

Enterprise DevOps at Scale

12/19/2012

Gap Inc Engineering

Cool Geeks DevOps Meetup

PRE-AMBLE WARNINGS:

- "DEVOPS" IS A ORGANIZATIONAL CULTURAL CHANGE AND MINDSET, NOT A THING YOU CAN BUY.
- THIS SH** IS HARD AT ENTERPRISE SCALE!

First Rule of Enterprise DevOps:

1. You must have a CIO and IT Executive leadership team with **COURAGE and BOLDNESS**.

Pivoting to a "DevOps" centric enterprise is NOT a trivial task for the meek. Anyone saying otherwise is FOS (full of sh**).



Second Rule of Enterprise DevOps:

2. You must have a technically STRONG leadership team from VP down to individual contributor.

These people must be willing to BUST their ASS and they need RULE # 1 to be SOLID.

- No PowerPoint architects need apply.
- PWHOK (People with hands on keyboards)



Third Rule of Enterprise DevOps:

3. The DOERS and the LEADERS MUST HAVE PATIENCE.
 - The enterprise is not as nimble as a startup.
 - 4 people agreeing to use Ubuntu LTS and SALT is easier than 20+!
 - This means being patient and letting others within the organization "catch up" to the core.
 - This means empowerment with a "controlled wild west".

Fourth Rule of Enterprise DevOps:

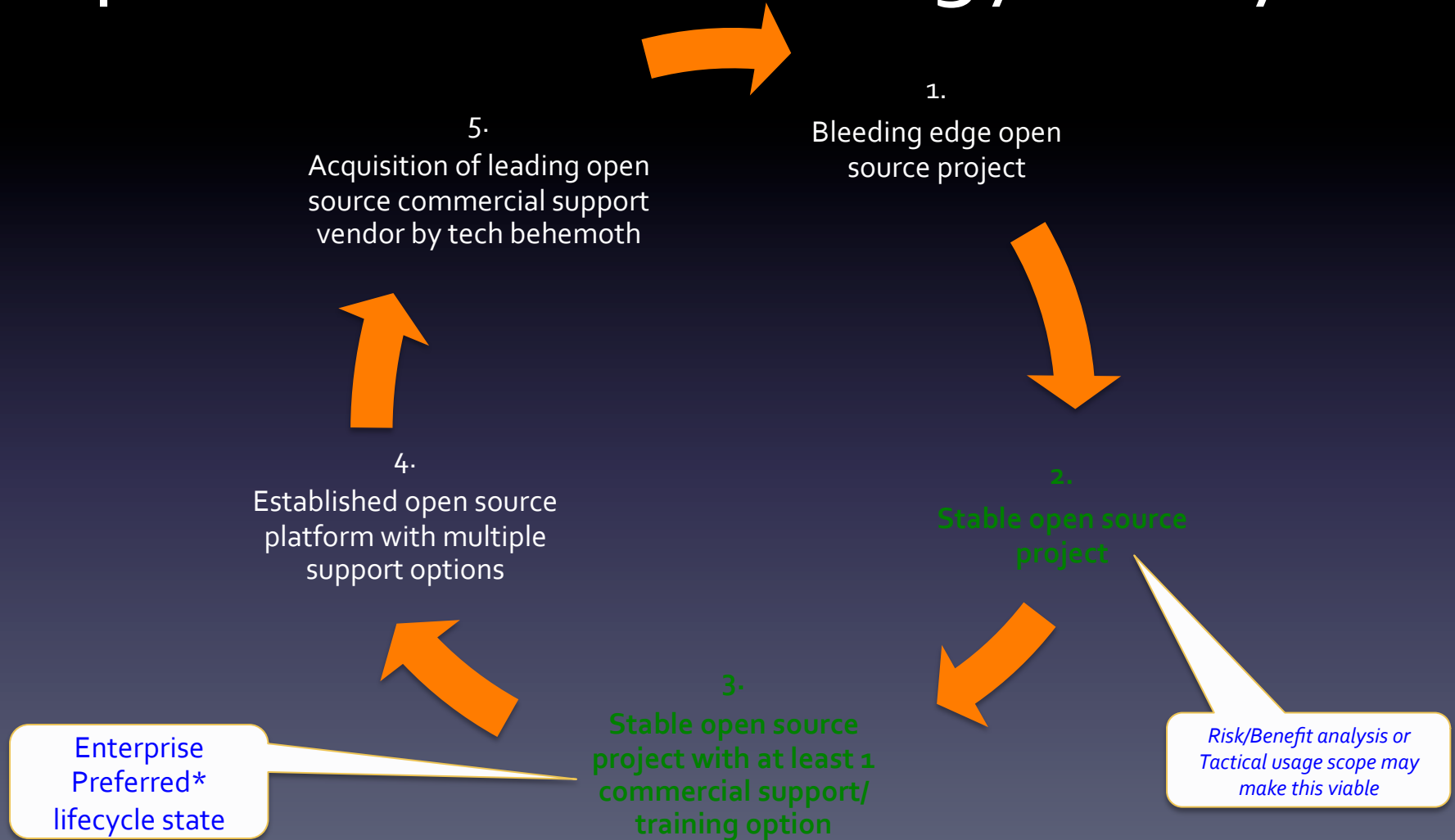
4. In most cases, the IT Leadership Team must give equal legitimacy for open source software with commercial software.

