: 1 week
  VA: 1 hour

Development (including test automation)
  %C/A: 80%
  LT: 1 week
  VA: 4 days

Exploratory and performance testing
  %C/A: 75%
  LT: 2 weeks
  VA: 1 day

Showcase and UAT
  %C/A: 50%
  LT: 1 week
  VA: 1 hour

Change approval
  %C/A: 90%
  LT: 1 week
  VA: 1 hour

Production deployment
  %C/A: 80%
  LT: 1 hour
  VA: 0 hours

Verify customer receives expected value
  %C/A: 30%
  LT: 3 days
  VA: 0 hours

Aggregate values:
  Total lead time: 10 weeks
  Value added time: 7.5 days
  Percent complete and accurate: 8.6%

Source: Lean Enterprise (upcoming): Jez Humble, Joanne Molesky, and Barry O'Reilly
When Lead Times == 3 Months

LT: 80d
VAT: 22d
CT: 104d

Source: Joseph Enochs

@RealGeneKim
From Worst to Best in 9 Months: Implementing a Drum-Buffer-Rope Solution in Microsoft’s IT Department

By David J. Anderson & Dragos Dumitriu, Microsoft Corporation, November 2005

Abstract
This is a case study about implementing common sense changes where they were needed. It’s a story not about the brilliance of the Theory of Constraints (TOC) but rather TOC playing a role as permission giver, reinforcing the beliefs of a manager and encouraging him to do the right thing. It’s also a story about simplicity – making just a few simple changes, collecting less data, spending less time on overhead and bureaucracy and more on productive tasks.

- “…lead time went from 155 days to 22 days. Lead times were so good that they created a new SLA guarantee of 25 days…”
Training Everyone Should Take!

- Dominica DeGrandis: “Kanban and DevOps” 2 Day Workshop
  - [http://www.ddegrandis.com/services/kanban-training/](http://www.ddegrandis.com/services/kanban-training/)
### One Of The Highest Predictors Of Performance

<table>
<thead>
<tr>
<th>Scenario A (Pathological)</th>
<th>Scenario B (Bureaucratic)</th>
<th>Scenario C (Generative)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Information is hidden</td>
<td>Information may be ignored</td>
<td>Information is actively sought</td>
</tr>
<tr>
<td>Messengers are “shot”</td>
<td>Messengers are tolerated</td>
<td>Messengers are trained</td>
</tr>
<tr>
<td>Responsibilities are shirked</td>
<td>Responsibility is compartmented</td>
<td>Responsibilities are shared</td>
</tr>
<tr>
<td>Bridging between teams and functional groups is discouraged</td>
<td>Bridging between teams and functional groups is allowed but discouraged</td>
<td>Bridging between teams and functional groups is rewarded</td>
</tr>
<tr>
<td>Failure is covered up</td>
<td>Organization is just and merciful</td>
<td>Failure causes enquiry</td>
</tr>
<tr>
<td>New ideas are crushed</td>
<td>New ideas create problems</td>
<td>New ideas are welcomed</td>
</tr>
</tbody>
</table>
One Of The Highest Predictors Of Performance

<table>
<thead>
<tr>
<th>Scenario A (Pathological)</th>
<th>Scenario B (Bureaucratic)</th>
<th>Scenario C (Generative)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Information is hidden</td>
<td>Information may be ignored</td>
<td>Information is actively sought</td>
</tr>
<tr>
<td>Messengers are “shot”</td>
<td>Messengers are tolerated</td>
<td>Messengers are trained</td>
</tr>
<tr>
<td>Responsibilities are shirked</td>
<td>Responsibility is compartmented</td>
<td>Responsibilities are shared</td>
</tr>
<tr>
<td>Bridging between teams and functional groups is discouraged</td>
<td>Bridging between teams and functional groups is allowed but discouraged</td>
<td>Bridging between teams and functional groups is rewarded</td>
</tr>
<tr>
<td>Failure is covered up</td>
<td>Organization is just and merciful</td>
<td>Failure causes enquiry</td>
</tr>
<tr>
<td>New ideas are crushed</td>
<td>New ideas create problems</td>
<td>New ideas are welcomed</td>
</tr>
</tbody>
</table>

How do you use Leankit?
• Doing
  – Epics
  – Lanes
  – Meetings
• @Waiting
• @Office
• @Internet
DevOps Enterprise Summit

- Save the date: October 21-23, 2014
- DevOps Enterprise is a conference for horses, by horses
  - Macy’s, Disney, GE Capital, Blackboard, Telstra, US Department of Homeland Security, CSG
- Leaders driving DevOps transformations will talk about
  - The business problem they set out to solve
  - The obstacles they had to overcome
  - The business value they created
- Submit talks at: http://devopsenterprisesummit.com/
YOUR QUESTIONS ANSWERED
Q&A: DevOps

• What are the major challenges typically faced when trying to implement DevOps in organizations?
• How does Release Engineering fit into the DevOps framework (specifics on the interaction between DEV, RE, and Ops)
• Please share techniques for inspiring executives to appreciate the value of DevOps enough that they will invest in it.
DevOps and Kanban

• Specifics of implementing Kanban for Ops...lots of examples for Engineering...not so many for Operations where there are more work streams?
• What underlying principles and values are most effective to get developers to think about delivery through to production?
• Kanban in firefighting teams - how to limit WIP when stories rarely complete before interruption?
• Excited to hear why Kanban should fight against other IT processes such as itil cobit iso and the like?
Q&A: Scrum vs. Kanban

• How do you get developers and ops to work together when the former use Scrum and the latter use Kanban?
• Is Kanban better than Scrum for an organization with quarterly releases?
• Kanban for large projects vs Sprints for large projects?
• Can DevOps be executed using the SAFe methodology?
Q&A: Getting Started

• How to get started... Baby steps or big leap?
• How does a DevOps approach change to accommodate a large vs. small organization?
Learning Resources

- The Top 11 Things You Need to Know About DevOps: http://itrevolution.com/devops-blog/
- Visit Gene’s website at: www.itrevolution.com
  - Blogs, whitepapers, books and more
Thank you!