- Gap did this through the creation of core Infrastructure Architecture Teams called Domain 0 (Apps) and Domain 1 (Infra) working together (DevOps).
- The Do and D1 teams created the TAP process to allow executive leadership comfort with open source software.

Do + D1 = Architecture/Development/Infra Formalized Collaboration

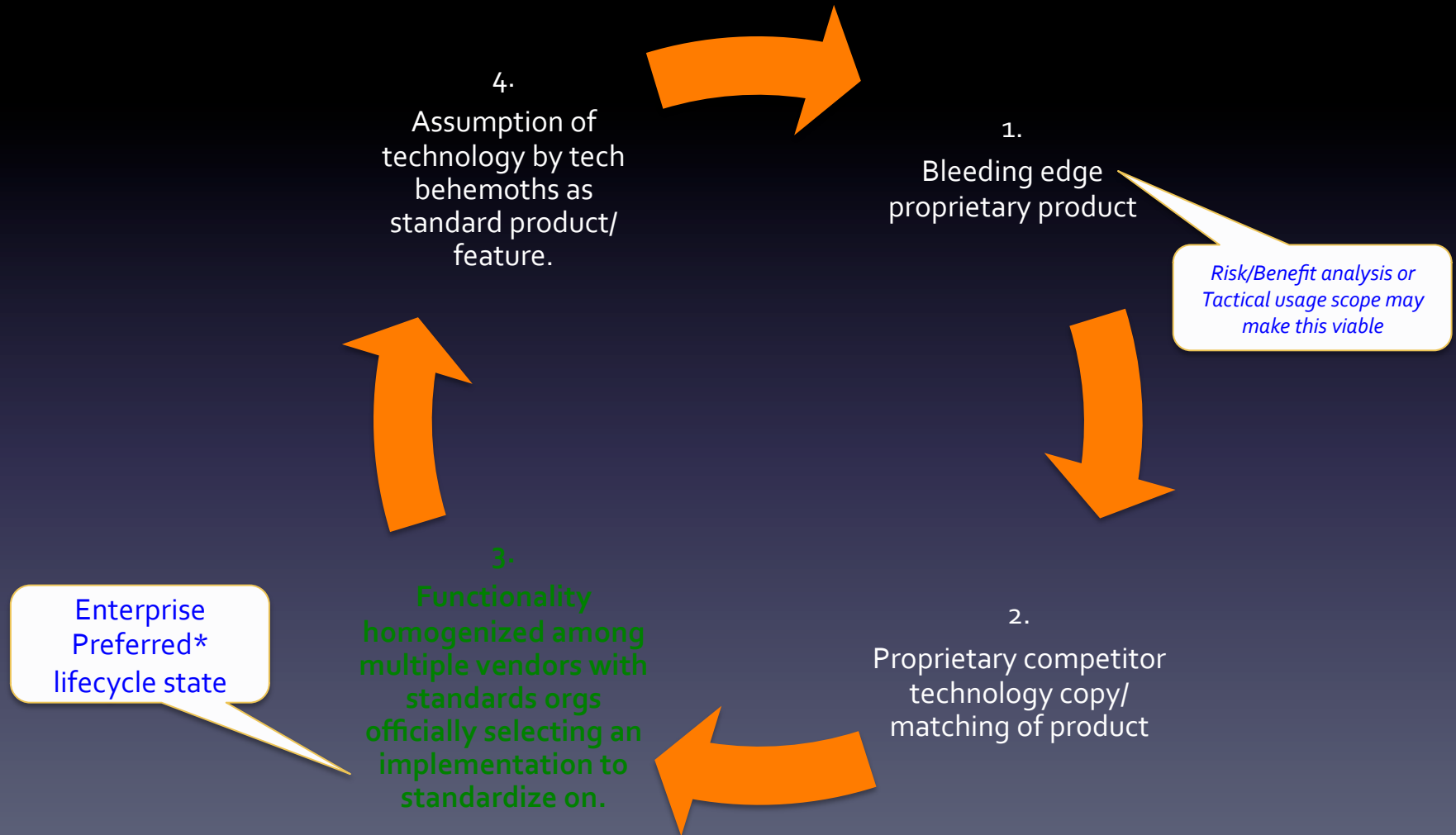
Gap TAP

Open Source Technology Lifecycle

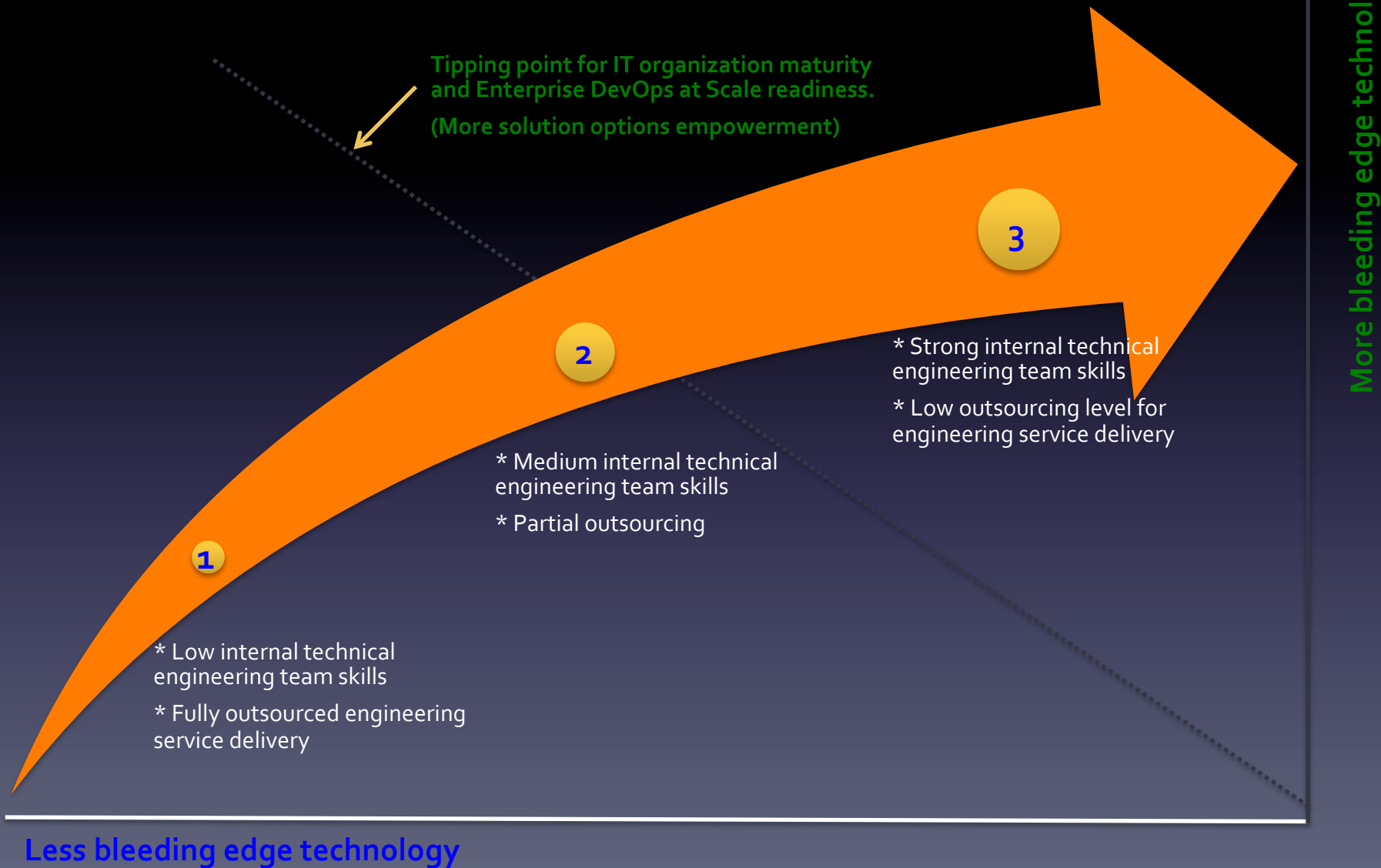


Gap TAP

Proprietary Technology Lifecycle



TAP Organizational Maturity Position



Fifth Rule of Enterprise DevOps:

5. Everyone must be speaking the same technical language.
 - This means the output of the D0 and D1 architectural teams at Gap provided 'Crystal Clear' verbiage for leadership.
- Don't rely on industry buzzwords. Create your own words and explain the relation and difference to industry buzzwords.
- Empower your leadership to rise above the industry buzzwords.
- This allows them to overcome FUD from outsiders (like sales people!)
- [NOTE: Having #1 again can stop outside FUD from interfering with your success]

Pivoting an Enterprise to DevOps

6. Pivoting to a "DevOps" centric enterprise means your organization and leadership:

- MUST be ready for real change in infrastructure operations.
- MUST be willing to eliminate departments, staffing, and some vendor relationships.

Pivoting an Enterprise to DevOps

7. Let's talk about that asterisk around "eliminate departments, staffing and some vendor relationships"
 - A key cultural premise in DevOps is making things work quickly and efficiently as possible.
 - For most developers and infra teams this means using open standards and open source centric solutions.
 - **Why?** because open standards mean that documentation is free and so is source code: meaning you can always just "Make it work" .

Enterprise DevOps Impacts:

Enterprise DevOps Impact 1:

- You may have to displace your entrenched configuration software vendors for something open source.
- **Why?** Long after a vendor is gone, open source code lives and any talented developer can fix your problems.....on your schedule.

Enterprise DevOps Impacts:

Enterprise DevOps Impact 2:

- Efficiency often means slow, irritating processes, software, and people are replaced with automated code.
- **Yep.** Get ready, entire departments may be gone in a matter of months. 2 people that can code become more valuable than entire groups.

Enterprise DevOps Impacts:

Enterprise DevOps Impact 3:

- Efficiency, automated infrastructure code, and DevOps empowerment is usually the antithesis of a traditional outsourcing relationship.
- **Wake Up.** “Your mess for less” no longer works. Co-sourcing is better. When the work of traditional support engineers is automated, what are you paying them for after all? Plus they stinking slow you down.

Enterprise DevOps Impacts:

When Gap made the decision to make the DevOps culture "strategic" it meant a lot in IT changed:

- Invest in highly technical people that solve problems with data and code rather than meetings and expensive monolithic solutions.
- Use Open standards and open source software whenever it makes sense (most of the time :-))
- See commodity hardware and operating systems as strategic, but deal with the fact that other ones exist in the enterprise.
- Empower infra and app teams to work together unencumbered.

Enterprise DevOps Impacts:

Let's re-visit the key point "Empower infra and app teams to work together unencumbered"

What exactly does this mean?

- It means letting 2 groups speak the same language to get things done: **CODE.**
For many, this means powerful and simple: RUBY.
- It means putting powerful capabilities in the hands of people via code.
True empowerment.

How to safely do this?

- **Make sure you implement TDD (Test Driven Development)**
- **You are free to write as much code as you want and COMMIT away: But nothing makes it to any machine without passing a series of automated tests.**

The Pipeline becomes the **CODE REVIEW POLICE & REBEL ALLIANCE** all-in-one.

DevOps Empowerment

What are ways to automate and empower with code?

Get Infrastructure and App Teams to align on Tools:

- Start with easier things like: Linux Distros, YUM Repos.
- Move on to Configuration Management (CM) tools. When in doubt on what to use, just take a step back and see where you fit on the TAP Lifecycle.
 - For example, you might have 2 folks that really think the latest SALTnPEPA tools is bombass. But, perhaps it is great for a startup... but not so much for an enterprise that sits at the periphery of the DevOps Maturity Curve.

DevOps Empowerment

What are ways to automate and empower with code?

Get Infrastructure and App Teams to align on Tools:

- Start with easier things like: Linux Distros, YUM Repos.
- Move on to Configuration Management (CM) tools. When in doubt on what to use, just take a step back and see where you fit on the TAP Lifecycle.
 - For example, you might have 2 folks that really think the latest SALTnPEPA tools is bombass. But, perhaps it is great for a startup... but not so much for an enterprise that sits at the periphery of the DevOps Maturity Curve.

DevOps Empowerment

On the CM topic, there are several important things you should consider when picking one in an enterprise:

- 1) When you choose to make the CM tool be part of your app deployment pipeline, you need app developers speaking to infra using the same language: RUBY.
- 2) When you choose to have the CM tool install an app (particularly the case with COTS apps) you need the flexibility to work with package installers and shell scripts, as well as custom ruby code.
- 3) Infra and App folks like the convenience of obfuscation of OS facilities through CM tools, but they also need it to be BROAD enough to be useful. Lightweight Resource Providers or LWRPs are super useful.
- 4) Logical, sequential processing of code like developers expect. Not all CM tools make this easy. Making it possible is not the same as easy.

DevOps Empowerment

In the case of Gap, Chef meet all our requirements:

- 1) TAP lifecycle position for open source software.
- 2) We wanted to wrap the CM tooling in with app deployments.
- 3) We had a significant amount of COTS and Custom Developed apps we wanted to automate the installs for using the CM tool.
- 4) Because of #3, and because of app developers preferences in #2, and because we had some legacy CFEngine stuff to migrate, sequential processing was important.
- 5) Infra system engineers like the LWRP implementation options for installing packages and interacting with the server OS (with the ease of puppet style code).

“Thanks!”

@jbpadgett

<http://Padgeblog.com>